Title

Computational and Machine Learning Methods for Understanding Gene Regulation and Variant Effects

Author

Benegas, Gonzalo Segundo

Publication Date

2023

Peer reviewed|Thesis/dissertation

Computational and Machine Learning Methods for Understanding Gene Regulation and Variant Effects

by

Gonzalo Benegas

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computational Biology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Yun S. Song, Chair
Professor Ian Holmes
Assistant Professor Peter Sudmant
Assistant Professor Karthik Shekhar

Fall 2023

Computational and Machine Learning Methods for Understanding Gene Regulation and Variant Effects

Abstract

Computational and Machine Learning Methods for Understanding Gene Regulation and Variant Effects

by

Gonzalo Benegas

Doctor of Philosophy in Computational Biology

University of California, Berkeley

Professor Yun S. Song, Chair

The field of genomics has been advancing at a fast pace ever since the development of high-throughput sequencing technologies. While we have access to more data than ever before, the number of open questions has only increased. In this dissertation, I present novel machine learning techniques to draw insights from genomic data. First, I tackle the analysis of alternative splicing — a crucial but overlooked step in gene regulation — from short-read single-cell RNA-seq data. To account for the large scale and sparsity of such data, I develop scQuint, a suite of efficient probabilistic methods for dimensionality reduction and differential splicing. Next, I approach the problem of genome-wide variant effect prediction with a new direction: DNA language models. We first propose GPN, trained on unaligned genomes, and apply it to study genetic variants in *Arabidopsis thaliana*. GPN shows an improved power for highlighting variants under negative selection as well as those affecting traits. Furthermore, I show that GPN learns important genomic features such as gene annotations and transcription factor binding site motifs, without any supervision. We then present GPN-MSA, a DNA language model trained on whole-genome alignments of vertebrates, and showcase its excellent performance predicting deleteriousness across the entire human genome. These contributions not only pave the way for enhanced genomic understanding but also propose a methodological shift in genome analysis.

To my father.

Contents

Contents ii

List of Figures iv

List of Tables vi

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Outline . . . 2

2 scQuint 4
- 2.1 Introduction . . . 4
- 2.2 Results . . . 6
- 2.3 Discussion . . . 17
- 2.4 Materials and Methods . . . 19

3 GPN 39
- 3.1 Introduction . . . 39
- 3.2 Results . . . 42
- 3.3 Discussion . . . 45
- 3.4 Materials and Methods . . . 46

4 GPN-MSA 55
- 4.1 Introduction . . . 55
- 4.2 Results . . . 56
- 4.3 Discussion . . . 58
- 4.4 Materials and Methods . . . 61

Bibliography 68

A Supplementary Information for Chapter 2 80

B Supplementary Information for Chapter 3 100

C Supplementary Information for Chapter 4 **119**

List of Figures

2.1 Clustering patterns by cell type and plate . . . 7
2.2 Overview of scQuint . . . 9
2.3 Comparison of splicing latent spaces obtained with PCA and VAE . . . 10
2.4 Evaluation of differential splicing test on simulated data . . . 12
2.5 Interactive visualizations of splicing patterns . . . 25
2.6 Splicing patterns in *BICCN Cortex* . . . 27
2.7 Global analysis of *Tabula Muris* . . . 29
2.8 Splicing in developing marrow B cells from *Tabula Muris* . . . 31
2.9 Alternative splicing patterns across epithelial and endothelial cell types . . . 33
2.10 Patterns across tissues. . . . 35
2.11 Associations between splicing factors and alternative splicing . . . 37

3.1 Overview of GPN (Genomic Pre-trained Network) . . . 41
3.2 Unsupervised clustering of genomic windows . . . 50
3.3 Sequence logos derived from model predictions . . . 51
3.4 Variant effect prediction: *in silico* mutagenesis . . . 52
3.5 Variant effect prediction: rare vs. common . . . 53
3.6 Variant effect prediction: GWAS . . . 54

4.1 Overview of GPN-MSA . . . 59
4.2 Comparison of variant effect prediction results . . . 60

A.1 Coverage artifacts in mammary gland basal cells from *Tabula Muris* . . . 82
A.2 Technical artifacts in *BICCN Cortex* . . . 83
A.3 Splicing latent space when alternative intron counts are shuffled . . . 84
A.4 Comparison with LeafCutter . . . 85
A.5 Marker genes for cell types in *BICCN Cortex* . . . 87
A.6 PSI distribution of `Pgm2_32951` . . . 88
A.7 PSI distribution of `Rbfox1_26172` . . . 89
A.8 PSI distribution of `Nrxn1_8067` . . . 90
A.9 PSI distribution of `Smarca4_28720` . . . 91
A.10 PSI distribution of `Foxp1_11076` . . . 92
A.11 Full-gene view of novel alternative TSS in *Itpr1* . . . 93

A.12 PSI distribution of `Itpr1_26257` . . . 94
A.13 PSI distribution of `Khk_24896` . . . 95
A.14 Full plot of associations between splicing factors and alternative splicing . . . 96
A.15 PSI distribution of `Khdrbs3_25689` . . . 97
A.16 PSI distribution of `Mbnl2_25376` . . . 98
A.17 PSI distribution of `Mbnl2_25378` . . . 99

B.1 UMAP visualization of k-mer spectrum of different windows . . . 102
B.2 UMAP visualization of GPN embeddings, annotated by repeat family . . . 103
B.3 Additional GPN sequence logos . . . 104
B.4 Perplexity on select positions . . . 104
B.5 Promoter motifs predicted by GPN and matching motifs in PlantTFDB . . . 105
B.6 Comparison of GPN models trained with different loss weights on repeats . . . 114
B.7 Allele frequency (AF) of variants in different GPN score bins . . . 115
B.8 Rare vs. common odds ratios for different thresholds . . . 115
B.9 Rare vs. common odds ratios for specific variant categories . . . 116
B.10 Comparison of GPN models trained on a different number of species . . . 117
B.11 GWAS hit odds ratios for different thresholds . . . 117
B.12 Odds ratios for GWAS hits, using the Bonferroni correction . . . 118

C.1 Phylogenetic tree of 100 vertebrates . . . 121
C.2 Functional impact of GPN-MSA . . . 122
C.3 Functional enrichment and depletion of deleterious tail . . . 123
C.4 Receiver Operating Characteristic and Precision-Recall curves . . . 124
C.5 Enrichment in rare vs. common gnomAD missense variants . . . 125
C.6 Histogram of GPN-MSA scores . . . 126
C.7 Variant effect prediction with conservation scores . . . 127
C.8 Ablation study . . . 128
C.9 GPN-MSA logo track on the UCSC Genome Browser . . . 128
C.10 Variant effect prediction with Nucleotide Transformer models . . . 129
C.11 Variant effect prediction with different norms of Enformer delta predictions . . . 129

List of Tables

2.1 Overview of analyzed data sets 12
2.2 Summary of differential expression and splicing for select cell types 16
2.3 VAE hyperparameters 21

A.1 Summary of methods available to analyze transcript variation 81

B.1 Genome assemblies used for training 100
B.2 Test perplexity 100
B.3 GPN training hyperparameters 101

C.1 GPN-MSA training hyperparameters 119
C.2 Functional annotations considered in our analysis of functional enrichment . . . 120

Acknowledgments

This PhD has been a wonderful journey. I profoundly thank everyone who helped me set a foot in Berkeley, especially Hernan G. Garcia, Alejo Salles and Federico J. Fernandez.

Being advised by Yun S. Song has truly been a privilege. Thank you for your inescapable curiosity and generous attention, spanning both green ideas and red signals in LaTeX. I am grateful to the wonderful people in the Song Lab for the rich environment they continue to create. Thank you Jonathan Fischer for helping me navigate my first paper from start to finish. Thank you Sanjit Batra for crafting together a completely new research direction for our second project. Thank you Carlos Albors, Alan Aw and Chengzhong Ye for infusing our third project with enthusiasm and fresh ideas. Thank you other members whom I have not formally collaborated with but have undeniably participated in flow of ideas: Nick Bhattacharya, Jeffrey Chan, Yun Deng, Will DeWitt, Dan Erdmann-Pham, Milind Jagota, Antoine Koehl, Jesús Martínez-Gómez, Sebastian Prillo, Jeffrey Spence, Neil Thomas, Yutong Wang, Bear Xiong, Jane Yu and Fanding Zhou.

I am grateful for the kindness and warmth of the Center for Computational Biology community. I have had the honor to share these years with Kennedy Agwamba, Ryan Chung, Adam Gayoso, Graham Northrup and Isabel Serrano. Thank you Kate Chase and Xuan Quach for always being present. I thank Ian Holmes, Peter Sudmant and Karthik Shekhar for serving on my dissertation committee, and Haiyan Huang for serving on my qualifying exam committee.

I have been blessed by beautiful friendships during my time at Berkeley. Thank you Diana, Mauri, Ian, Debora and Daniel for having so much fun together. Thank you Bonnie mates for all we shared, and Ben and Kelsey for tying it together. Thank you to my friends in Argentina, a strong root and smile throughout my life.

I thank my family for your constant support of my passions, even when they would take me far away from home. Thanks to my father, certainly the first to bring computers and biology into my life; I would have loved to share my discoveries. Thank you to my grandmother Niti for being an example of living with joy.

Finally, I thank my loving partner Joana. At the start of the pandemic we marked a line in our dining table to delineate two desks — I hope this simple gesture remains a symbol of our shared journey and strength together.

Chapter 1

Introduction

1.1 Background

The field of genomics has seen remarkable advancements in recent decades. This progress has been fueled by rapid technological developments and a growing understanding of the complex interplay between genetic sequences and their functional outputs. At the heart of this field lies the endeavor to decode the wealth of information encoded within our DNA, which serves as the blueprint for building and operating biological organisms.

One of the most significant applications of genomics is personalized medicine. The ability to tailor medical treatment based on an individual's genetic makeup is a big promise in healthcare. By understanding the genetic basis of diseases, treatments can be more effectively targeted, leading to better patient outcomes and fewer side effects [47].

In agriculture, genomics is playing a pivotal role in the development of more resilient and productive crops. Through genetic engineering and breeding programs informed by genomic insights, crops can be made more resistant to pests, diseases, and environmental stresses. This progress is crucial for ensuring food security in the face of a growing global population and changing climate conditions [165].

Additionally, genomics has applications in other fields such as forensic science [53] and ancestry inference [104], providing powerful tools for identifying individuals and tracing genetic lineages. This has profound implications for law enforcement, historical research, and understanding human migration patterns.

However, the field of genomics is not without its challenges. One such challenge is understanding the complex mechanisms of gene regulation. While transcriptional regulation has been a major focus, especially with the advent of technologies like single-cell RNA sequencing (scRNA-seq), it represents just one layer in the multifaceted process of gene expression. Alternative splicing, a process by which a single gene can produce multiple transcript isoforms, is a key component of this regulatory complexity in higher eukaryotes. Despite its importance, alternative splicing is often underrepresented in genomic studies, largely due to technical challenges in accurately quantifying these events with current

sequencing technologies.

Another major challenge in genomics is identifying the functional impacts of genetic variants. While genome-wide association studies (GWAS) have been successful in associating genetic variants with particular traits or diseases, pinpointing the causal variants and understanding their mechanisms of action remains difficult. The sheer number of variants and the complexity of their interactions within the genome make it a daunting task. This challenge is compounded by the high cost and labor intensity of experimental validation methods.

Thus, the field of genomics stands at a crossroads, where the need for advanced computational methods to analyze and interpret genomic data has never been greater. These methods hold the key to unlocking the full potential of genomic information, enabling us to understand and harness the complexities of life at its most fundamental level.

1.2 Outline

In this dissertation, I present computational methods to improve our understanding of two areas of genomics: gene regulation and the effect of genetic variants. The specific tasks and computational tools developed in each chapter are summarized as follows:

- Chapter 2: alternative splicing analysis with variational auto-encoders and generalized linear models.
- Chapter 3: variant effect prediction with alignment-free DNA language models.
- Chapter 4: variant effect prediction with alignment-based DNA language models.

I will now introduce the content of each chapter in more detail.

In Chapter 2, I undertake the analysis of alternative splicing across numerous diverse murine cell types from two large-scale single-cell datasets—the *Tabula Muris* [117] and BRAIN Initiative Cell Census Network [162]—while accounting for understudied technical artifacts and unannotated events. I find strong and general cell-type-specific alternative splicing, complementary to total gene expression but of similar discriminatory value, and identify a large volume of novel splicing events. I specifically highlight splicing variation across different cell types in primary motor cortex neurons, bone marrow B cells, and various epithelial cells, and I show that the implicated transcripts include many genes which do not display total expression differences. To elucidate the regulation of alternative splicing, I build a custom predictive model based on splicing factor activity, recovering several known interactions while generating new hypotheses, including potential regulatory roles for novel alternative splicing events in critical genes like *Khdrbs3* and *Rbfox1*. I make the results available using public interactive browsers to spur further exploration by the community.

Inspired by recent progress in natural language processing, unsupervised pre-training on large protein sequence databases has proven successful in extracting complex information related to proteins [116]. These models showcase their ability to learn variant effects in

coding regions using an unsupervised approach [97]. Expanding on this idea, in Chapter 3 I introduce the **G**enomic **P**re-trained **N**etwork (**GPN**), a model designed to learn genome-wide variant effects through unsupervised pre-training on genomic DNA sequences. The model also successfully learns gene structure and DNA motifs without any supervision. To demonstrate its utility, I train GPN on *unaligned* reference genomes of *Arabidopsis thaliana* and seven related species within the Brassicales order, and evaluate its ability to predict the functional impact of genetic variants in *Arabidopsis thaliana* by utilizing allele frequencies from the 1001 Genomes Project [1] and a comprehensive database of GWAS [137]. Notably, GPN outperforms predictors based on popular conservation scores such as phyloP [110] and phastCons [122].

Whereas protein language models have demonstrated remarkable efficacy in predicting the effects of missense variants, DNA counterparts have not yet achieved a similar competitive edge for genome-wide variant effect predictions, especially in complex genomes such as that of humans. To address this challenge, in Chapter 4 I introduce GPN-MSA, a novel framework for DNA language models that leverages whole-genome sequence alignments across multiple species and takes only a few hours to train. Across several benchmarks on clinical databases (ClinVar [74], COSMIC [132], and OMIM [123]) and population genomic data (gnomAD [28]), our model for the human genome achieves outstanding performance on deleteriousness prediction for both coding and non-coding variants, surpassing widely-used models such as CADD [115], ESM-1b [116], SpliceAI [61] and Enformer [7].

Chapter 2

Robust and annotation-free analysis of alternative splicing across diverse cell types in mice

This is joint work with Jonathan Fischer and Yun S. Song, published in *eLife* [11]. I would like to thank Angela Oliveira Pisco, Spyros Darmanis, and Kif Liakath-Ali for helpful discussions. I also thank the Chan Zuckerberg Biohub for hosting our cell×gene sessions and Aaron McGeever for assistance.

2.1 Introduction

The past decade's advances in single-cell genomics have enabled the data-driven characterization of a wide variety of distinct cell populations. Despite affecting more than 90% of human pre-mRNAs [149], isoform-level variation in gene expression has often been ignored because of quantification difficulties when using data from popular short-read sequencing technologies such as 10x Genomics Chromium and Smart-seq2 [109]. Long-read single-cell technologies, which greatly simplify isoform quantification, are improving [26, 50, 145, 76, 64], but remain more costly and lower-throughput than their short-read counterparts. For these reasons and others, short-read datasets predominate and we must work with short reads to make use of the rich compendium of available data. In response, researchers have developed several computational methods to investigate splicing variation despite the sizable technical challenges inherent to this regime. A selection of these challenges and methods are summarized in Appendix A.

To complement single-cell gene expression atlases, we analyze alternative splicing in large single-cell RNA-seq (scRNA-seq) datasets from the *Tabula Muris* consortium [117] and BRAIN Initiative Cell Census Network (BICCN) [162]. These data span a broad range of mouse tissues and cell types, and remain largely unexplored at the level of transcript variation. During our initial analyses, we encountered pervasive coverage biases, a heretofore largely

unappreciated mode of technical variation which greatly confounds biological variation across cell types. Unsatisfied with the performance of current methods when confronted by these biases, we implemented our own quantification, visualization, and testing pipeline, named scQuint (single-cell quantification of introns), which allowed us to continue our analyses in a robust, annotation-free, and computationally tractable manner. Parts of the scQuint pipeline are based on adaptations of the bulk RNA-seq alternative splicing analysis method LeafCutter [81] to handle the unique challenges of scRNA-seq data. As we demonstrate in subsequent sections, our modifications in the quantification, statistical modeling, and optimization procedures lead to improved robustness, scalability, and calibration when working with data from single cells (Figure A.4, also see Methods).

Applying scQuint to these data sets, we find a strong signal of cell-type-specific alternative splicing and demonstrate that cell type can be accurately predicted given only splicing proportions. Moreover, our annotation-free approach enables us to detect a large quantity of cell-type-specific novel splicing events. In certain cell types, particularly the neuron subclasses, as many as 30% of differential splicing events that we detect are novel. In general, across the many considered cell types and tissues in both datasets, we find only a narrow overlap between the top differentially expressed and the top differentially spliced genes within a given cell type, illustrating the complementarity of splicing to expression. Our examination of neurons in the primary motor cortex suggests that splicing distinguishes neuron classes and subclasses as readily as does expression. We showcase alternative splicing patterns specific to the GABAergic (inhibitory) and Glutamatergic (excitatory) neuron classes as well as the subclasses therein. The implicated transcripts include key synaptic molecules and genes which do not display expression differences across subclasses. In developing marrow B cells, we find alternative splicing and novel transcription start sites (TSS) in critical transcription factors such as *Smarca4* and *Foxp1*, while further investigation reveals dissimilar trajectories for expression and alternative splicing in numerous genes across B cell developmental stages. These findings buttress our belief in the complementary nature of these processes and provide clues to the regulatory architecture controlling the early B cell life cycle. To facilitate easy exploration of these datasets and our results, we make available several interactive browsers as a resource for the genomics community.

Finally, to advance our understanding of alternative splicing regulation, we build a statistical machine learning model to predict splicing events by leveraging both the expression levels and splicing patterns of splicing factors across cell types. This model recovers several known regulatory interactions such as the repression of splice site 4 exons in neurexins by *Khdrbs3*, while generating new hypotheses for experimental follow-up. For example, in addition to the regulatory effect of the whole-gene *Khdrbs3* expression, the model predicts a regulatory role for a novel alternative TSS in this gene. In aggregate, our results imply that alternative splicing serves as a complementary rather than redundant component of transcriptional regulation and supports the mining of large-scale single-cell transcriptomic data via careful modeling to generate hypothetical regulatory roles for splicing events.

2.2 Results

Methods overview

Robust, annotation-free quantification based on alternative introns. Most methods rely on the assumption that coverage depth across a transcript is essentially uniform (e.g., *Akr1r1*, Figure A.1a). We instead found that Smart-seq2 data [109] frequently contain sizable fractions of genes with coverage that decays with increasing distance from the 3' ends of transcripts. For example, in mammary gland basal cells from the *Tabula Muris* data set [117], *Ctnbb1* shows a gradual drop in coverage (Figure A.1b) while *Pdpn* displays an abrupt reduction halfway through the 3' UTR (Figure A.1c). That the magnitude of these effects varies across technical replicates (plates) suggests they could be artifacts, possibly related to degradation or interrupted reverse transcription. Similar coverage bias artifacts are also apparent in the BICCN primary motor cortex data [162] (Figure A.2).

Such coverage biases affect gene expression quantification, and in some cases these batch effects are sufficient to comprise a significant proportion of the observed variation in expression levels. For the *Tabula Muris* mammary gland data set, a low-dimensional embedding of cells based on gene expression reveals that some cell type clusters exhibit internal stratification by plate (Figure 2.1a). A subsequent test of differential gene expression between plate B002438 and all other plates returns 2,870 significant hits after correction for multiple hypothesis testing, and all manually inspected differentially expressed genes exhibit these types of coverage biases. Perhaps unsurprisingly, quantification at the transcript level is apt to be even more sensitive to these artifacts than gene-level quantification, especially if it is based on coverage differences across the whole length of the transcript. The UMAP embeddings of isoform proportions (kallisto by Bray et al. [22]), exon proportions (DEXSeq by Anders, Reyes, and Huber [3]), 100 bp bin coverage proportions (ODEGR-NMF by Matsumoto et al. [92]) or junction usage proportions across the whole gene (DESJ by Liu et al. [86]) depict a plate clustering pattern which scrambles the anticipated cell type clusters (Figure 2.1b-e).

With these considerations in mind, we sought to quantify transcript variation in a fashion that would be more robust to coverage differences along the transcript. Although some bulk RNA-seq methods such as RSEM [80] can model positional bias, they do so globally rather than in the gene-specific manner we encounter. One potential approach is alternative intron quantification as performed by bulk RNA-seq methods MAJIQ [142], JUM [150], and LeafCutter [81]. Promisingly, quantification via LeafCutter (Figure 2.1f) yields an embedding that displays less clustering by plate than the other approaches we tried. We therefore based scQuint's quantification approach on LeafCutter's, with the key difference of restricting to alternative introns which share a common 3' acceptor site (Figure 2.2). This results in alternative splicing events that are equidistant from the 3' end of transcripts and which are less affected by the coverage biases we observed in scRNA-seq data. The embedding of cells based on our quantification approach (Figure 2.1g) shows less clustering by plate than LeafCutter and other methods.

Another advantage of alternative intron quantification is the ability to easily discover

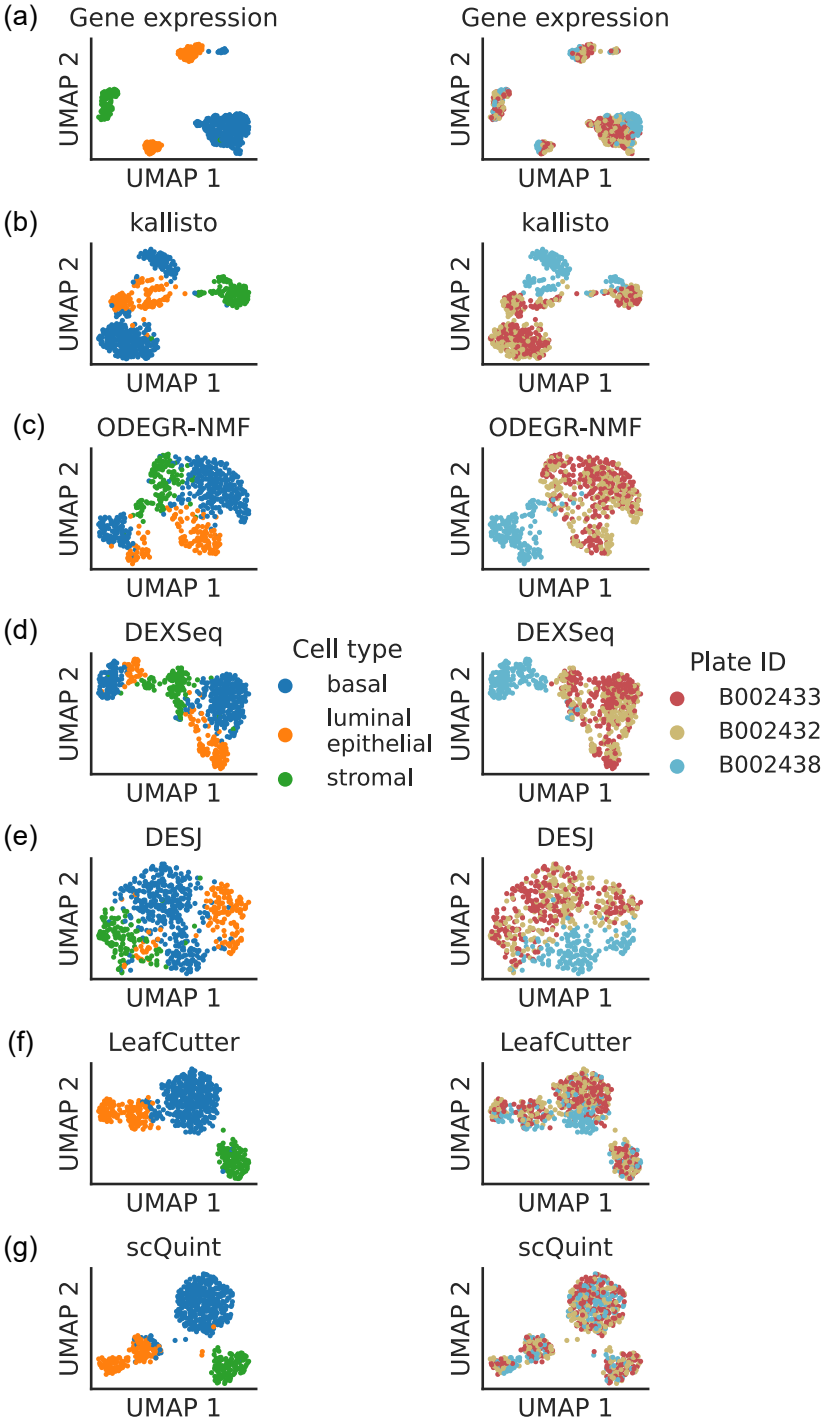


Figure 2.1: **Clustering patterns by cell type and plate in the mammary gland from a three month-old female mouse in Tabula Muris.**

Figure 2.1 *(continued)*: Cell embeddings based on different features were obtained by running PCA (gene expression) or VAE (the rest) followed by UMAP and subsequently colored by cell type (left column) and the plate in which they were processed (right column). **(a)** Gene expression, quantified using featureCounts (log-transformed normalized counts). **(b)** Isoform proportions. Isoform expression was estimated with kallisto and divided by the total expression of the corresponding gene to obtain isoform proportions. **(c)** Coverage proportions of 100 base-pair bins along the gene, as proposed by ODEGR-NMF. **(d)** Exon proportions, as proposed by DEXSeq. **(e)** Intron proportions across the whole gene, as proposed by DESJ. **(f)** Alternative intron proportions quantified by LeafCutter. **(g)** Alternative intron proportions (for introns sharing a 3' acceptor site) as quantified by scQuint.

novel alternative splicing events. Whereas short reads generally cannot be associated with specific transcript isoforms, nor even exons if they partially overlap, split reads uniquely associate with a particular intron. Consequently, intron-based quantification does not depend on annotated transcriptome references and permits the discovery of novel alternative splicing events. This is important since, as detailed later, we estimate up to 30% of cell-type-specific differential splicing events are novel. Other annotation-free methods have been applied to single-cell short-read full-length data, but they do not provide a statistical test for differential splicing between two groups of cells (Table A.1).

We do not recommend using scQuint to analyze alternative splicing in 10X Genomics Chromium data given its strong 3' transcript bias and evidence suggesting that these data can detect about half the number of junctions detected by Smart-seq2 [151]. This imposes a fundamental limit on the number of transcripts that can be distinguished, and we expect alternative intron quantification to be sub-optimal in this setting. Nonetheless, several approaches for differential transcript usage in 10X data have been developed: Sierra [106], SpliZ [103], and a kallisto-based approach which could be adapted for this task [102].

Dimensionality reduction with Variational Autoencoder. To perform dimensionality reduction using splicing profiles, we developed a novel Variational Autoencoder (VAE) [71] with a Dirichlet-Multinomial noise model, a natural distribution for sparse, overdispersed count data (Figure 2.2b, Materials and Methods). For example, the often encountered "binary" splicing [24] can be modeled by fitting a concentration parameter close to zero. VAEs are flexible and scalable generative models which have been successfully applied to analyze gene expression [87] but have not yet been employed to investigate alternative splicing. To verify that we prevent leakage of gene expression information into our splicing profiles, we applied our VAE to embed a shuffled data set obtained by resampling alternative intron counts with a fixed proportion in all cells. This shuffled data set contained expression variability between cells but no splicing differences, and, as hoped, the resulting splicing latent space did not distinguish among cell types, indicating that it captures differences in splicing proportions rather than changes in absolute gene expression (Figure A.3). We compared the latent space obtained with the VAE to the one obtained using Principal Component Analysis (PCA), a

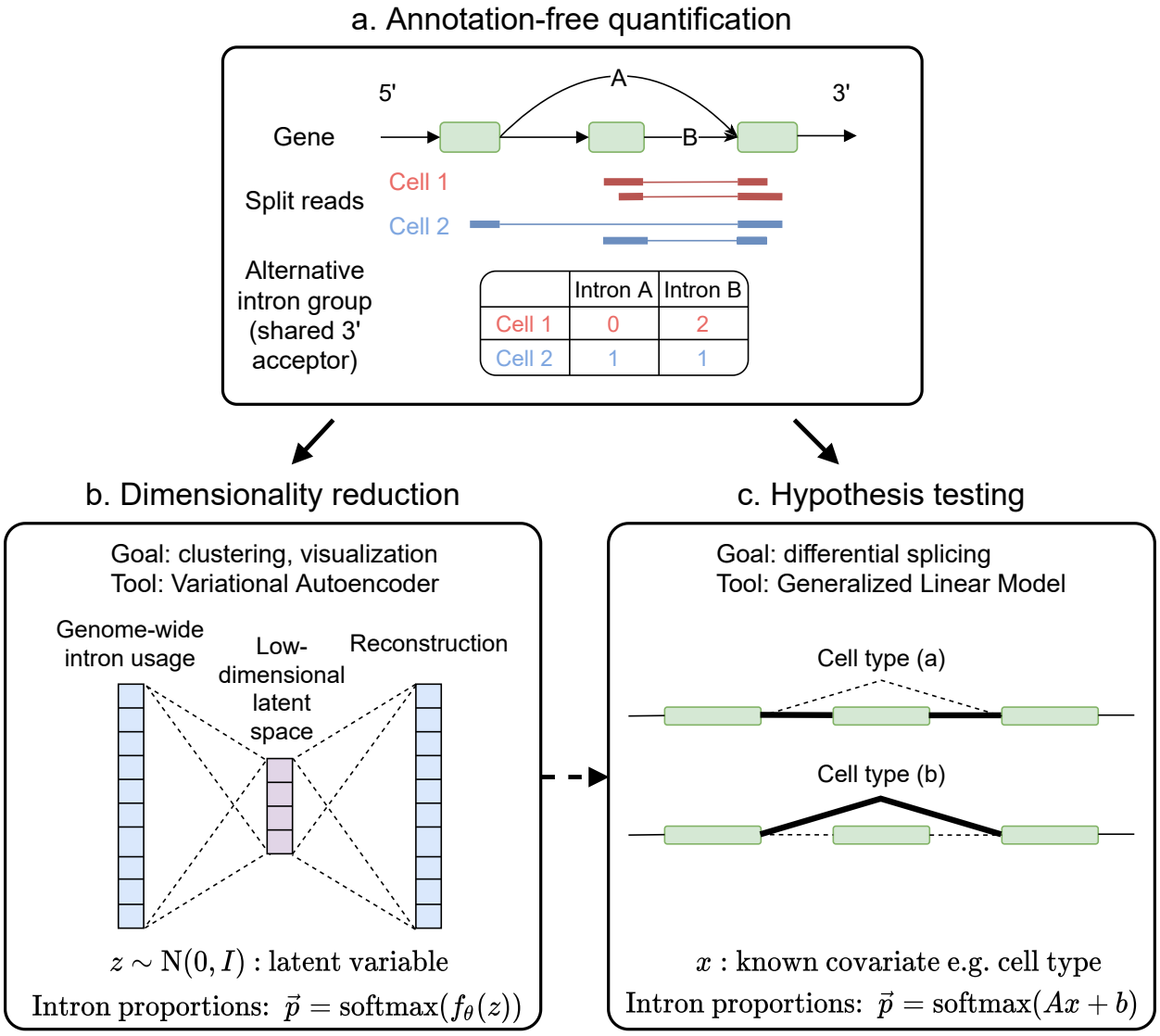


Figure 2.2: **Overview of scQuint. (a)** Intron usage is quantified from split reads in each cell, with introns sharing 3' splice sites forming alternative intron groups. **(b)** Genome-wide intron usage is mapped into a low dimensional latent space using a Dirichlet-Multinomial VAE. Visualization of the latent space is done via UMAP. **(c)** A Dirichlet-Multinomial GLM tests for differential splicing across conditions such as predefined cell types or clusters identified from the splicing latent space.

standard dimensionality reduction technique used in the LeafCutter and BRIE2 software packages. The VAE better distinguishes cell types than PCA (Figure 2.3), especially in the

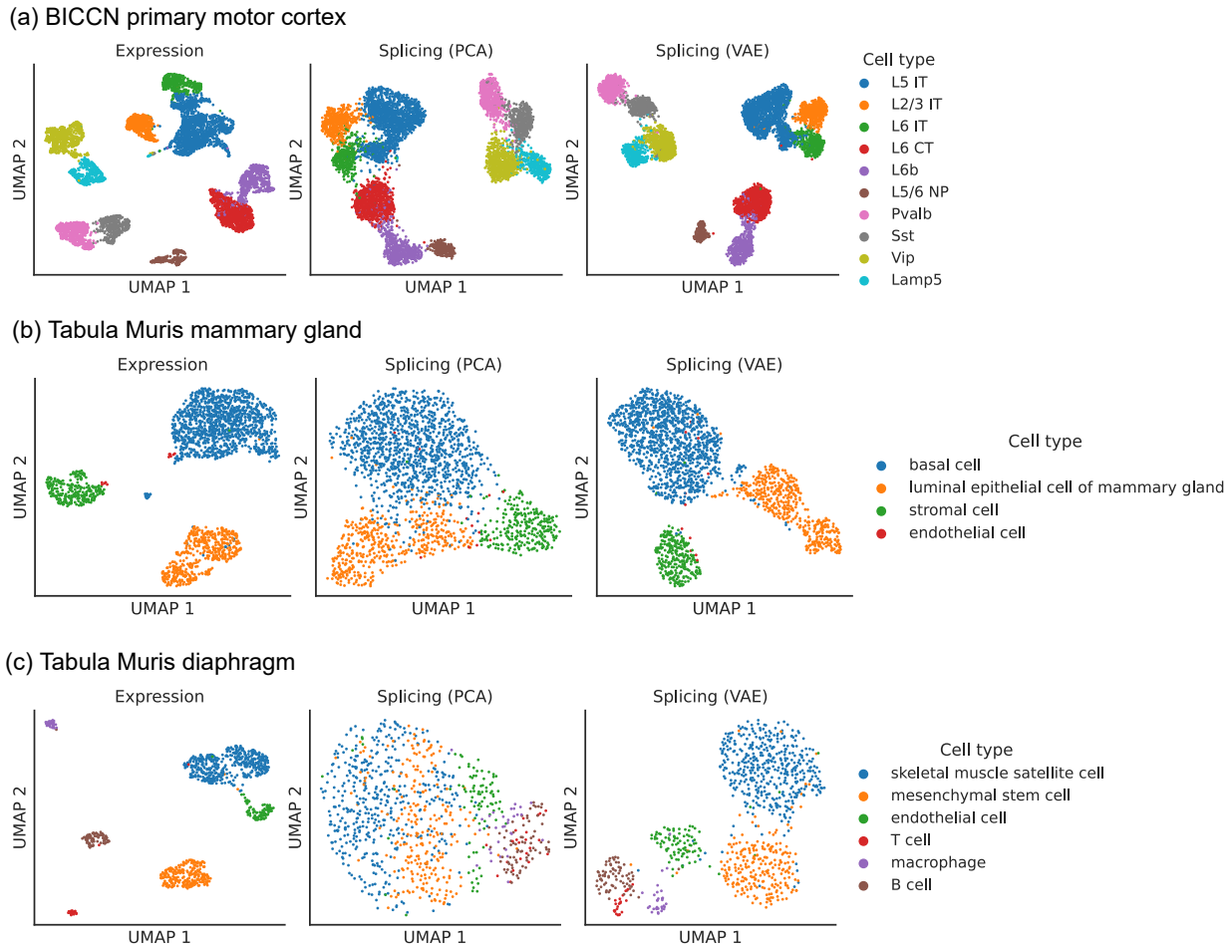


Figure 2.3: **Comparison of splicing latent spaces obtained with PCA and VAE.** Cells from **(a)** the cortex, **(b)** mammary gland and **(c)** diaphragm are projected into a latent space using PCA or VAE and visualized using UMAP. Cell type labels are obtained from the original data sources and are based on clustering in the expression latent space. The VAE is able to better distinguish cell types in the splicing latent space than PCA.

mammary gland and diaphragm.

Differential splicing hypothesis testing with Generalized Linear Model. To test for differential splicing across cell types or conditions, we adopt a Dirichlet-Multinomial Generalized Linear Model (GLM) coupled with a likelihood-ratio test (Figure 2.2c, Materials and Methods). We do so by adapting one of LeafCutter's proposed models for bulk RNA-seq to the scRNA-seq setting and apply it to our Smart-seq2 intron quantification. Namely, due to the sparse nature of scRNA-seq splicing data, we implement a more parsimonious statistical model featuring gene-level rather than intron-level parameters. Furthermore, we adjust the model-fitting algorithm at the initialization and optimization stages (see Materials

and Methods). After our modifications, we obtain well-calibrated p-values whereas those from LeafCutter's original differential splicing model are anti-conservative (Figure A.4) and perhaps prone to extra false positives if applied directly to scRNA-seq data. We also find improvements in computational cost, both in runtime and memory usage (Figure A.4).

As described in Materials and Methods, we generated synthetic data in order to benchmark scQuint against three other methods that also offer two-sample tests for differential transcript usage proportions: BRIE2 and DTUrtle, both designed for scRNA-seq, and LeafCutter, designed for bulk RNA-seq (Figure 2.4). While the choice of an appropriate simulation model for scRNA-seq data is very much an open area of debate, particularly at the transcript level, we attempted to recreate a challenging setting for inference by assuming low coverage (1-2X) and high overdispersion (variance-to-mean ratio of 8). We performed three *in silico* experiments to assess performance under the differing conditions of even transcript coverage, unannotated events, and coverage decay across the transcript. In the case of even coverage, scQuint, LeafCutter, and BRIE2 perform similarly and do a good job of correctly identifying events, while DTUrtle is slightly behind. scQuint does only slightly worse with low cell counts and low coverage, which is probably a trade-off for the robustness that comes from only using reads from junctions sharing 3' acceptor sites. Next, we recreated the unannotated setting by masking the reference given to methods. Only scQuint and LeafCutter are able to perform differential transcript usage testing in this setting, and, as expected, they performed nearly identically to the annotated setting with even coverage. Lastly, we created a setting where transcript coverage decays with distance from the 3' in one of the two groups, mirroring a pattern we often saw in the real data analyzed for this paper. Here, scQuint outperforms the other tested methods by a wide margin with performance improving at higher coverages, unlike other methods. These results validate that scQuint is robust to both incomplete annotations and coverage decay while only paying a modest penalty relative to other methods under ideal conditions (even coverage and annotated events).

Augmenting cell atlases with splicing information

We applied scQuint to two of the largest available Smart-seq2 data sets. The first comprehensively surveys the mouse primary motor cortex (*BICCN Cortex*) [162] while the second contains over 100 cell types distributed across 20 mouse organs (*Tabula Muris*) [117] (Table 2.1). We detect more alternative introns in *BICCN Cortex* neurons than in the entire broad range of cell types present in *Tabula Muris* (which includes neurons but in much smaller number). This observation comports with previous findings that the mammalian brain has exceptionally high levels of alternative splicing [163]. Booeshaghi et al. [16] analyzed *BICCN Cortex* at the transcript level, but focused on changes in absolute transcript expression rather than proportions. While the authors indirectly find some differences in transcript proportions by inspecting genes with no differential expression, this is not a systematic analysis of differential transcript usage. Meanwhile, only microglial cells in *Tabula Muris* [101] have been analyzed at the transcript level. (*Tabula Muris* also contains 10x Chromium data analyzed at the transcript level [106]).

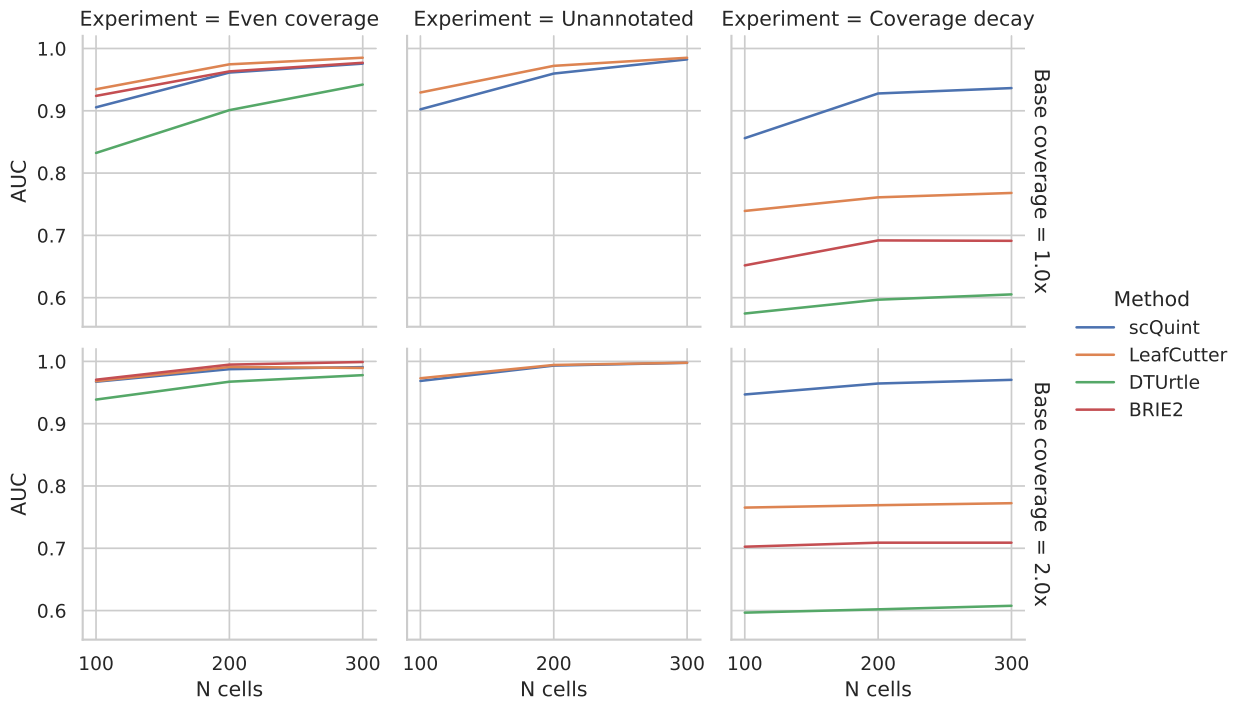


Figure 2.4: **Evaluation of differential splicing test on simulated data.** ROC AUC for detecting differential transcript usage between two groups, based on the p-value produced by different methods. *Unannotated*: the transcript reference given to methods is masked. *Coverage decay*: coverage decay with distance to the 3' end of the transcript is induced in one of the two groups.

Table 2.1: **Overview of analyzed data sets.** Number of cells, tissues, cell types, individuals, detected genes, and detected alternative introns (including the percentage of introns that are not present in the Ensembl reference) for both data sources.

Data set	Cells	Tissues	Cell types	Individuals	Genes	Alt. introns	Unannotated
BICCN Cortex	6220	1	11	45	26488	39357	29%
Tabula Muris	44518	23	117	8	27348	29965	25%

As a community resource, we provide complementary ways to interactively explore splicing patterns present in these data sets (Figure 2.5), available at https://github.com/songlab-cal/scquint-analysis/ with an accompanying tutorial video. The UCSC Genome Browser [69] permits exploration of alternative splicing events within genomic contexts such as amino acid sequence, conservation score, or protein binding sites, while allowing users to select different length scales for examination. We additionally leverage the cell×gene browser [96] (designed for gene expression analysis) to visualize alternative intron PSI (percent spliced-in,

defined as the proportion of reads supporting an intron relative to the total in the intron group) via cell embeddings. Further, one can generate histograms to compare across different groups defined by cell type, gender, or even manually selected groups of cells. These tools remain under active development by the community, and we hope that both the genome- and cell-centric views will soon be integrated into one browser.

Cell-type-specific splicing signal is strong and complementary to gene expression

Primary motor cortex. We first explored the splicing latent space of *BICCN Cortex* cells by comparing it to the usual expression latent space (Figure 2.6a). Cells in the splicing latent space strongly cluster by cell type (annotated by Yao et al. [162] based on gene expression). A similar analysis was recently performed [36] on a different cortex subregion in which most, but not all, neuron subclasses could be distinguished based on splicing profiles (e.g., L6 CT and L6b could not be separated). However, the authors only considered annotated skipped exons, a subset of the events we quantify, and used a different dimensionality reduction technique.

Figure 2.6b (top left) highlights some differentially spliced genes between Glutamatergic and GABAergic neurons, including the glutamate metabotropic receptor *Grm5* as well as *Shisa9/Ckamp44*, which associates with AMPA ionotropic glutamate receptors [146]. The expression pattern of these genes, meanwhile, does not readily distinguish the neuron classes (Figure 2.6b, top right). In *Pgm2*, a gene of the glycolysis pathway thought to be regulated in the developing cortex by mTOR [119], we discover a novel exon preferentially included in Glutamatergic neurons (Figure 2.6c, Figure A.6).

Our differential splicing test reveals thousands of cell-type-specific splicing events (further discussed below in subsection **Comparison of selected tissues**), highlighting marker introns that distinguish neuron subclasses, while the expression of their respective genes does not; e.g., compare bottom left and bottom right panels of Figure 2.6b. Genes that better distinguish cell types at the expression level can be seen in Figure A.5. As another example of the many novel events we discover, we showcase a novel alternative transcription start site in *Rbfox1*, a splicing factor known to regulate cell-type-specific alternative splicing in the brain [148] (Figure 2.6d, Figure A.7). This novel TSS (exon `chr16:5763871-5763913`, intron `Rbfox1_26172`), which lies in a highly-conserved region, is (partially) used by only L6b neurons. We are also able to detect well-known cell-type-specific alternatively spliced genes such as *Nrxn1*, which encodes a key pre-synaptic molecule (Figure 2.6e, Figure A.8) [41]. In this case, we observe an exon (known as splice site 2) exclusively skipped in Vip and Lamp5 neurons.

General patterns in *Tabula Muris*. We next turned our attention to *Tabula Muris*, which comprises a wide variety of organs and cell types from across the entire body. As before, we initially compared the expression and splicing latent spaces using UMAP (Figure 2.7a). This revealed broadly consistent clusters between projections, but a visible shift in the global layout of these clusters. In particular, whereas cell types were better separated in the expression

projection, cell classes (e.g., endothelial, epithelial, immune) formed more coherent clusters in the splicing projection.

To supplement our qualitative comparison of UMAP projections with a more rigorous approach, we built dendrograms and a tanglegram using the respective distances between cells in each of the expression and splicing latent spaces (Figure 2.7b). Despite minor shifts, the dendrograms resemble one another, and most subtree structure is preserved. The low value of their entanglement, a quantitative measure of the discrepancy between hierarchical clusterings, at only 6% indicates a high degree of similarity. (For comparison, the entanglement value between the dendrogram for all expressed genes and that for transcript factors is 11% [117].) As in the UMAP visualization, immune cells group together more closely in the splicing dendrogram. However, unlike the UMAP projection, we observe that several types of pancreatic cells cluster together with neurons, a cell type long believed to share an evolutionary origin [75]. Notably, the left dendrogram in Figure 2.7b shows that hepatocytes are clear outliers in the expression latent space. We suspect this may be due to technical differences from using 96-well plates rather than the 384-well plates used for other cell types.

B cell development in the marrow. We then focused on developing B cells from the bone marrow in *Tabula Muris*. In the splicing latent space, we found that immature B cells are harder to distinguish from the other B cell subpopulations (Figure 2.8a), reflecting less refined splicing programs or limitations in transcript capture efficiency. Immature B cells have also fewer differential splicing events when compared to the other stages of B cell development (Figure 2.8b). The top differential splicing events we identified throughout development displayed splicing trajectories mostly independent from the trajectories of gene expression (Figure 2.8c). We highlight alternative TSSs (one of them novel) in two transcription factors essential for B cell development: *Smarca4*, encoding BRG1 [18] (Figure 2.8d, Figure A.9); and *Foxp1* [57] (Figure 2.8e, Figure A.10). While *Foxp1* expression peaks in pre-B cells and does not follow a monotonic trend over developmental stages, the alternative TSS is progressively included throughout B cell development. Combining gene-level expression with TSS usage, which can influence translation rate, provides a more nuanced characterization of the expression patterns of these important transcription factors. Some other differentially spliced genes with well-known roles in B cell development are *Syk* [31], *Dock10* [43], *Selplg/Psgl-1* [141], and *Rps6ka1* [125].

Epithelial and endothelial cell types across organs. Having compared different cell types within organs, we analyzed putatively similar cell types which are present in multiple organs to investigate splicing variation associated with tissue environment and function. We find many alternative introns with strong PSI differences across epithelial cell types, including several which are novel (Figure 2.9a). Conversely, apart from those in the brain, endothelial cell types fail to display such striking differences (Figure 2.9b). These patterns are consistent with the UMAP projection and dendrogram, both of which suggested less heterogeneity among endothelial than epithelial cells (Figure 2.7).

Our analysis revealed a novel alternative TSS in *Itpr1* (Figure 2.9c, Figure A.12), an

intracellular calcium channel in the endoplasmic reticulum, which regulates secretory activity in epithelial cells of the gastrointestinal tract [79]. This novel TSS yields a shorter protein isoform (full view in Figure A.11) which preserves the transmembrane domain, though it is unclear whether this isoform is functional. Notably, it is the predominant isoform in large intestine secretory cells, and these cells express *Itpr1* at the highest level among all epithelial cell types in the dataset. All nine novel alternative splicing events in Figure 2.9a are alternative TSSs, with four affecting the 5' UTR and five affecting the coding sequence.

Figure 2.9d (PSI distribution in Figure A.13) illustrates a complex alternative splicing event in *Khk* involving the well-studied exons 3a and 3c [55]. Khk catalyzes the conversion of fructose into fructose-1-phosphate, and the two protein isoforms corresponding to either exon 3a or 3c inclusion differ in their thermostability and substrate affinity [6]. While the literature describes these exons as mutually exclusive, the transcriptome reference includes transcripts where neither or both may be included. Although we did not find cell types with high inclusion rates for both exons, we did see multiple cell types where both exons are predominantly excluded, e.g., epithelial cells from the large intestine. Other differentially spliced genes are involved in cellular junctions, which are particularly important in epithelial tissue. These include *Gsn*, *Eps8*, *Tln2*, *Fermt3*, and *Mapre2*.

Comparison of selected tissues. Because of the breadth of the *Tabula Muris* data set, we can look for general trends across a diverse array of tissues and cell types. Table 2.2 summarizes differential expression and splicing for some of the cell types and tissues with the largest sample sizes. First, we note the intersection between the top 100 most differentially expressed and top 100 most differentially spliced genes (ranked by p-value) is consistently low. This means that most differentially spliced genes, which might be of critical importance in a biological system, will go unnoticed if a study only considers differential expression. Second, L5 IT neurons have a larger fraction of genes with differential splicing relative to the number of differentially expressed genes.

We found many more cell-type-specific differential splicing events in the cortex than in the marrow, as expected [163], as well as a higher proportion of events involving novel junctions, which can reach 30% (Figure 2.10a). Differences in proportion of novel junctions should be interpreted with care, however, since they can be affected by sequencing depth and number of cells, both of which vary between the two tissues. Very similar patterns are seen when grouping differential splicing events that occur in the same gene (Figure 2.10b). Most differential splicing events that we detected with alternative introns fall in the coding portion of the gene, with high proportions in the 5' UTR (Figure 2.10c). This is a property of our quantification approach and does not reflect the total number of alternative splicing events in different gene regions; still, the relative proportion can be compared across tissues. We find an increased proportion of differentially spliced non-coding RNA in the cortex, the majority of which are previously unannotated events. To systematically evaluate how well cell types can be distinguished in the expression and splicing latent spaces, we calculated the ROC AUC score for the one-versus-all classification task for each cell type in each tissue using a binary logistic regression model (Figure 2.10d). Since cell type labels were defined

Table 2.2: **Summary of differential expression and splicing for select cell types with the largest sample sizes.** The overlap between the top 100 differentially expressed genes and the top 100 differentially spliced genes is low, indicating that splicing provides complementary information. In addition, L5 IT neurons have a higher ratio of differentially spliced genes to differentially expressed genes than the other cell types. *Diff. spl. genes*: number of differentially spliced genes between the cell type and other cell types in the same tissue. *Diff. exp. genes*: number of differentially expressed genes between the cell type and other cell types in the same tissue. See Materials and Methods for details on the tests for differential splicing and expression.

Tissue	Total # cells	# cell types	Cell type	# cells	Diff. spl. genes	Diff. exp. genes	Ratio	Top-100 overlap
Brain Non-Myeloid	3049	6	Oligodendrocyte	1390	880	8835	0.10	4
Cortex	6220	10	L5 IT	1571	1447	6402	0.23	2
Heart	4144	6	Endothelial cell of coronary artery	1126	465	7108	0.07	5
Large Intestine	3729	5	Enterocyte of epithelium	1112	586	10786	0.05	2
Marrow	4783	10	Hematopoietic stem cell	1363	692	9909	0.07	2

using gene expression values, near-perfect classification is to be expected using the expression latent space. Classification based only on the splicing latent space is very good in general, suggesting that cell-type-specific differential splicing is rather pervasive. A few cell types were more challenging to classify correctly using splicing patterns alone. One such example is immature B cells, a reflection of the lower degree of separation observed in the embedding of Figure 2.8a.

Finding splicing factors associated with specific alternative splicing events

Several splicing factors have been identified as regulators of specific alternative splicing events, but most regulatory interactions remain unknown (see Vuong, Black, and Zheng [147] for a review focused on the brain). To complement expensive and laborious knockout experiments, we sought to generate regulatory hypotheses by analyzing the correlation between splicing outcomes and splicing factor variation across cell types. Focusing on a subset of highly expressed genes in BICCN primary motor cortex neurons, we fit a sparse linear model regressing PSI of skipped exons on both expression and splicing patterns of splicing factors (Figure 2.11a and Figure A.14). Our model recovers several known regulatory interactions such as Khdrbs3/Slm2/T-Star's repression of splice site 4 (SS4) in neurexins, modulating their binding with post-synaptic partners [140]. Additionally, the proportion of a novel alternative TSS (though annotated in the human reference) in *Khdrbs3* (Figure 2.11b, Figure A.15) is

negatively associated with SS4 in *Nrxn1* and *Nrxn3*. This novel isoform lacks the first 30 amino acids of the Qua1 homodimerization domain and could affect dimerization, which modulates RNA affinity [37]. The model also recovers the known regulation of a skipped exon in *Camta1*, a transcription factor required for long-term memory [9], by Rbfox1 [108]. The skipping of exon 5 (E5) of *Grin1*, which controls long-term synaptic potentiation and learning [120], is known to be regulated by Mbnl2 and Rbfox1 [147]. The model associates *Grin1* E5 PSI with the expression of *Rbfox1* but not *Mbnl2*; however, it does suggest an association with the PSI of two skipped exons in *Mbnl2* (Figure 2.11c, Figure A.16, Figure A.17) and further implicates the inclusion level of the novel alternative TSS in *Rbfox1* reported above (`Rbfox1_26172, chr16:5763912-6173605`, Figure 2.6d). These results help clarify the disparate impacts of expression and alternative splicing in splicing factors, and encourage the use of regression models to suggest candidate regulators of cell-type-specific alternative splicing. Such computationally generated hypotheses are particularly valuable for splicing events in splicing factors because of the heightened difficulty to experimentally perturb specific exons rather than whole genes.

2.3 Discussion

In this study, we introduce scQuint, a toolkit for the quantification, visualization, and statistical inference of alternative splicing in full-length scRNA-seq data without the need for annotations. This allows us to successfully extend the analysis of two single-cell atlases to the level of alternative splicing, overcoming the usual technical challenges as well as coverage artifacts and incomplete annotations. Our results, which we make available for public exploration via interactive browsers, indicate the presence of strong cell-type-specific alternative splicing and previously unannotated splicing events across a broad array of cell types. In most cases, splicing variation is able to differentiate cell types just as well as expression levels. We also note a striking lack of overlap between the most strongly differentially expressed and spliced genes (Table 2.2), suggesting that expression and splicing are complementary rather than integrated processes. Moreover, this complementarity may also manifest temporally, as we show in developing B cells in the marrow. Another outstanding question is the functional significance of isoforms, and we find that most differential splice sites appear in the coding sequence with a sizeable minority also mapping to 5' UTRs. The apparent predilection for events to occur in these regions rather than 3' UTRs poses questions about the role of splicing in protein synthesis from translational regulation to the formation of polypeptide chains. Answering these questions requires a more precise understanding of how variation in UTRs and coding sequences affects final protein output as well as the biophysical characteristics of protein isoforms and their roles in different biological systems. These factors, combined with the large fraction of unannotated events in several cell types, should encourage tissue specialists to more deeply consider the contribution of transcript variation to cell identity and cell and tissue homeostasis.

Despite the clear association between splicing and cell identity, our analyses are yet to pro-

duce instances in which clustering in the splicing latent space reveals new cell subpopulations not visible in the expression latent space. This, of course, does not preclude the possibility in other settings where alternative splicing is known to be important, such as in specific developmental transitions or disease conditions. Nevertheless, our current experience leads us to believe that gene expression and splicing proportions provide two different projections of the same underlying cell state. Incidentally, RNA Velocity [73] estimates can be distorted by alternative splicing, and Bergen et al. [14] discuss incorporating isoform proportions into the model as a future direction.

To support our understanding of cell-type-specific splicing, we implemented a regularized generalized linear regression model which exploits the natural variation of splicing factors in different cell types. We recovered a number of previously identified (via knockout experiments) regulatory interactions and propose novel regulatory interactions involving genes known to play important regulatory roles. A key component of our analysis is the decision to include both the expression and alternative splicing patterns of splicing factors as features in the model. Consequently, we infer that several alternative splicing events in splicing factors themselves (some previously unannotated) contribute to their regulatory activity. Our model thus provides several opportunities for follow-up and does so with an increased granularity that distinguishes between effects due to expression and splicing differences. To facilitate further exploration of these data, we have uploaded our results to cell and genome browsers (linked at `https://github.com/songlab-cal/scquint-analysis/`).

Our experience analyzing these large data sets, initially with prior methods and then scQuint, has led to a series of general observations regarding the analysis of splicing in scRNA-seq data. As most analyses use full-length short-read protocols because of the cost of long-read data and the necessary focus on the 3' end of transcripts in most UMI-based techniques, we restrict our attention to the full-length short-read setting and its incumbent challenges. For example, low transcript capture efficiency introduces additional technical noise into isoform quantification [5, 156, 24], and incomplete transcriptome annotations result in discarded reads and reduced sensitivity to cross-cell differences [156]. Nonetheless, we considered several methods (summarized in Table A.1) to analyze transcript variation in short-read, full-length scRNA-seq. We found each of the classes of current methods to be problematic in the context of our data sets for varying reasons. Methods which depend on transcript annotations [22, 111, 59, 58, 159, 155, 86, 60, 134] cannot easily identify unannotated alternative splicing events. In large collections of previously unsurveyed cell types, these may comprise a sizable fraction of events. Indeed, we found up to 30% of differential splicing events were unannotated in certain cell types. Annotation-free approaches are also available, but they either do not provide a formal statistical test for differential transcript usage across conditions [124, 84, 101, 154], or only do so in a specialized manner [92], reducing their potential impacts. Finally, methods' different approaches to quantification are affected by coverage biases to varying degrees. Some methods may thus lead to erroneous inference of cell clusters due to technical rather than biological variation. Until the prevalence and severity of coverage biases are better understood, we advocate quantifying transcript variation in a robust manner.

Recent and future experimental advances will catalyze the study of isoform variation in single cells. For instance, Smart-seq3 [54] allows sequencing of short reads from the entire length of a gene together with unique molecular identifiers, improving mRNA capture and allowing for the filtering of PCR duplicates; however, experiments show that less than 40% of reads can be unambiguously assigned to a single (annotated) isoform. Ultimately, long-read scRNA-seq will provide the definitive picture of isoform variation between cells. Until then, there is much biology to be studied using short-read protocols, and variation at the transcript level should not be disregarded.

2.4 Materials and Methods

Data sets. *Tabula Muris* data [117] have accession code GSE109774. Cells were filtered to those from three month-old mice present in this collection: https://czb-tabula-muris-senis.s3-us-west-2.amazonaws.com/Data-objects/tabula-muris-senis-facs-processed-official-annotations.h5ad (filtering details in [131]). *BICCN Cortex* data [162] were downloaded from https://assets.nemoarchive.org/dat-ch1nqb7 and filtered as in [16].

Simulation. A preliminary set of exon skipping events was obtained by running briekit-event from the BRIE2 software package. For each event, one pair of transcripts was selected if they only differed on the skipped exon, resulting in 561 pairs, each from a different gene. Reads were simulated using Polyester [39], which allows to control overdispersion and induce different kinds of biases. For roughly half of the genes, differential transcript usage (DTU) was induced by overexpressing one transcript 1.5 fold in one of the two conditions. The number of reads was generated using a highly-overdispersed negative binomial distribution, with variance equal to eight times the mean. To simulate coverage decay in one of the conditions, the option bias="cdnaf" was added. To ensure coverage decays as a function of absolute distance to the 3' end of the transcript, reads were generated no farther away from the 3' than the minimum of the lengths of the two alternative transcripts. The Area Under the Receiver Operating Characteristic Curve (ROC AUC) for classifying genes into DTU vs. non-DTU was computed using the p-values from each method, excluding genes that were not tested by a given method (e.g., because of a minimum reads threshold).

Quantification. The bioinformatic pipeline was implemented using Snakemake [72]. Raw reads were trimmed from Smart-seq2 adapters using Cutadapt [90] before mapping to the GRCm38/mm10 genome reference (https://hgdownload.soe.ucsc.edu/goldenPath/mm10/chromosomes/) and the transcriptome reference from Ensembl release 101 (ftp://ftp.ensembl.org/pub/release-101/gtf/mus_musculus/Mus_musculus.GRCm38.101.gtf.gz). Alignment was done using STAR [34] two-pass mode allowing novel junctions as long as they were supported by reads with at least 20 base pair overhang (30 if they are non-canonical) in at least 30 cells. Also, multimapping and duplicate reads were discarded using the flag --bamRemoveDuplicatesType UniqueIdentical (while this can

remove duplicates from the second PCR step of Smart-seq, it will not remove duplicates from the first PCR step). Soft-clipped reads were removed as well. Additionally, reads were discarded if they belonged to the ENCODE region blacklist [2] (downloaded from https://github.com/Boyle-Lab/Blacklist/raw/master/lists/mm10-blacklist.v2.bed.gz).

Gene expression was quantified using featureCounts [82], and total-count normalized such that each cell had 10,000 reads (as in the Scanpy [157] tutorial). Intron usage was quantified using split reads with an overhang of at least 6 base pairs. Introns were discarded if observed in fewer than 30 cells in *BICCN Cortex* or 100 cells in *Tabula Muris*. Introns were grouped into alternative intron groups based on shared 3' splice acceptor sites. Introns not belonging to any alternative intron group were discarded. Additionally, we decided to subset our analysis to introns with at least one of their donor or acceptor sites annotated, so we could assign a gene to them and facilitate interpretation for our specific analyses.

Dimensionality reduction. To run PCA, we worked with alternative intron proportions (PSI, Percent Spliced In) rather than their absolute counts, as the latter would be confounded by gene expression differences. We first introduce some notation:

- c: cell identifier
- g: intron group identifier
- $\vec{y}_g^{(c)}$: vector of counts of introns in intron group g and cell c
- $\text{normalize}(\vec{x}) = \frac{\vec{x}}{sum(\vec{x})}$: function to divide each entry of a vector by the total sum.

Then, PSI can be defined as:

$$\overrightarrow{\text{PSI}}_g^{(c)} = \text{normalize}\left(\vec{y}_g^{(c)}\right)$$

However, given the sparsity of single-cell data, a very high proportion of alternative intron groups will have no reads in a given cell, leaving PSI undefined. More generally, an intron group may contain few reads, resulting in defined but noisy PSI estimates. To navigate this issue, we introduce a form of empirical shrinkage towards a central value. We first define the "global PSI" by aggregating reads from all cells and normalizing. Then, we add this global PSI as a pseudocount vector to each cell before re-normalizing to obtain each cell's shrunken PSI profile (these are non-uniform pseudocounts adding up to one).

$$\overrightarrow{\text{PSI}}_g^{(\text{global})} = \text{normalize}\left(\sum_c \vec{y}_g^{(c)}\right)$$

$$\overrightarrow{\text{SMOOTHED\_PSI}}_g^{(c)} = \text{normalize}\left(\vec{y}_g^{(c)} + \overrightarrow{\text{PSI}}_g^{(\text{global})}\right)$$

We then run standard PCA on the cell-by-intron-smoothed PSI matrix.

The VAE was implemented using PyTorch [105] and scvi-tools [45]. The following is the generative model, repeated for each cell (we drop the superscript indexing the cell in $\vec{z}$, $\vec{p}$, $\vec{y}$ and $\vec{n}$):

Table 2.3: **VAE hyperparameters.**

Data set	Decoder	Layers	σ	Latent dimension
BICCN Cortex	Linear	1	26.8	18
Tabula Muris	Non-linear	2	-	34

1. Sample the latent cell state $\vec{z} \sim \text{Normal}(0, \text{I})$
2. For each intron group g:
   a) Obtain the underlying intron proportions:
   $\vec{p}_g = \text{softmax}(f_g(\vec{z}))$
   b) Sample the intron counts conditioning on the total observed n_g:
   $\vec{y}_g | n_g \sim \text{DirichletMultinomial}\,(n_g, \alpha_g \cdot \vec{p}_g)$

Here f_g, known as the decoder, can be any differentiable function, including linear mappings and neural networks. α_g is a scalar controlling the amount of dispersion. We optimize a variational posterior on cell latent variables $q(z|y)$ (Gaussian with diagonal covariance, given by an encoder neural network) as well as point estimates of global parameters f_g, α_g. The encoder takes as input the smoothed PSI values, as in PCA, but the likelihood is based on the raw intron counts. The objective to maximize is the evidence lower bound (ELBO), consisting of a reconstruction term and a regularization term:

$$\text{ELBO}(y) = \mathbb{E}_{z \sim q(z|y)}[\log p(y|z)] - \text{KL}(q(z|y) \| p(z)),$$

where $\text{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence. Optimization is performed using Adam [70], a stochastic gradient descent method. To avoid overfitting in cases of relatively few cells with respect to the number of features, we considered a linear decoder [129], as well as a $\text{Normal}(0, \sigma)$ prior on the entries of the decoder matrix. Hyperparameters were tuned using reconstruction error on held-out data and are described in Table 2.3.

Differential splicing test. Our differential splicing test across conditions (such as cell types) is based on a modified version of the Dirichlet-Multinomial Generalized Linear Model proposed in LeafCutter [81] for bulk RNA-seq. For each intron group g with L alternative introns:

- $\vec{y}_g$ is a vector of counts for each of the L introns;
- The independent variable, x, equals 0 in one condition and 1 in the other;
- $\vec{a}_g, \vec{b}_g \in \mathbb{R}^{L-1}$ are the intercept and coefficients of the linear model;
- $\alpha_g \in \mathbb{R}$ is a dispersion parameter shared across conditions; and

- the function softmax : $(z_1, \ldots, z_{L-1}) \mapsto \left(\frac{e^{z_1}}{1+\sum_{i=1}^{L-1} e^{z_i}}, \ldots, \frac{e^{z_{L-1}}}{1+\sum_{i=1}^{L-1} e^{z_i}}, \frac{1}{1+\sum_{i=1}^{L-1} e^{z_i}}\right)$ maps from $\mathbb{R}^{L-1}$ to the $(L-1)$-dimensional probability simplex.

The Dirichlet-Multinomial Generalized Linear Model then proceeds as follows:

1. Obtain the underlying intron proportions:
   $\vec{p}_g = \text{softmax}(\vec{a}_g + \vec{b}_g x)$
2. Sample the intron counts conditioned on the total observed, n_g:
   $\vec{y}_g | n_g \sim \text{DirichletMultinomial}\,(n_g, \alpha_g \vec{p}_g)$

We implemented this model in PyTorch and optimized it using L-BFGS [85].

To test for differential splicing across the two conditions, we compare the following two hypotheses:

$$\text{Null hypothesis } H_0\text{: } \vec{b}_g = \vec{0}$$
$$\text{Alternative hypothesis } H_1\text{: } \vec{b}_g \neq \vec{0}$$

We use the likelihood-ratio test, the test statistic for which is asymptotically distributed as a χ^2 random variable with $L-1$ degrees of freedom under H_0. Finally, we correct p-values for multiple testing using the Benjamini-Hochberg FDR procedure [13].

The differences with LeafCutter are the following:

- LeafCutter groups introns that share a 5' donor or 3' acceptor site while scQuint groups introns that share a 3' acceptor site.
- LeafCutter has a vector of concentration parameters, one for each intron, while scQuint uses a single concentration parameter per intron group.
- The LeafCutter and scQuint optimization procedures were implemented separately and differ in initialization strategies as well as L-BFGS hyperparameters.

Latent space analysis. The expression latent space was obtained by running PCA with 40 components on log-transformed and normalized gene expression values. The splicing latent space was obtained by running the VAE on the alternative intron count matrix (or equivalent features, e.g., Kallisto transcript counts, DEXSeq exon counts). Both latent spaces were visualized using UMAP [93]. In the comparison of Figure 2.1, we used our own implementation of the quantifications proposed by ODEGR-NMF, DEXSeq, and DESJ for ease of application to large single-cell datasets.

Dendrograms were constructed using hierarchical clustering (R function hclust) based on euclidean distance between the median latent space embedding of cells of each type. Tanglegram and entanglement were calculated using the dendextend R package, with the step2side method, as also described in Schaum et al. [117].

Reported scores for cell type classification within a tissue were obtained by running a binary logistic regression classifier over different splits of cells into train and test sets. To assess generalization across individuals, we ensured the same individual was not present in both train and test sets.

Cell-type-specific differential splicing. For differential splicing testing between a given cell type and the rest of the tissue, we only considered introns expressed in at least 50 cells and intron groups with at least 50 cells from both of the conditions. We called an intron group "differentially spliced" if it was both statistically significant using a 5% FDR and if it contained an intron with a PSI change greater than 0.05. We considered a differentially spliced intron group as unannotated if it contained an unannotated intron with a PSI change greater than 0.05. Differential expression was performed using the Mann-Whitney test. A gene was considered differentially expressed if it was statistically significant using a 5% FDR and if the fold change was at least 1.5.

For selection of marker genes or introns, we proceeded in a semi-automated fashion. For each cell type, we first filtered to keep only significant genes or introns and then ranked them by effect size. We picked a certain number of genes or introns from the top of this list for each cell type, while ensuring there were no repetitions.

Splicing factor regression analysis. We obtained 75 mouse splicing factors using the Gene Ontology term "alternative mRNA splicing, via spliceosome" (http://amigo.geneontology.org/amigo/term/GO:0000380). A skipped exon annotation, processed by BRIE [59], was downloaded from https://sourceforge.net/projects/brie-rna/files/annotation/mouse/gencode.vM12/SE.most.gff3/download. Instead of using single cells as replicates, we partitioned the BICCN primary motor cortex dataset into roughly 200 clusters of 30 cells each that were pooled to create pseudobulks, aiming to reduce variance in the expression and splicing of splicing factors used as covariates in the model. We filtered target exon skipping events to those defined in at least 95% of the replicates, and those having a PSI standard deviation of at least 0.2. We used log-transformed normalized expression and PSI of alternative splicing events as input features. We chose to keep the PSI of only one intron per intron group to avoid the presence of highly correlated features and improve clarity, even if some information from non-binary events is lost. Input features were filtered to those having standard deviation of at least 0.05, and then standardized. A lasso Dirichlet-Multinomial GLM was fit to the data (in this instance, the model reduces to a Beta-Binomial because skipped exons are binary events), with the sparsity penalty selected via cross-validation. As a first approach, we fit a regular lasso linear regression model on PSI instead of raw counts, resulting in roughly similar patterns in the coefficients. Figure 2.11c shows the coefficients of the lasso Dirichlet-Multinomial model for the top 30 targets with the highest variance explained by the regular lasso model, all above 68%.

Code and data availability. scQuint implementation in Python is available at https://github.com/songlab-cal/scquint. Differential splicing results and access to cell and genome browsers, together with code to reproduce results, are available at https://github.com/songlab-cal/scquint-analysis. Processed alternative intron count matrices are

provided in the AnnData format (anndata.readthedocs.io) for easy manipulation with Scanpy [157], Seurat [126], and other tools.

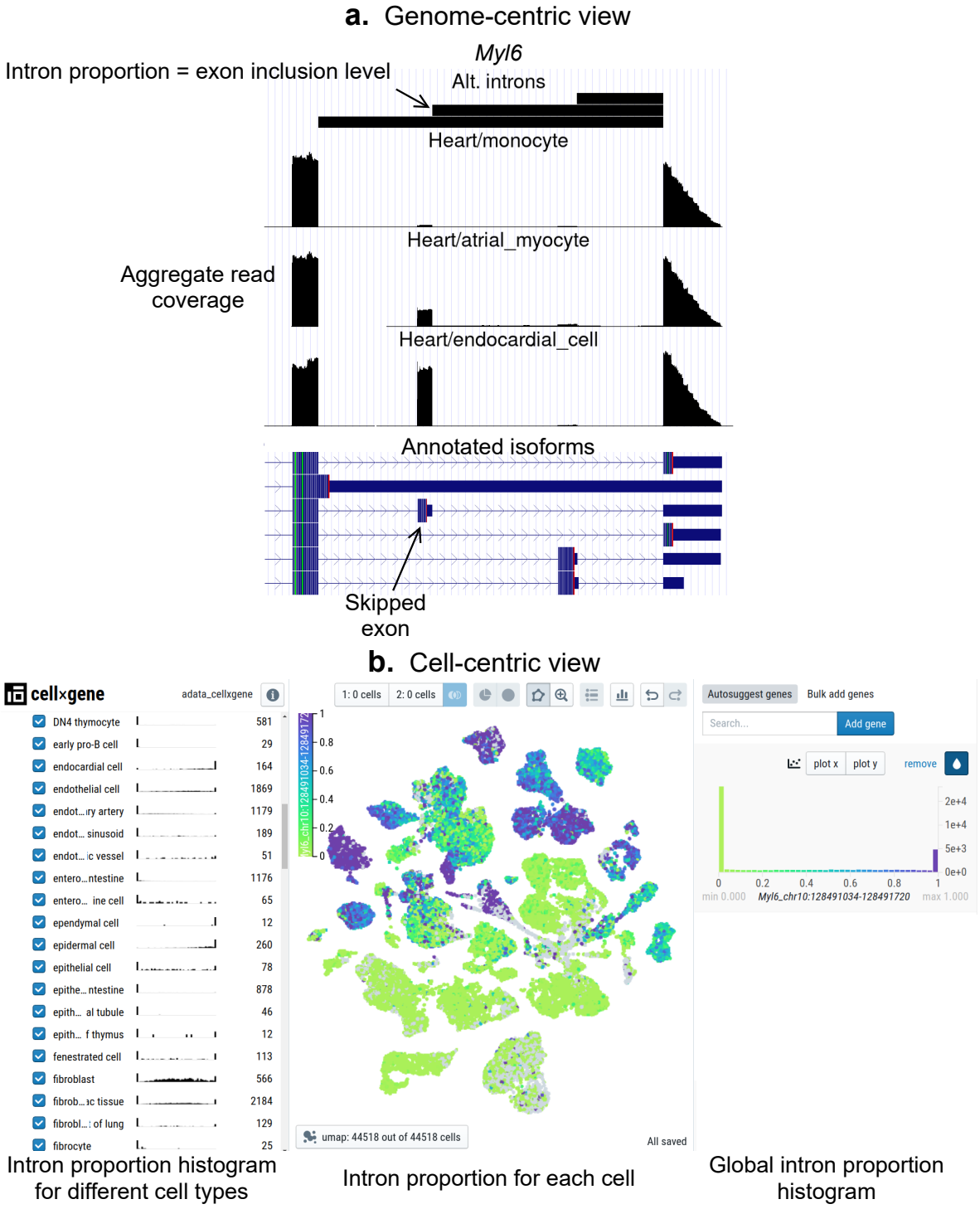


Figure 2.5: **Interactive visualizations of splicing patterns.**

Figure 2.5 *(continued)*: As an example, a skipped exon in *Myl6*. **(a)** The UCSC Genome browser visualization of this locus. Bottom: annotated isoforms of *Myl6*, including a skipped exon. Center: aggregate read coverage in three cell types with varying inclusion levels of the skipped exon. Top: three alternative introns that share a 3' acceptor site. The identified intron's proportion corresponds to the skipped exon's inclusion level. **(b)** cell×gene browser visualization of the marked intron's proportions (`Myl6_chr10:128491034-128491720`). Center: intron proportion for each cell in the UMAP expression embedding. Sides: intron proportion histogram for (left) different cell types and (right) all cells.

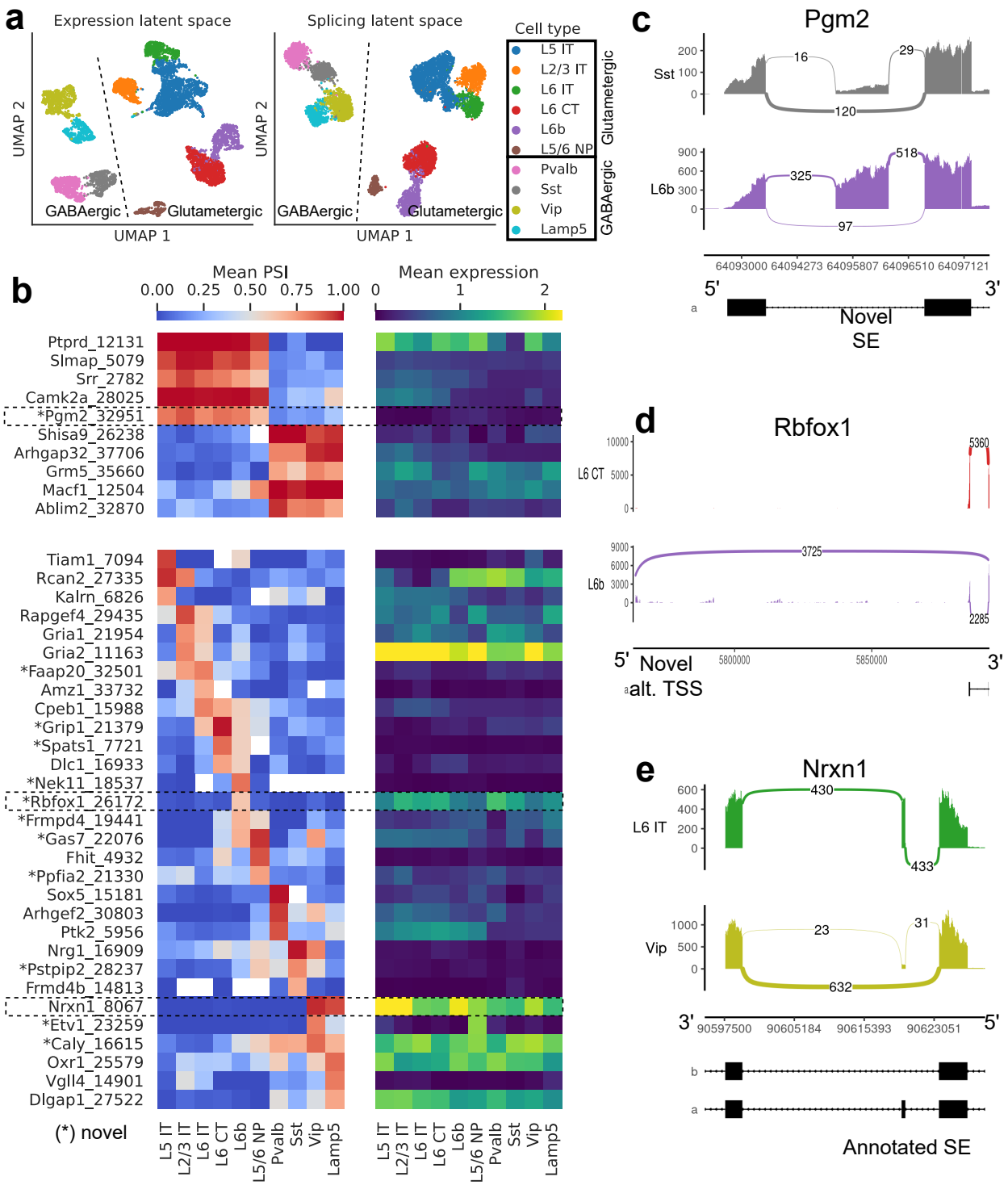


Figure 2.6: **Splicing patterns in *BICCN Cortex*.**

Figure 2.6 *(continued)*: **(a)** Expression and splicing latent spaces, visualized using UMAP. The expression (splicing) latent space is defined by running PCA (VAE) on the gene expression (alternative intron proportion, PSI) matrix. Cell types separate well in both latent spaces. **(b)** PSI of selected introns (left) and expression (log-transformed normalized counts) of their respective genes (right) averaged across cell types. Top: introns distinguishing Glutamatergic and GABAergic neuron classes. Bottom: introns distinguishing neuron subclasses. **(c-e)** Sashimi plots [44] of specific alternative splicing events, displaying overall read coverage with arcs indicating usage of different introns (certain introns are shrunk for better visualization). **(c)** Novel skipped exon in *Pgm2*. **(d)** Novel alternative transcription start site (TSS) in *Rbfox1*. **(e)** Annotated skipped exon (SE) in *Nrxn1*.

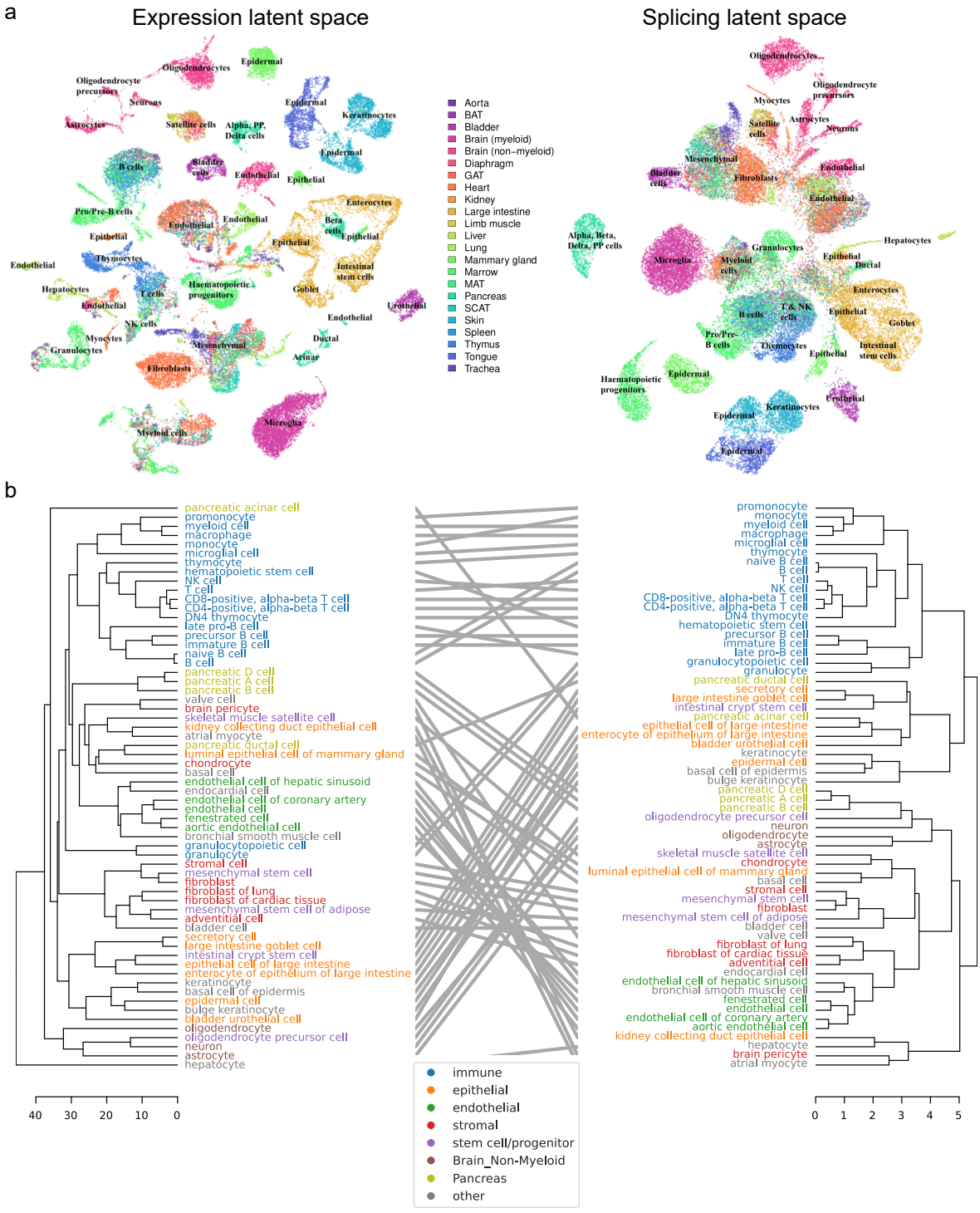


Figure 2.7: **Global analysis of *Tabula Muris*.**

Figure 2.7 *(continued)*: **(a)** UMAP visualization of the expression (left) and splicing (right) latent spaces. Each dot is a cell, colored by organ, and overlays indicate the primary cell type comprising that cluster. **(b)** Tanglegram comparing dendrograms of major cell types based on distances in the expression (left) and splicing (right) latent spaces, highlighting functional classes with specific colors.

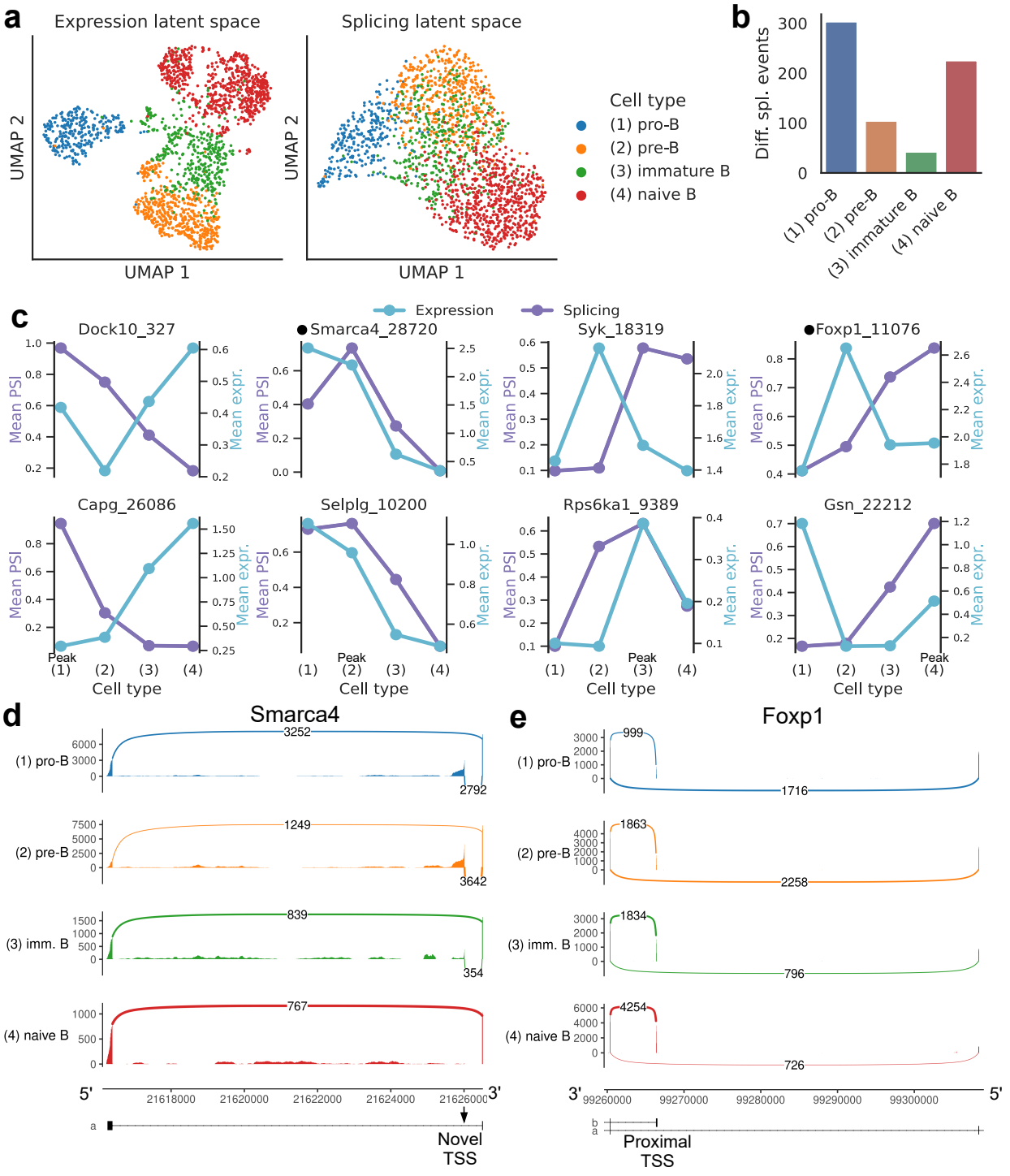


Figure 2.8: **Splicing in developing marrow B cells from *Tabula Muris*.**

Figure 2.8 *(continued)*: B cell developmental stages include pro-B, pre-B, immature B, and naive B. **(a)** Expression versus splicing latent space, as defined previously. In the splicing latent space, some cells types (pro-B) are better distinguished than others (immature B). **(b)** Number of differential splicing events when comparing a B cell stage vs. the rest. **(c)** PSI of some introns that are differentially spliced throughout development, together with expression of the respective genes (log-transformed normalized counts). Expression and splicing can have very different trajectories. **(d)** Sashimi plot of novel alternative transcription start site (TSS) in *Smarca4*. The novel TSS has maximum usage in pre-B cells, and then decays, while the expression peaks at pro-B cells. **(e)** Sashimi plot of an annotated alternative TSS in *Foxp1*. The proximal TSS in increasingly used as development progresses, while the expression peaks at pre-B cells.

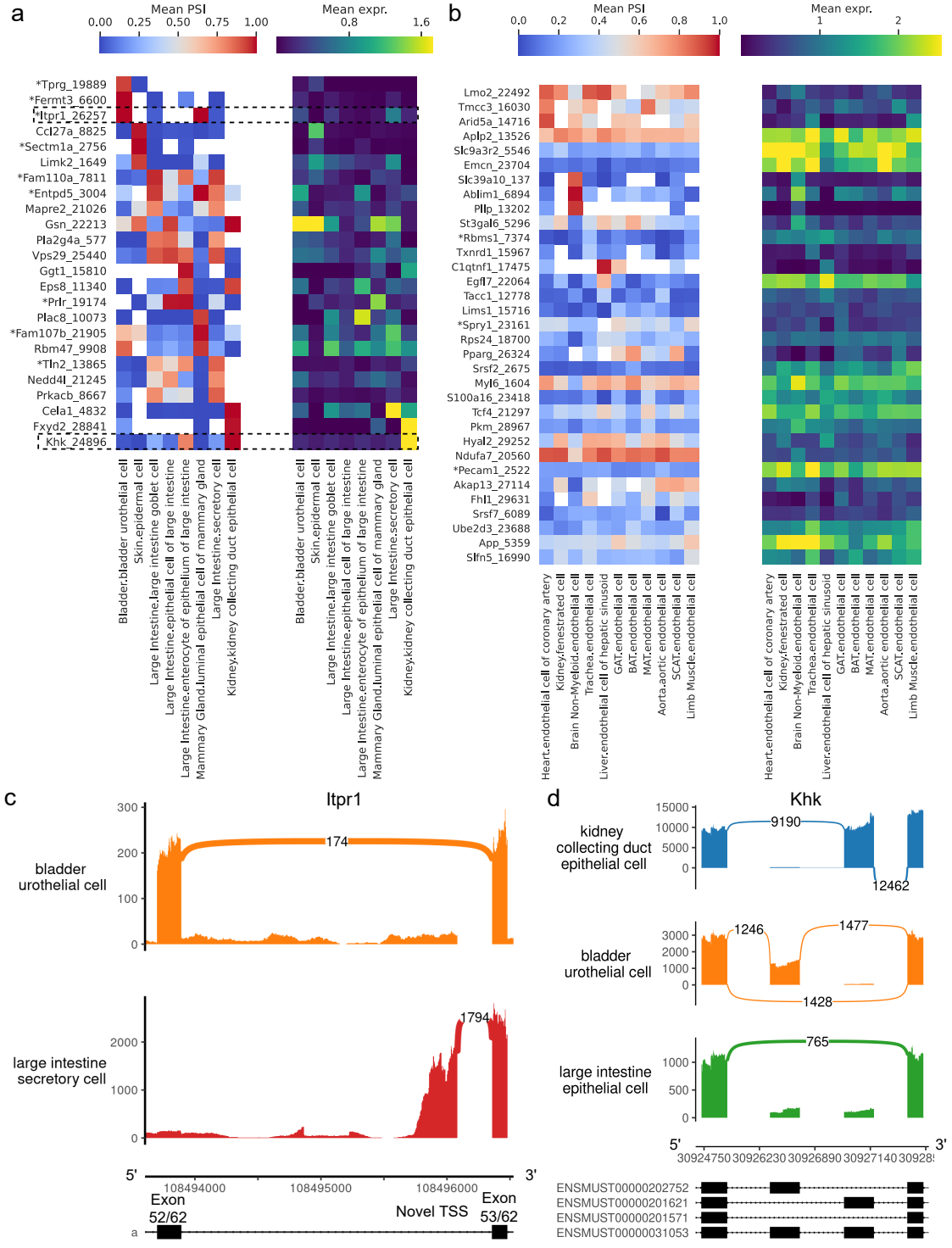


Figure 2.9: **Alternative splicing patterns across epithelial and endothelial cell types.**

Figure 2.9 *(continued)*: **(a-b)** PSI of selected introns (left) and expression (log-transformed normalized counts) of the corresponding genes (right) averaged across cell types. Novel intron groups are marked with (*). **(a)** Introns distinguishing epithelial cell types. **(b)** Introns distinguishing endothelial cell types. **(c)** Sashimi plot of an alternative TSS in *Itpr1*. **(d)** Sashimi plot of a complex alternative splicing event in *Khk*.

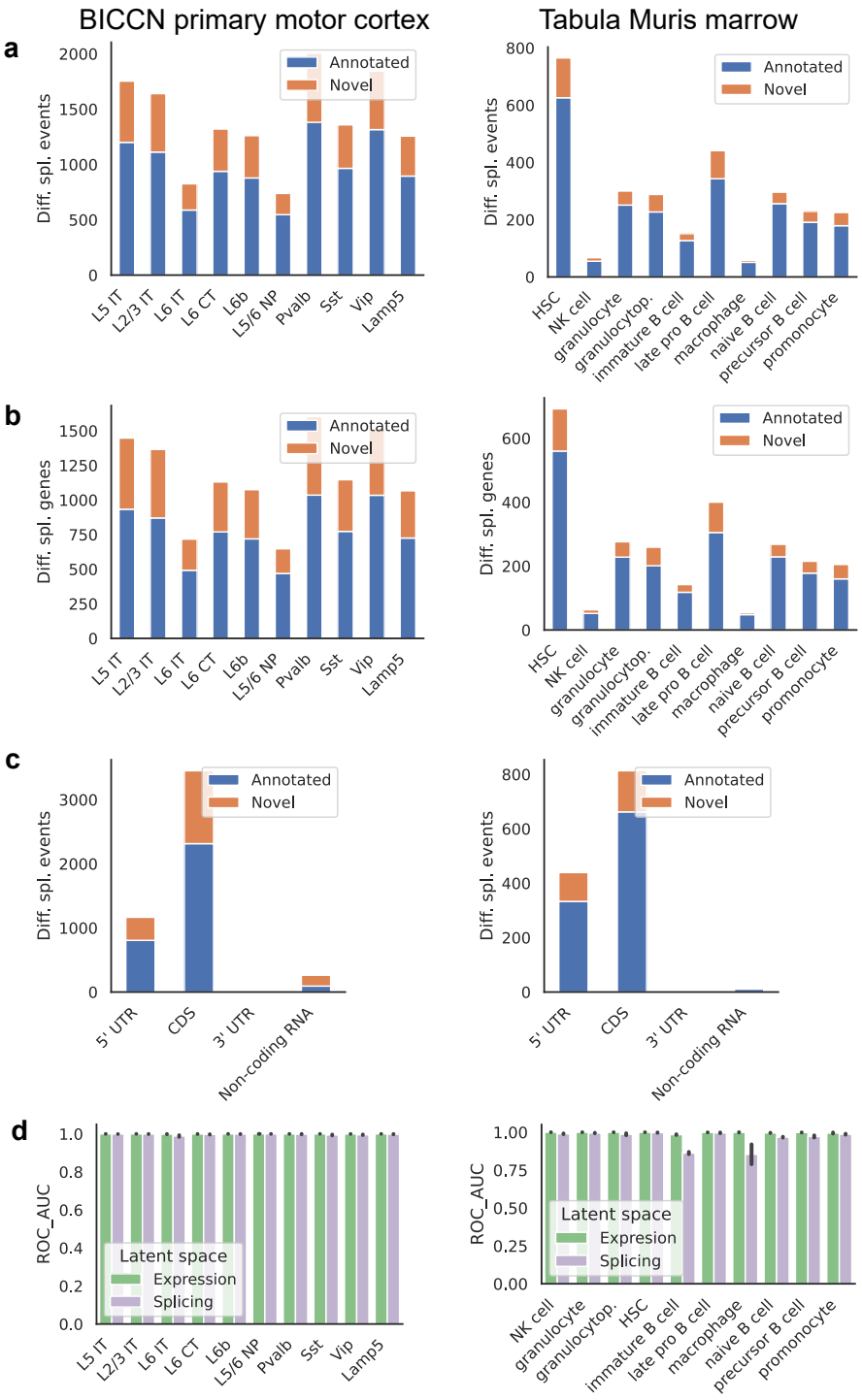


Figure 2.10: **Patterns across tissues.**

Figure 2.10 *(continued)*: **(a)** Number of differential splicing events detected in each cell type. Cortex cell types have more differential splicing events and larger proportions of novel events (those involving an intron absent from the reference). **(b)** Number of genes with a detected differential splicing event, for different cell types. **(c)** Number of differential splicing events in different gene regions aggregated over cell types (duplicate events removed). Cortex cell types have higher proportions of events in coding regions and non-coding RNAs. Note: y-axes are not on the same scale. **(d)** ROC AUC score for classification of each cell type versus the rest based on either the expression or splicing latent space, using logistic regression, training and testing in non-overlapping sets of individuals. The score for splicing-based classification is near-perfect in most cell types with some exceptions such as immature B cells in the marrow.

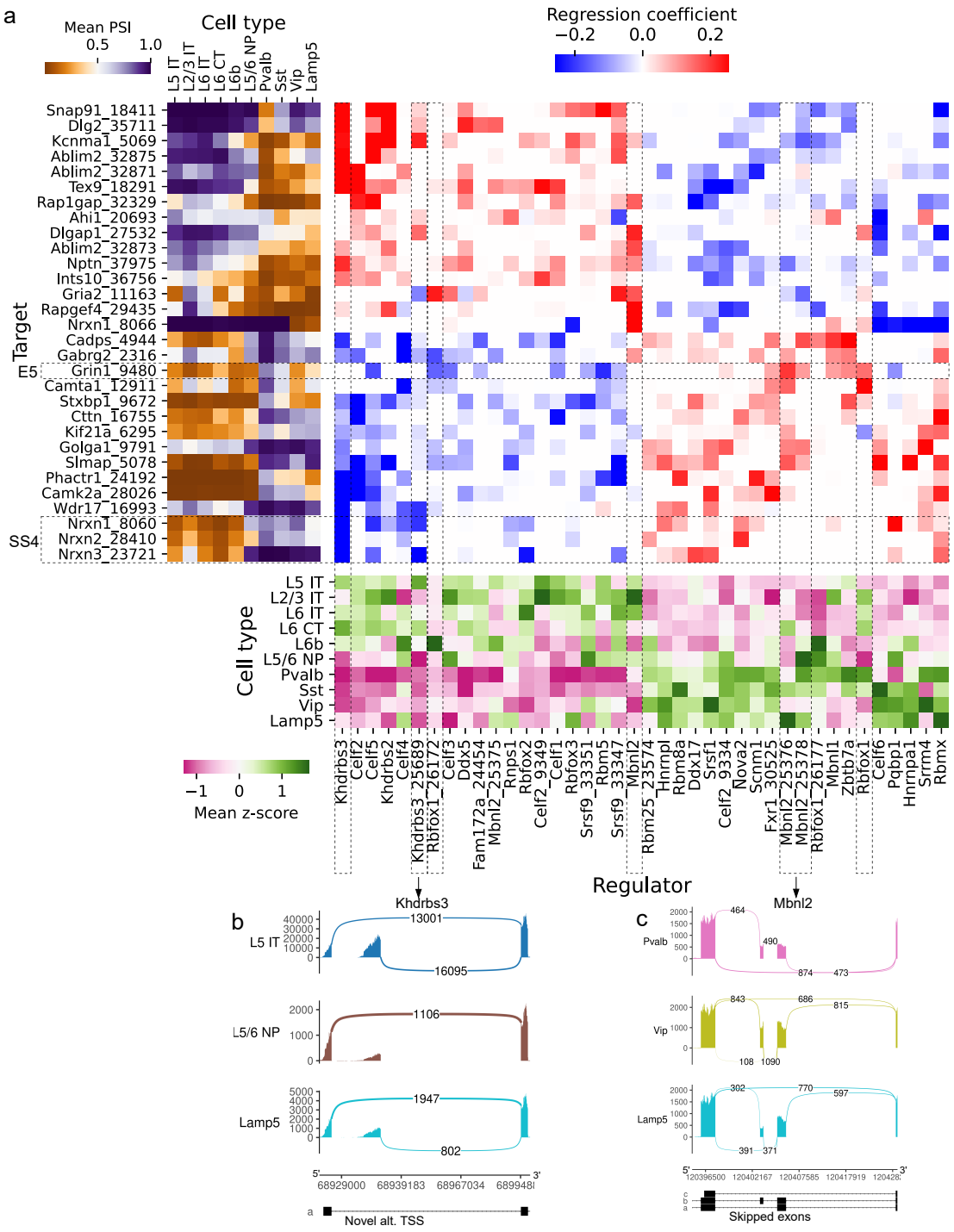


Figure 2.11: **Associations between splicing factors and alternative splicing.**

Figure 2.11 *(continued)*: **(a)** Regression analysis of exon skipping based on expression and splicing of splicing factors, using the BICCN mouse primary motor cortex dataset. Left panel: mean PSI of skipped exons across cell types. Bottom panel: mean z-scores of selected splicing factor features across cell types, including whole-gene expression (gene name) and PSI of alternative introns (gene name and numerical identifier). Center panel: regression coefficients (log-odds) of each splicing factor feature used to predict skipped exon PSI in our sparse Dirichlet-Multinomial linear model. **(b)** Novel alternative TSS in *Khdrbs3*. **(c)** Annotated skipped exons in *Mbnl2*.

Chapter 3

DNA language models are powerful predictors of genome-wide variant effects

This is joint work with Sanjit Singh Batra and Yun S. Song, published in *PNAS* [10]. I would like to thank Carlos Albors, Jesús Martínez-Gómez, Eyes Robson, Nilah Ioannidis and Allison Gaudinier for helpful discussions.

3.1 Introduction

The emergence of genome-wide association studies (GWAS) has significantly enhanced our ability to examine the genetic basis of complex traits and diseases in both humans and plants. In humans, GWAS have played a crucial role in identifying genetic variants associated with a range of traits, including schizophrenia and obesity [144]. Similarly, in plants, GWAS have shed light on the genetic factors influencing traits such as drought tolerance, disease resistance, and yield [136]. A central challenge in GWAS is pinpointing causal variants for a trait, as linkage disequilibrium (LD) can lead to spurious associations [20]. This process, known as fine-mapping, serves as a foundation for constructing accurate, portable polygenic risk scores and understanding the underlying biological mechanisms. Although experimental validation of causal variants is the gold standard, it is not scalable. Instead, a scalable fine-mapping strategy involves utilizing computational variant effect predictors [153], which vary from conservation scores to deep learning models trained on functional genomics data. Accurate variant effect prediction is also vital for diagnosing rare diseases and interpreting rare variants that lie beyond the scope of traditional GWAS [91].

Recently, state-of-the-art performance in predicting the effects of missense (coding) variants has been achieved by training unsupervised models on extensive protein sequence databases [97] or their corresponding multiple sequence alignments [40]. These large language models can predict missense variant effects in an unsupervised manner, without the need

for additional training on labeled data. This progress has been driven by advancements in natural language processing, where significant strides have been made by pre-training language models on vast text corpora. Pre-trained models such as BERT can be fine-tuned for downstream tasks such as sentiment analysis [33]. More recently, language models like GPT-4 have demonstrated impressive leaps in test performance across various disciplines, from law to computer science [23].

A widely-used approach to interpreting non-coding variant effects involves training a supervised model to predict functional genomics data — such as chromatin accessibility, transcription factor binding, or gene expression — and then evaluating variants based on how they disrupt these predictions. This approach was first introduced by DeepSEA [169], which utilized 919 functional genomics tracks, and has since been refined by Enformer [7] with 6,956 tracks and Sei [27] with 21,907 tracks. However, this approach's success depends on the availability of high-quality functional genomics data from a diverse array of cell types, which can be prohibitively expensive to generate for most species. Certain models focus on specific classes of non-coding variants. For instance, classifiers trained solely on sequence data can predict the impact of intron variants on splicing patterns [61, 29]. To evaluate the effects of regulatory variants, Lee *et al.* [77] developed a support vector machine that distinguishes putative regulatory sequences from random genomic sequences. More recently, a deep learning model capable of predicting Hi-C signal from sequence data demonstrated its potential to predict the impact of regulatory variants on DNA folding within the nucleus [42]. Additionally, a deep learning model [166] was successfully trained to predict DNA methylation levels of CpG sites from sequence data, enabling the prediction of non-coding variant effects on DNA methylation.

However, variant type-specific models may not be well-suited for detecting trait-associated rare variants, fine-mapping, or calculating polygenic scores, as these tasks are facilitated by the comparison of genome-wide variants all together. For instance, a model that is exclusively designed for either missense or regulatory variants would not be able to prioritize between a *de novo* missense variant and a *de novo* promoter variant observed in an individual with a rare disease. An important class of genome-wide scores are conservation scores such as phyloP [110] and phastCons [122], which are computed from genome-wide alignment of multiple species. Since these do not require functional genomics data, they have been widely applied to many systems, including non-model organisms [135]. In humans, CADD is another important genome-wide variant effect predictor that combines conservation and functional genomics annotations, and is trained to distinguish between an inferred set of putative benign and putative pathogenic variants [115, 112].

In this paper, we introduce the **G**enomic **P**re-trained **N**etwork (**GPN**), a multi-species DNA language model trained using self-supervision. While existing DNA language models [164, 63, 98, 161, 56, 52, 8] have not yet demonstrated the ability to make accurate variant effect predictions based on self-supervision alone, GPN presents a unified approach capable of accurate unsupervised prediction of genome-wide variant effects. We demonstrate its utility by achieving state-of-the-art performance in *Arabidopsis thaliana*, a model organism for plant biology closely related to many agriculturally important species, as well as a source of insight

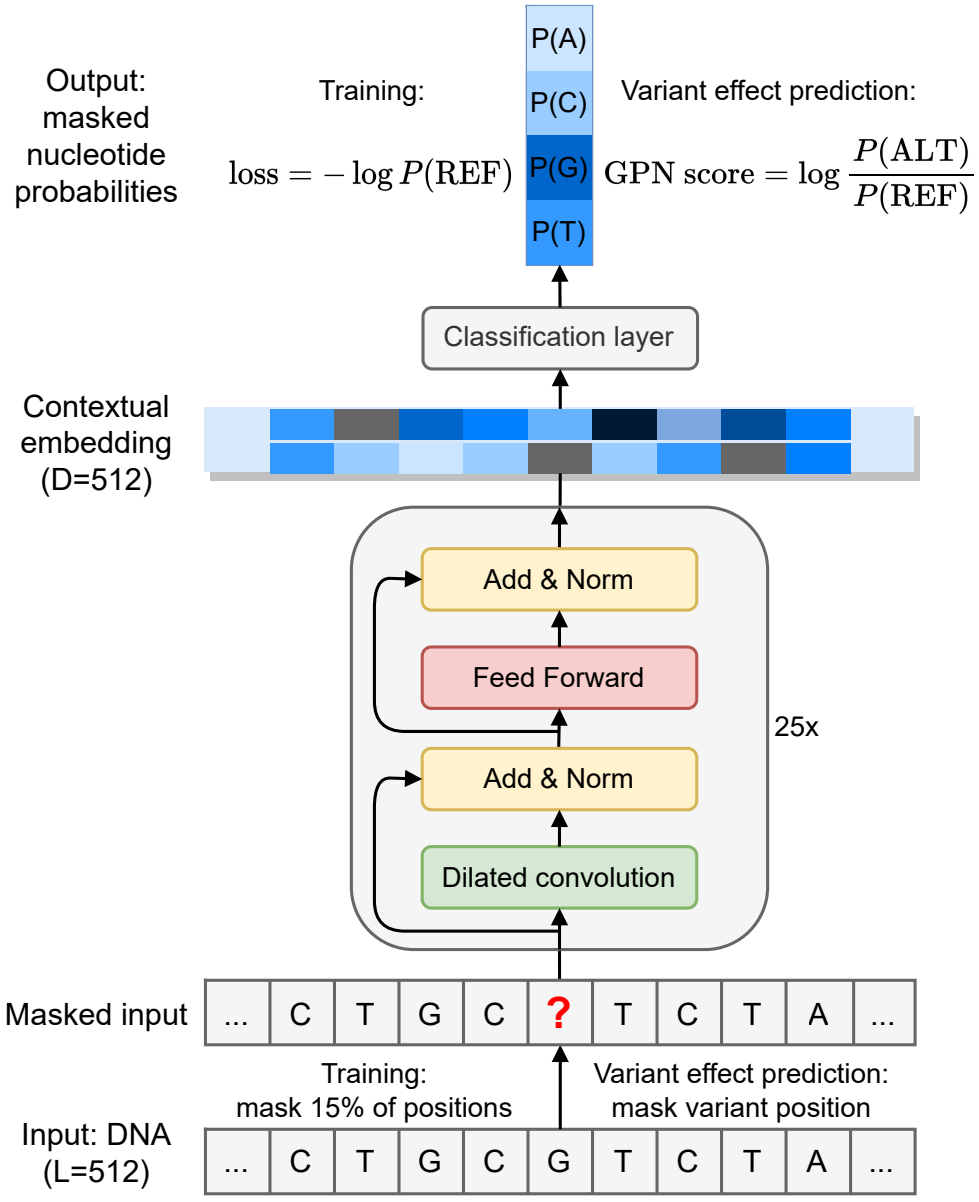


Figure 3.1: **Overview of GPN (Genomic Pre-trained Network).** The input is a 512 bp DNA sequence where certain positions have been masked, and the goal is to predict the nucleotides at the masked positions. During training, 15% of the positions are masked. During variant effect prediction, only the variant position is masked. The sequence is processed through a convolutional neural network resulting in a high-dimensional contextual embedding of each position. Then, a final layer outputs four nucleotide probabilities at each masked position. The model is trained on the reference sequence with the cross-entropy loss. The GPN variant effect prediction score is defined as the log-likelihood ratio between the alternate and reference allele. L: window length in base pairs. D: embedding dimension. REF: reference allele. ALT: alternate allele.

into human diseases [65]. Moreover, GPN outperforms genome-wide conservation scores such as phyloP and PhastCons, which rely on whole-genome alignments of 18 closely related species [135]. GPN's internal representation of DNA sequences can distinguish genomic regions like introns, untranslated regions, and coding sequences. Additionally, the confidence of GPN's predictions can help reveal regulatory grammar, such as transcription factor binding motifs. Our results lay the foundation for developing state-of-the-art genome-wide variant effect predictors for any species using genomic sequence alone, which can be readily integrated into GWAS fine-mapping and polygenic risk scores.

3.2 Results

Training a multi-species DNA language model. We used *unaligned* reference genomes from *Arabidopsis thaliana* and seven related species within the Brassicales order to pre-train a language model based on a convolutional neural network (Table B.1). This model was designed to predict masked nucleotides conditioned on their local genomic context (Figure 3.1, *Materials and Methods*). During the training process, we encountered challenges with repetitive elements, which can be functionally significant but are heavily overrepresented in the genomes [19]. We found that reducing the weight of prediction loss for repetitive regions led to lower test perplexity in non-repetitive regions, which are often of greater interest (Table B.2). Compared to full down-weighting, moderate down-weighting results in a similar improvement in perplexity for non-repetitive regions without sacrificing genome-wide perplexity as much. Consequently, we focus on this model throughout the remainder of the paper unless otherwise specified.

Unsupervised clustering of genomic regions. To understand how well the model has learned the structure of the genome, we averaged GPN's contextual embeddings (512 dimensions) of nucleotides over 100 base pair (bp) windows from the reference genome and visualized them using UMAP [93] (Figure 3.2a). Notably, GPN, trained without any supervision, has learned to distinguish genomic regions such as intergenic, introns, coding sequences (CDS), untranslated regions (UTR) and non-coding RNA (ncRNA). To quantify GPN's ability to distinguish genomic regions, we trained a logistic regression classifier using the averaged embeddings as features, achieving the highest accuracy on CDS (96%) and the lowest on ncRNA (51%), the least frequent class. As summarized in Figure 3.2b, the highest confusion was observed between intergenic regions and ncRNAs; this may be partly explained by errors in ncRNA annotation, which is especially challenging given their low expression levels and poor conservation [88]. This level of classification accuracy cannot be achieved merely through k-mer frequencies ($k = 3$: 8% to 70%; $k = 6$: 15% to 67%; see Figure B.1). We also note that, to some extent, GPN embeddings can distinguish different repeat families (Figure B.2).

DNA motifs revealed by high-confidence model predictions. To further understand GPN, we individually masked each position in the genome and obtained the model output

distribution over nucleotides, given its context. To facilitate utilizing these predicted distributions, we created sequence logos that can be visualized in the UCSC Genome Browser [69, 99] (https://genome.ucsc.edu/s/gbenegas/gpn-arabidopsis), where the height of each letter is proportional to its probability, and the overall height is given by the information content, measured in bits [118] (see Figure 3.3a for an example). The model's prediction confidence correlates with the expected functionality of the sites. For example, exonic positions are predicted with higher confidence than the surrounding introns, except for the canonical splice acceptor and donor dinucleotide motifs. Similarly, within codons, the third nucleotide position (CDS3), which usually does not affect amino acid identity, is generally predicted with lower confidence than the first two positions (CDS1, CDS2). Start and stop codon motifs are also generally well predicted (examples in Figure B.3). Across a 1 Mb region in the test chromosome (containing 264 genes and 471 transcripts), model perplexities in splice donors (median = 1.02), splice acceptors (median = 1.03), start codons (median = 1.08), CDS2 (median = 2.24), CDS1 (median = 2.44), CDS3 (median = 2.79), and stop codons (median = 2.8) are significantly smaller than those in intergenic and intronic regions (median = 3.24, all Mann–Whitney p-values $< 10^{-17}$, Figure B.4). Perplexity in CDS2 is significantly smaller than that in CDS1, which in turn is significantly smaller than that in CDS3 (all Mann–Whitney p-values $< 10^{-300}$), consistent with their different expected levels of constraint [110].

We hypothesized that scanning promoters for small regions of high-confidence GPN predictions could help identify transcription factor binding sites. To achieve this, we adapted TF-MoDISco [121], a tool for *de novo* discovery of transcription factor binding sites using supervised models. This tool clusters high-scoring regions into motifs and compares them to databases of known motifs. Applying the adapted TF-MoDISco to GPN scores in promoter regions, we discovered approximately a hundred and sixty motifs (Figure B.5), with four examples shown in Figure 3.3b, the first two having a significant match in PlantTFDB [135] (with q-value < 0.05 in Tomtom [51]). Some of the discovered motifs are well-documented in the literature but do not have a significant match in this database, such as the third motif [35] in Figure 3.3b. Some motifs could represent novel promoter elements, like the fourth motif, which is palindromic with symmetrical entropies, suggesting that it could potentially form RNA or DNA alternative secondary structure [130].

Unsupervised variant effect prediction. GPN can be employed to calculate a pathogenicity or functionality score for any single-nucleotide polymorphism (SNP) in the genome using the log-likelihood ratio between the alternate and reference allele (GPN score, Figure 3.1). Visually, this involves comparing the heights of the letters in the logo plot (Figure 3.3a).

In silico mutagenesis. We first computed GPN scores for *in silico* mutagenesis of SNPs within a 1 Mb region and aggregated the results across variant types (Figure 3.4). The ranking of variant types based on the lowest percentile of GPN scores is generally consistent with established notions of deleteriousness [94][1]. For example, the four lowest scored variant types

[1] https://useast.ensembl.org/info/genome/variation/prediction/predicted_data.html

are splice donor, splice acceptor, stop gained and start lost variants, which significantly disrupt the open reading frame. As expected, missense variants are predicted to have a bigger impact than synonymous variants. However, we observed that some variants within repetitive elements were assigned rather low GPN scores, ranking close to missense variants. Furthermore, the proportion of low GPN scores for repeat variants depends on the training loss weight on repeats (Figure B.6a). More precisely, in models with 0.0 and 0.1 down-weighting, respectively, 8% and 9% of repeat variants are ranked before the first decile of missense variants. These represent a substantial decrease compared to the 27% observed in the model without any down-weighting (Figure B.6b, Fisher's exact test $p < 10^{-300}$).

Benchmarking using allele frequencies in 1001 Genomes. Following our *in silico* mutagenesis experiments, we analyzed over 10 million SNPs from naturally occurring accessions of the 1001 Genomes Project [1]. While most variants have a neutral GPN score, there is a heavy tail of putative functional variants with negative GPN scores (Figure 3.5a). Notably, variants with lower GPN scores are, on average, less frequent in the population, suggesting they could be under purifying selection (Figure 3.5b, full distribution in Figure B.7). To evaluate the capability of identifying putative functional variants, we assessed the enrichment of rare versus common variants in the tail of genome-wide score distributions. Putative functional SNPs, defined as the lowest 0.1% of GPN scores, exhibit a 5.5-fold enrichment in rare variants (Figure 3.5c); see Figure B.8 for different allele frequency thresholds. GPN outperforms other genome-wide variant effect predictors for *Arabidopsis*, specifically phyloP and phastCons, which are conservation scores derived from a broader set of 18 Brassicales species (Figure 3.5d). In fact, GPN scores are only weakly correlated with phyloP ($r = 0.22$, $p < 10^{-300}$) and phastCons ($r = 0.13$, $p < 10^{-300}$). We also considered the alternative abs(phyloP) (the absolute value of phyloP), but it did not achieve a significant enrichment. A notable advantage of GPN is that it is able to score variants that could not be scored by phyloP and phastCons due to unsuccessful whole-genome alignment (14.2% of all variants). GPN performs comparably to phyloP and phastCons when using less stringent thresholds for defining putative functional SNPs (Figure B.9), indicating its particular strength in detecting deleterious variants at the extreme tail. GPN also achieves significant odds ratios when computed only within particular variant classes, but its performance relative to phyloP and phastCons varies (Figure B.9). On a separate note, a slightly higher odds ratio is achieved by the GPN model trained with an intermediate loss weight on repeats (Figure B.6c). The model trained on only a single species performs substantially worse (Figure B.10a).

Enrichment of GWAS hits in regions with low GPN scores. In our pursuit to further evaluate the efficacy of GPN, we examined the AraGWAS Catalog [137], a comprehensive database of genome-wide association studies (GWAS) in *Arabidopsis thaliana*. We hypothesized that GWAS hits may be enriched in regions with low GPN scores. An advantage of GPN is that it can give substantially different scores to variants in strong linkage disequilibrium (LD) with each other, if their surrounding contexts are different (e.g., see Figure 3.6a, top). In contrast, the standard GWAS would give similar scores to such variants; in particular, neutral variants in strong LD with a functional variant would also be associated with a trait. To account for

this difference, we devised a new score, GPN×LD, which weighs GPN scores by LD (*Materials and Methods*). With this approach, GPN×LD effectively distinguishes GWAS hits from non-hits in this example locus (Figure 3.6a, bottom). More generally across the genome and all traits, the tail of GPN×LD scores is greatly enriched in GWAS hits, much more so than the tail of raw GPN scores (Figure 3.6b). In particular, by analyzing odds ratios (Figure 3.6c), we found that SNPs with the lower 1% of GPN×LD scores are 10.3-fold enriched in GWAS hits compared to the upper 99% of GPN×LD scores, while less than 7.5-fold enrichment was observed for other methods (Figure 3.6d); see Figure B.11 for different thresholds. Using the Bonferroni correction instead of the permutation-based significance threshold recommended by AraGWAS [138] yields lower odds ratios for all methods, but GPN×LD still achieves the highest enrichment (Figure B.12). Interestingly, the GPN model trained with an intermediate loss weight on repeats achieves the best performance (Figure B.6d). The model trained on only a single species performs worse (Figure B.10b). Furthermore, GPN×LD achieves much higher odds ratios when considering the full variant set, including regions that do not align to other Brassicales (Figure 3.6e); failed alignment could be partly due to genomic rearrangements that may be potentially associated with local adaptation in *Arabidopsis thaliana* [67].

3.3 Discussion

Here we present the first unsupervised genome-wide variant effect predictor based on unsupervised pre-training of DNA language models. We demonstrate that GPN outperforms other genome-wide variant effect predictors in *Arabidopsis thaliana*, a model species for plant biology. Since GPN is trained only on DNA sequence, it can be readily applied to understudied non-model organisms even in the absence of extensive functional genomics data, while still providing state-of-the-art unsupervised variant effect prediction genome-wide.

We can think of GPN as a generalized conservation score. Similar to phyloP and phastCons, GPN is genome-wide, can be trained on genomic sequence alone, and is cell-type and mechanism agnostic [128]. The key distinction is that while phyloP and phastCons only consider nucleotide frequencies at a specific site, GPN can learn from joint nucleotide distributions across all similar contexts appearing in the genome. Furthermore, GPN does not rely on whole-genome alignments, which can often have a lower quality in non-coding regions.

The capability of GPN to score genome-wide variants on a unified scale renders it ideal for integration into rare disease diagnosis, fine-mapping, and polygenic risk scores, including burden tests. The separation of genomic regions based on GPN embeddings suggests that it could be further fine-tuned for *de novo* genome annotation. Combining GPN predictions with TF-MoDISco offers a promising strategy for discovering functional motifs. Although in this study we focused on transcription factor binding sites, we believe that GPN predictions around splice junctions could also facilitate the identification of splicing factor binding sites.

Repetitive elements, which are inherent components of eukaryotic genomes, pose several

challenges that have been underexplored in DNA language modeling studies. First, these elements are significantly over-represented [19]. The lower perplexity in non-repetitive regions upon down-weighting repeats can be attributed to the model allocating fewer parameters exclusively to repetitive elements. Second, repetitive elements display reduced sequence variation compared to other regions, in particular younger repeats with little time to accumulate mutations [170]. We believe that these factors together may cause differences in model likelihoods in these regions to be less clearly associated with differences in fitness. Our proposed down-weighting of repeats only partially mitigates these issues, and we encourage further investigation by the scientific community. Potential research directions include examining the effects of down-weighting repeats based on their respective families or inferred age.

While the current implementation of GPN achieves state-of-the-art variant effect prediction for *Arabidopsis thaliana*, there is room for improving its training scheme. Mounting evidence suggests that larger models and more extensive training data can enhance performance [68]. Our current proof-of-concept model is considerably smaller — by 200 times — than the largest published protein language model [83]. One strategy to improve GPN, inspired by protein modeling, involves explicitly incorporating multiple sequence alignments [113, 66]. However, this enhancement will be bottle-necked by the quality of alignment in non-coding genome regions. Other promising avenues for DNA language modeling include incorporating DNA-specific inductive biases, such as reverse-complement equivariance [167], as opposed to our current method of averaging model outputs for both strands during testing. Additionally, integrating long-range information using recent advances in state space models [49] may further boost performance. In conclusion, DNA language models represent a powerful framework for genome-wide variant effect prediction, and we believe that exploring the above avenues to further improve GPN would be worthwhile.

3.4 Materials and Methods

Pre-training. We obtained a list of Brassicales reference genome assemblies from NCBI Genome (`https://www.ncbi.nlm.nih.gov/data-hub/genome/`), filtered for RefSeq-annotated and kept only one per genus, resulting in a total of 8 reference genomes (Table B.1). We held out *Arabidopsis thaliana* chromosomes 4 and 5 for validation and testing, respectively. For each genome, we subsampled genomic windows of size 512 bp, with a step of 256 bp and augmented with the reverse complement. However, we did not draw genomic windows uniformly from the whole genome, but emphasized certain regions. In particular, we took the union of exons (with a small intronic flank), promoters (1000 bp upstream of transcription start sites) as well as an equivalent amount of random windows from the whole genome. We think this decision may improve performance, but leave experimentation for further studies. Additionally, we subset the number of windows from each genome to the number of windows from *Arabidopsis*, given its unusually small genome.

We set up a masked language modeling task [33], in which 15% of the tokens in a nucleotide sequence were masked and had to be predicted from their context. In contrast to most DNA

language models that tokenize sequences into overlapping k-mers [63, 161, 52] or use byte-pair encoding [164], we used bare nucleotides as tokens. While a thorough benchmark of different tokenization strategies is lacking, using single-nucleotide tokens makes interpretation easier, in particular for unsupervised variant effect prediction.

While language model pre-training successes were first showcased by transformer architectures, convolutional models have shown similarly good performance in natural language [133] and protein modeling [160]. In our initial experiments, we noticed that convolutional models converged faster than transformer models. The locality of convolutions may be a good inductive bias for modeling DNA sequences at this scale. The linear complexity of convolution also simplifies inference or fine-tuning on longer sequences such as entire chromosomes, which in the case of transformers might require chunking (with some overlap) and aggregating the results.

We implemented GPN, a convolutional neural network, using the Hugging Face library [158]. The masked DNA sequence was one-hot encoded and then consecutively processed by 25 convolutional blocks. Each convolutional block consisted of a dilated convolutional layer followed by a feed-forward layer, with intermediate residual connections and layer normalization (Figure 3.1). Throughout the layers, the embedding dimension (number of convolutional filters) was kept fixed at 512. The dilation was increased exponentially up to a certain value and then cycled. A list of hyperparameters is displayed in Table B.3. We trained three models varying only in the loss weight on repetitive elements (marked lowercase in the FASTA file). We trained each model for 150 K steps, taking approximately 4 days with 4 NVIDIA A100 80GB GPUs. Perplexity is defined as the exponentiation of the cross-entropy loss, which is equivalent to 1 over the probability given to the correct nucleotide. Test perplexity is displayed in Table B.2. We also trained a separate model on the single genome of *Arabidopsis thaliana*, with a repeat weight of 0.1 and the same hyperparameters except for only 12,000 steps with decaying learning rate, as we noticed it would soon start overfitting. This model obtained a higher test perplexity of 3.13 (3.17 on non-repeat regions).

Analysis of model embeddings. Model embeddings were averaged over non-overlapping 100-bp windows. Embeddings from the forward and reverse strand were averaged, and then standardized. UMAP was run with default parameters. The gene annotation was downloaded from EnsemblPlants. The annotation of repetitive elements was downloaded from `http://ucsc.gao-lab.org/cgi-bin/hgTables?hgsid=167291_E9nY5UIAQRUOAR01xJAsum4vDukw`. We considered intergenic regions with 100% overlap with repeats as a separate "Repeat" class. Windows with ambiguous annotation (e.g., 50% CDS and 50% intron) were excluded from the analysis. Genomic region classification was performed with logistic regression as implemented by `scikit-learn` [107], using class weight inversely proportional to frequency and L2 regularization strength chosen via cross-validation. Windows in each chromosome were predicted by a model trained on the remaining chromosomes.

Motif analysis. Each position in the genome was independently masked and the model distribution over nucleotides was extracted. The distribution was averaged between the results from the forward and reverse strands. The held-out model perplexity was computed

for splice acceptors, splice donors, start codons, stop codons, CDS and intergenic and intronic positions in the 1 Mb region `Chr5:3,500,000-4,500,000`, after excluding repeats.

An adaptation of TF-MoDISco was run with model predictions in regions 1000 bp upstream and downstream of transcription start sites (all chromosomes), after filtering repeats and coding exons. The exact score fed into Modisco was the nucleotide probability minus 0.25, so it would be roughly centered at 0. Since TF-MoDISco expects genomic windows of equal length, we concatenated our variable-length windows into one large window, interspersed with 20 undefined 'N' nucleotides.

Variant effect prediction. We scored variants by masking the position and calculating the log-likelihood ratio between the alternate and reference allele. Scores computed from the forward and reverse strands were averaged. We calculated odds ratio and p-value with Fisher's exact test. When comparing to phyloP and phastCons, we excluded variants where these scores are undefined (due to the lack of whole-genome alignment).

All possible SNPs in the region `Chr5:3,500,000-4,500,000` were generated and their consequences annotated with Ensembl Variant Effect Predictor [94] web interface https://plants.ensembl.org/Arabidopsis_thaliana/Tools/VEP, with the `upstream/downstream` argument set to 500, used to call variants as upstream/downstream instead of intergenic. We compared scores for variant types with at least 1000 variants, and we excluded variants with different consequences in different transcripts.

The 1001 Genomes genotype matrix was downloaded from https://aragwas.1001genomes.org/api/genotypes/download and combined with metadata from https://1001genomes.org/data/GMI-MPI/releases/v3.1/1001genomes_snp-short-indel_only_ACGTN.vcf.gz. This genotype matrix is binary, since all the accessions are homozygous, as *Arabidopsis* is predominantly selfing. For variants with alternate allele frequency greater than 50%, we flipped the sign of GPN scores (equivalent to taking the log-likelihood ratio between the minor and the major allele), and did all analyses in terms of minor allele frequency. Variant consequences produced by Ensembl Variant Effect Predictor were downloaded from Ensembl Plants. Conservation scores were downloaded from http://plantregmap.gao-lab.org/download.php#alignment-conservation. For conservation scores phyloP and phastCons, we simply flipped the sign to obtain a variant score, i.e., variants at conserved sites should be considered more pathogenic. We additionally scored variants using (minus) the absolute value of phyloP, referred to as abs(phyloP), which means prioritizing putative accelerated regions over putative neutral ones. We defined rare variants as those with allele count equal to 1 (to be precise, it is two alleles in the same homozygous accession), and common variants as those with allele frequency above 5%. Model scores were defined as pathogenic or benign based on a quantile threshold that we varied from 0.1% to 10%.

GWAS summary statistics for all 462 phenotypes were downloaded through the AraGWAS API, with the default threshold of minimum allele count of 6 (i.e., at least 6 homozygous accessions having the allele). The summary statistics include information on whether an association is significant according to a permutation-based approach (recommended [138]) as well as a Bonferroni threshold. The LD matrix of squared Pearson correlations

(r^2) was calculated within a radius of 100 kb around each variant, using `sgkit` (https://pystatgen.github.io/). We define a weighted sum of GPN scores according to LD (i and j index SNPs):

$$\text{GPN}\times\text{LD}_i = -\sum_j |\text{GPN}_j| \cdot r_{ij}^2.$$

This is known as a stratified LD Score [46] and can also be interpreted as the multiplication between the LD matrix and the vector of GPN scores. The reason why we used unsigned LD and model scores is that we focused on assessing whether a variant would have a significant association with differences in a trait, regardless of the direction of the association. Since the association p-value is invariant to recoding of reference and alternate alleles, we took the absolute value of GPN scores. We arbitrarily added a negative sign in front to be consistent with more negative implying more likely functional. We similarly defined phyloP×LD (first shifting the scores to reside entirely on the negative side of the number line), abs(phyloP)× and phastCons×LD. We considered the baseline LD Score [25], the unweighted sum of LD with a given variant:

$$\text{LD Score}_i = -\sum_j r_{ij}^2.$$

Code availability. Code to reproduce all results, including instructions to load the pre-trained model, is available at https://github.com/songlab-cal/gpn.

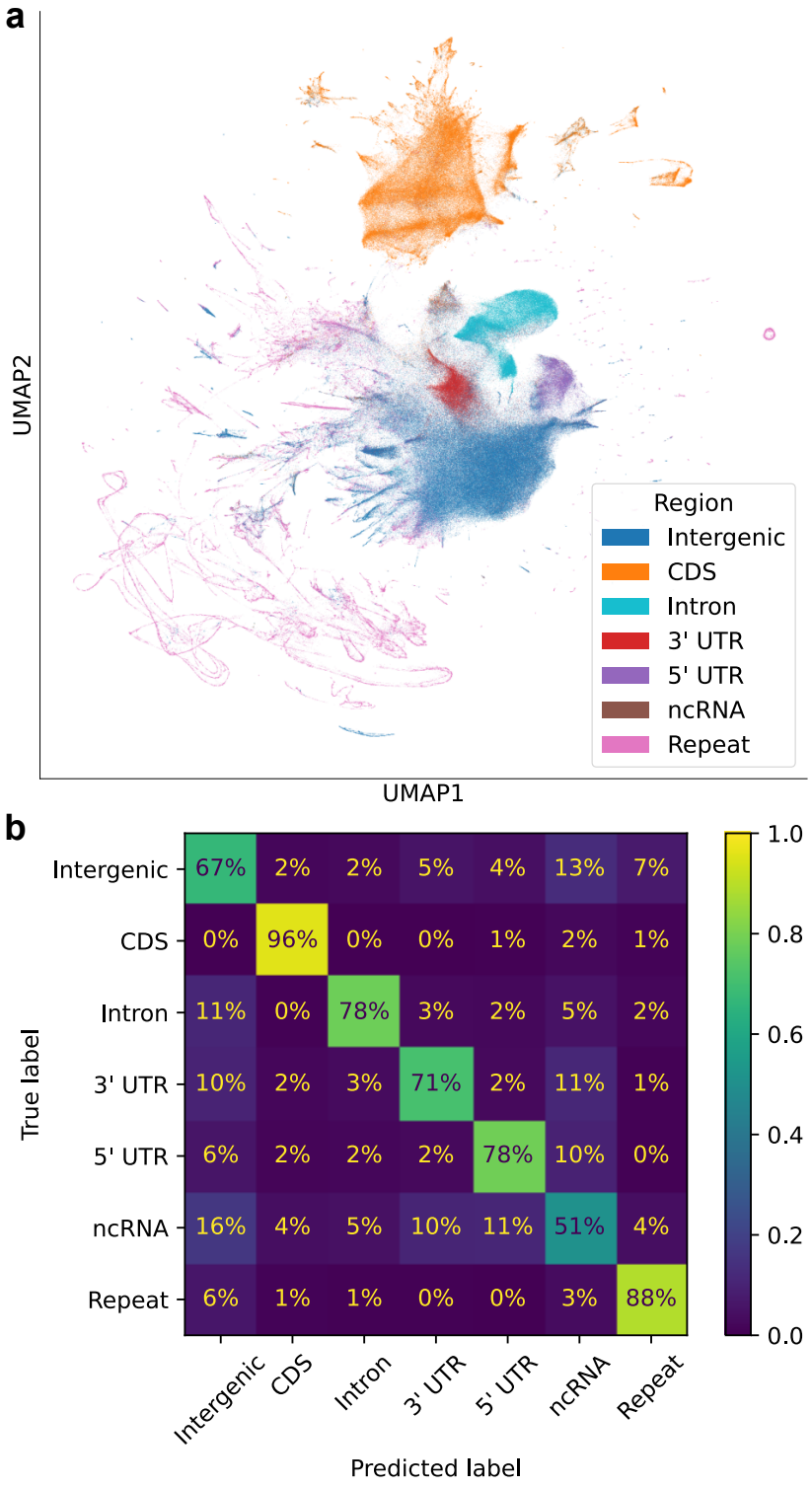


Figure 3.2: **Unsupervised clustering of genomic windows.**

Figure 3.2 *(continued)*: (a) UMAP visualization of GPN embeddings averaged over non-overlapping 100 bp windows along the genome, annotated with gene region. (b) Confusion matrix for classification of gene regions using a logistic regression model trained on averaged embeddings. Each chromosome was predicted from a model trained on the remaining chromosomes.

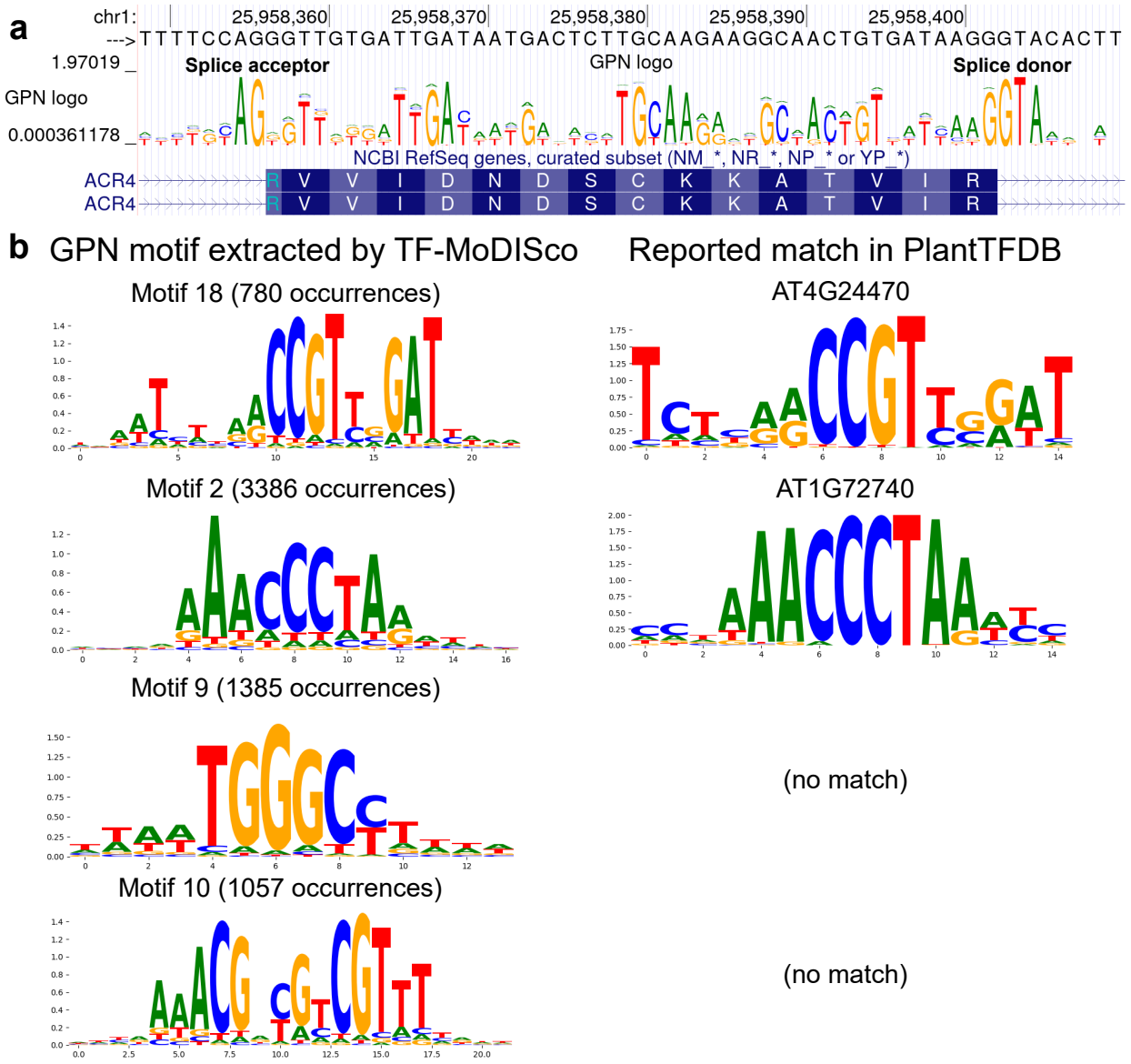


Figure 3.3: **Sequence logos derived from model predictions.** Each position in the genome was independently masked and the model distribution over the four nucleotides was computed. (a) Sequence logo visualized in the UCSC Genome Browser (https://genome.ucsc.edu/s/gbenegas/gpn-arabidopsis). The height of each letter is proportional to its probability, while the overall height at each position is equal to 2 minus the entropy of the distribution. (b) Example GPN motifs in promoter regions, extracted by TF-MoDISco, with significant matches in PlantTFDB.

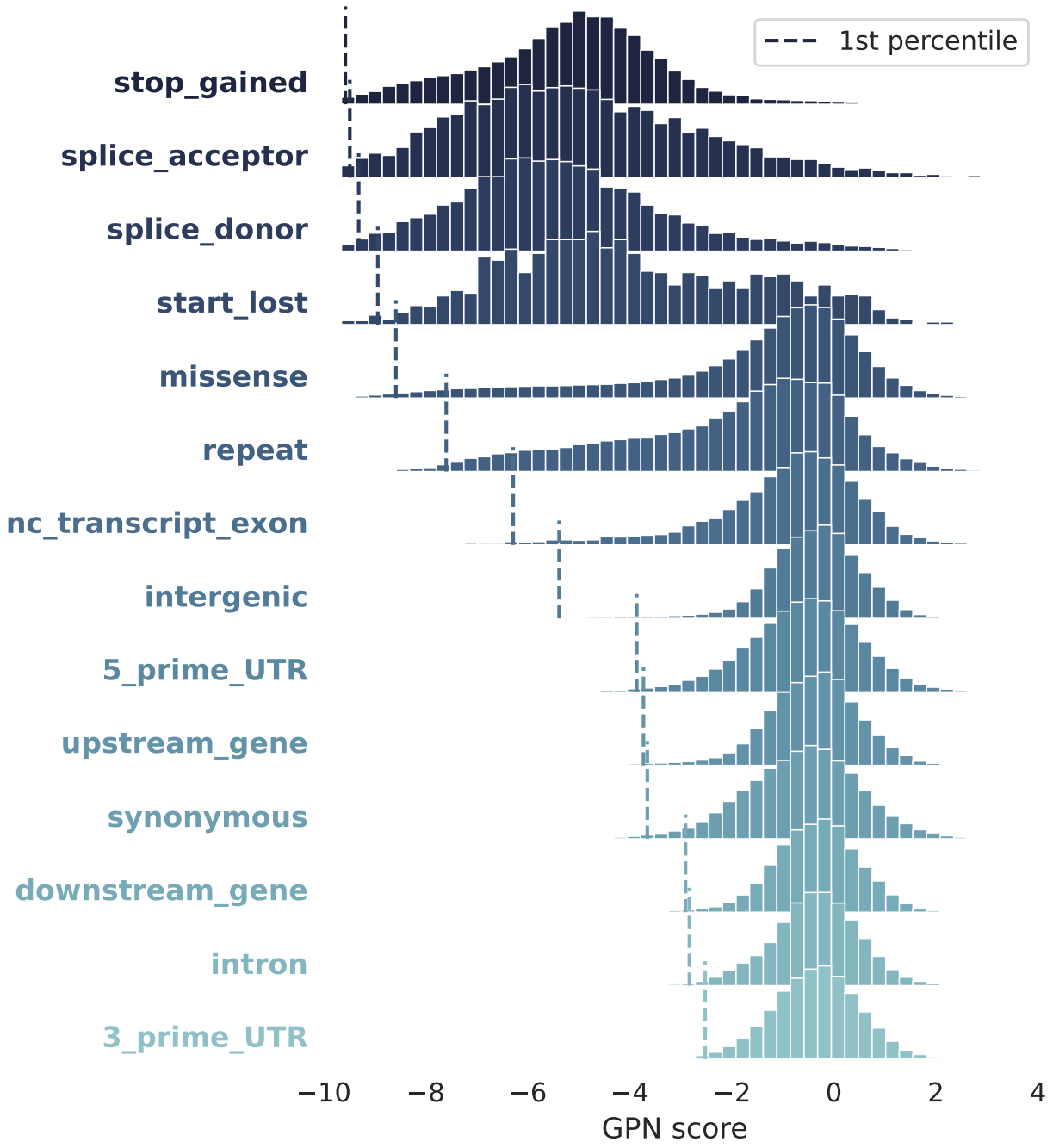
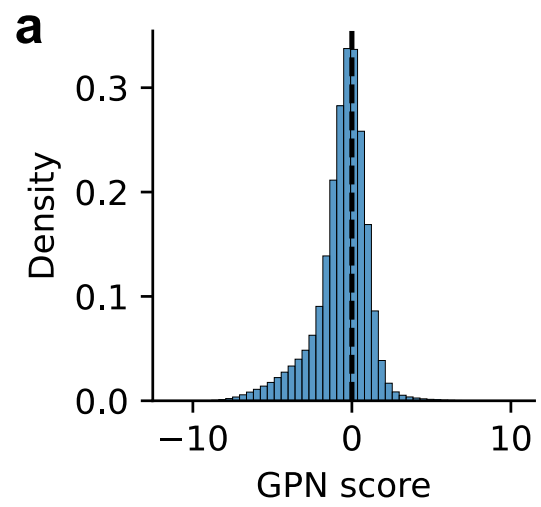
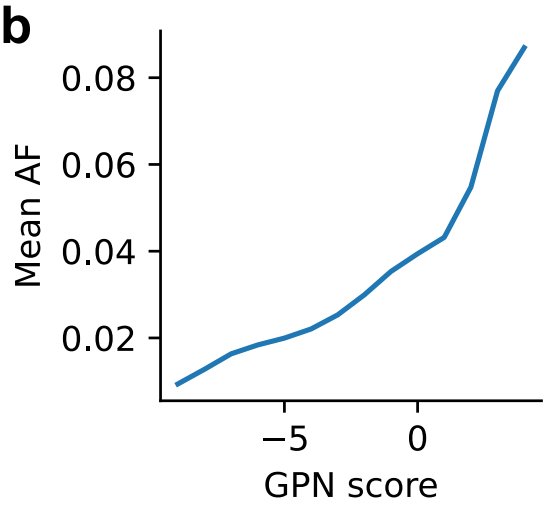


Figure 3.4: **Variant effect prediction: *in silico* mutagenesis.** Distribution of GPN scores computed for all possible single-nucleotide polymorphisms (SNPs) in a 1 Mb region, across categories, sorted by 1st percentile (dashed vertical lines).





c

GPN score	Rare (AC = 1)	Common (AF ≥ 5%)
Putative functional (lower 0.1%)	4 173	391
Putative non-functional (upper 99.9%)	3 007 991	1 551 191

Odds ratio: 5.5, p-value < 1e−300

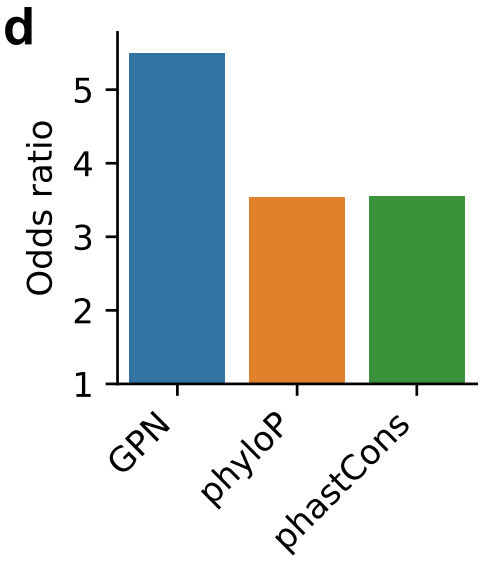
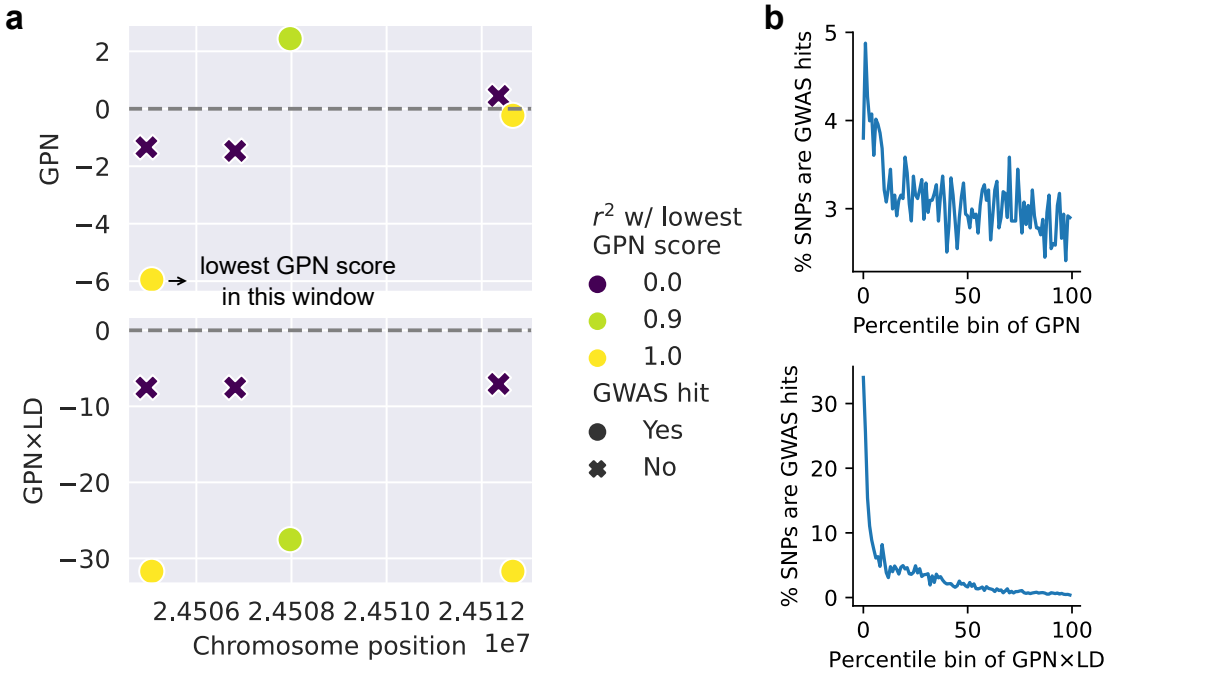


Figure 3.5: **Variant effect prediction: rare vs. common.** The GPN score was computed for over 10 million variants in the 1001 Genomes. (a) Distribution of GPN scores. (b) Mean allele frequency for different GPN score bins ($[-9.5, -8.5), [-8.5, -7.5), \ldots, [3.5, 4.5)$). (c) Contingency table and odds ratio showing enrichment of putative functional GPN scores in rare (AC = 1) vs. common (AF ≥ 5%) variants. AC: allele count. AF: allele frequency. (d) Comparison of odds ratios as in (c) obtained with different models. abs(phyloP) is excluded as it did not achieve a significant enrichment.



c

GPN×LD	GWAS hit	Not GWAS hit
Putative functional (lower 1%)	1 015	3 530
Putative non-functional (upper 99%)	12 233	436 503

Odds ratio: 10.3, p-value $<$ 1e−300

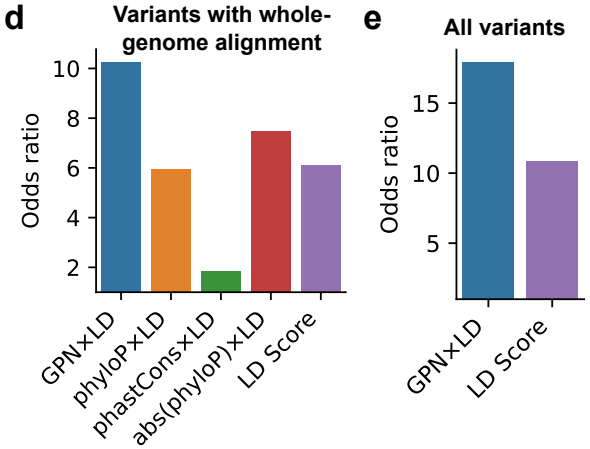


Figure 3.6: **Variant effect prediction: GWAS.** GPN scores were analyzed for around half a million variants tested in AraGWAS. (a) Example window with six variants tested for association with maximum temperature in January. GPN×LD successfully separates GWAS hits and non-hits. (b) Percentage of GWAS hits (for any trait) in each percentile bin of GPN and GPN×LD scores. (c) Contingency table and odds ratio showing enrichment of GWAS hits (for any trait) in putative functional (associated) GPN×LD scores. (d) Comparison of odds ratios obtained with different models ($n = 453,281$ variants with whole-genome alignment). (e) Odds ratios with the full variant set ($n = 510,462$).

Chapter 4

GPN-MSA: an alignment-based DNA language model for genome-wide variant effect prediction

This is joint work with Carlos Albors, Alan J. Aw, Chengzhong Ye and Yun S. Song, released on *bioRxiv* [12] and currently under review. I thank Martin Kircher for helpful correspondence regarding CADD.

4.1 Introduction

With the rising trend in whole-genome sequencing, there is a pressing need to understand the effects of genome-wide variants, which would lay the foundation for precision medicine [48]. In particular, predicting variant deleteriousness is key to rare disease diagnosis [91] and rare variant burden tests [78]. Indeed, a recent review highlights analysis of functional rare variants as the biggest contribution of human genetics to drug discovery [139].

Language models are gaining traction as deleteriousness predictors, with their ability to learn from massive sequence databases and score variants in an unsupervised manner. Given the success of accurately scoring missense variants with protein language models [97, 21, 62], it is natural to consider scoring genome-wide variants with DNA language models. For this task, we recently developed the Genomic Pre-trained Network (GPN), a model based on a convolutional neural network trained on unaligned genomes, and showed that it achieves excellent variant effect prediction results in the compact genome of *Arabidopsis thaliana* [10]. The human genome – which harbors a similar number of genes but interspersed over nearly 23 times larger regions and contains much more repetitive elements, most of which may not be functional – is substantially harder to model, however. In fact, previous attempts at unsupervised variant effect prediction with human DNA language models (e.g., Nucleotide Transformer [32]) have shown inferior performance compared to simpler conservation scores. Increasing the scale of the model, data, and compute improves performance, but it can still

be poor, even for a model trained for 28 days using 128 top-line graphics processing units (GPUs) [32].

To address the above challenge, we here introduce GPN-MSA, a novel DNA language model which is designed for genome-wide variant effect prediction and is based on the biologically-motivated integration of a multiple-sequence alignment (MSA) across diverse species using the flexible Transformer architecture [143]. We apply this modeling framework to humans using an MSA of diverse vertebrate genomes [4] and show that it outperforms not only previous DNA language models but also current widely-used models such as CADD [114], phyloP [110], ESM-1b [116, 21], Enformer [7], and SpliceAI [61]. Our model takes only 4.75 hours to train on 4 GPUs, which is a considerable reduction in the required computing resources compared to the aforementioned Nucleotide Transformer [32]. We anticipate that this massive reduction in computational footprint will enable the efficient exploration of new ideas to train improved DNA language models for genome-wide variant effect prediction.

4.2 Results

GPN-MSA is trained on a whole-genome MSA of 100 vertebrate species (Figure 4.1a, full tree in Figure C.1), after suitable processing (Figure 4.1b) and filtering (Figure 4.1c). It is an extension of GPN [10] to learn nucleotide probability distributions conditioned not only on surrounding sequence contexts but also on aligned sequences from related species that provide important information about evolutionary constraints and adaptation (Figure 4.1d, Materials and Methods). It draws heavy inspiration from the MSA Transformer [113], a protein language model trained on MSAs of diverse protein families; it was originally designed for structure prediction but was later shown to achieve excellent missense variant effect prediction performance [97]. Besides the fact that our model operates on whole-genome DNA alignments – which comprise small, fragmented synteny blocks with highly variable levels of conservation, and hence considerably more complex than protein alignments – there are also essential differences in the architecture and training process of GPN-MSA from the MSA Transformer (Materials and Methods).

We demonstrate the capability of GPN-MSA to improve unsupervised deleteriousness prediction on several human variant datasets (Materials and Methods). We emphasize that only the reference genome is used to train GPN-MSA and that no human variant dataset is utilized in training. Nevertheless, GPN-MSA can still capture several functional attributes of variants, such as epigenetic marks and the impact of natural selection (Figure C.2, Figure C.3).

For evaluation, we first consider the classification of ClinVar [74] pathogenic vs. common missense variants in gnomAD [28]. We use common variants as control instead of ClinVar benign-labeled variants, as recommended by the developers of CADD to reduce ascertainment bias [114]. We find that GPN-MSA achieves the best performance compared to genome-wide predictors CADD [115], phyloP [110], the Nucleotide Transformer (NT) [32], and the missense-specific ESM-1b [116, 21] (Figure 4.2a, Figure C.4a).

Next, we consider the classification of somatic missense variants frequently observed across cancer tumors (COSMIC, the Catalogue of Somatic Mutations in Cancer [132]) vs. gnomAD common missense variants. Because of the extreme class imbalance in this case, we focus on the precision and recall metrics. GPN-MSA again achieves the highest performance, with substantial margins of improvement over other models (Figure 4.2b, Figure C.4b).

Moving on to regulatory variants, we evaluate on the classification of a curated set of variants implicated in Mendelian disorders (OMIM, Online Mendelian Inheritance in Man [123]) vs. gnomAD common variants. We again consider precision and recall because of the extreme class imbalance, and find that GPN-MSA achieves the best performance overall, as well as in each variant category (Figure 4.2c, Figure C.4c). For several variant categories, CADD's precision increases from near zero as recall increases, which indicates that a substantial fraction of its top discoveries are actually false (Figure C.4c).

Lastly, we evaluate on the enrichment of rare vs. common gnomAD variants in the tail of deleteriousness scores. Deleterious mutations should be under purifying selection and hence their frequencies tend to be low in the population. Therefore, if a variant effect predictor is accurate, we expect rare variants to be enriched compared to common variants for extreme deleteriousness scores. GPN-MSA achieves the highest overall enrichment, as well as in each variant category, with different margins (Figure 4.2d). For missense variants, high enrichment is obtained by GPN-MSA with even less stringent deleteriousness score cutoffs (Figure C.5). In the case of intron variants, it also outperforms SpliceAI [61], a state-of-the-art splicing predictor. We note that the overall performance is not merely an averaging of the performances in the different categories; it also involves scoring variants relative to each other across these categories. On a separate enrichment analysis of low-frequency vs. common gnomAD variants in gene flanking and intergenic regions, GPN-MSA achieves a substantially improved performance over Enformer [7] (Figure 4.2e).

Examining the complete gnomAD variant set, there seems to be a near-linear relationship between the GPN-MSA score bin and the logarithm of the average minor allele frequency within that specific bin (Figure 4.2f). We believe that the deleteriousness of GPN-MSA scores should be interpreted as a continuum; if a hard threshold is helpful, we recommend a cutoff around -7, based on the distribution of scores in different datasets (Figure C.6). Incidentally, the bimodality of score distribution for frequent variants in COSMIC suggests that many of them could be passenger mutations (Figure C.6b).

While we observe that SpliceAI and Enformer, which are functional genomics models, perform worse than the simpler phyloP in deleteriousness prediction, we note that this is an application they were not designed for. It is also worth noting that although phyloP trained on 241 mammals (Zoonomia) was recently proposed as a deleteriousness predictor [128], the older vertebrate phyloP actually achieves better results in most of our benchmarks (Figure C.7).

To understand the importance of different components of our model, we perform an ablation study and assess the impact on variant effect prediction performance (Figure C.8). We find that the inclusion of the MSA is most critical and that different ways of prioritizing conserved regions can have a significant impact on the results.

GPN-MSA's predictions for every position of chromosome 6 can be visualized as sequence logos [118] in the UCSC Genome Browser [69, 99] (example in Figure C.9); we plan to release predictions for all ∼9 billion possible single nucleotide variants in the human genome using the final, revised model upon publication. We also provide scores for ∼530 million single-nucleotide variant in gnomAD, as well as a Jupyter notebook detailing how to run predictions on a given VCF file using our trained model.

4.3 Discussion

To recapitulate, our main contributions are threefold. First, we propose the first DNA language model operating directly on a whole-genome alignment. Second, we demonstrate state-of-the-art performance in humans on a number of clinically-relevant variant effect prediction datasets. Lastly, the general approach we have developed for humans is computationally efficient, which would enable future research in the field.

In the rapidly advancing landscape of DNA language modeling, scaling up model and context sizes has been the primary avenues of exploration [32, 100, 38]. In contrast, in our work we focus on the explicit modeling of related sequences (known as retrieval augmentation in natural language processing [17]). This has led to a highly computationally efficient model and state-of-the-art variant effect prediction performance for both coding and non-coding variants. It remains to be explored how useful GPN-MSA's learned representations would be for downstream applications, e.g., for genome annotation or gene expression prediction. Expanding the context length, possibly through leveraging recent technical developments [100], might be beneficial for such tasks.

The masked language modeling objective can be too easy if sequences very similar to the human genome are included in the MSA, resulting in the learned probability distribution being not very useful for variant effect prediction. This observation has led us to exclude most primate genomes during training. To tackle this limitation, we are actively exploring alternative training objectives which are aware of phylogenetic relationships. We are also exploring how best to integrate population genetic variation information, instead of relying on a single reference genome.

In our view, one of the most promising applications of GPN-MSA is effective genome-wide rare variant burden testing, which has been mostly restricted to coding regions [152]. We envision that several other statistical genetics tasks can be empowered by GPN-MSA, such as functionally informed fine-mapping [153] and polygenic risk scores [89].

Sequence models (such as phyloP and GPN-MSA) might achieve better deleteriousness prediction results but are still less interpretable than functional genomics models such as SpliceAI and Enformer. While both functional genomics models and DNA language models have much room for independent improvement, it is likely that jointly modeling DNA sequence and functional genomics may have the biggest impact.

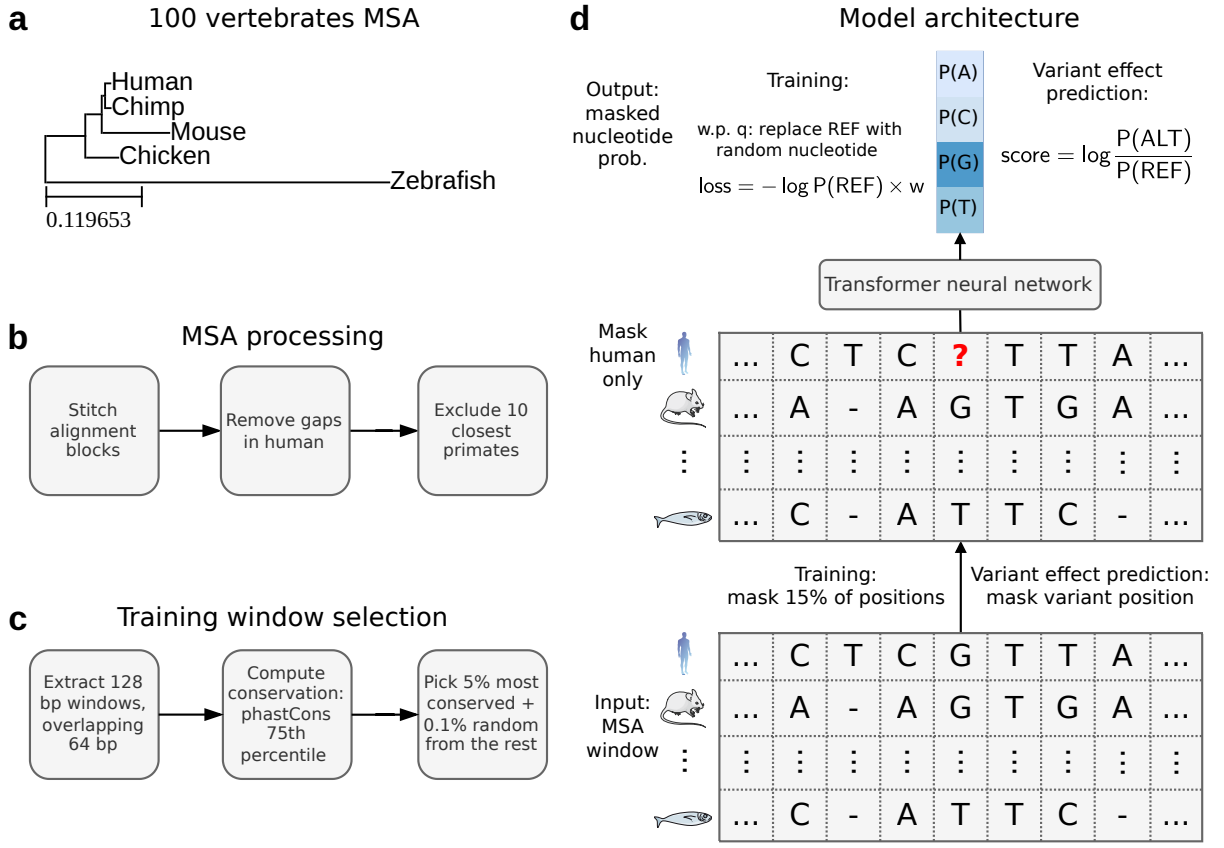


Figure 4.1: **Overview of GPN-MSA. (a)** Subsampled phylogenetic tree of 100 vertebrate species constituting the whole-genome MSA (full tree in Figure C.1). **(b)** MSA processing. Starting with a Multiple Alignment Format file, alignment blocks are stitched together following the order in the human reference. Columns with gaps in the human reference are discarded, followed by the removal of the 10 primate species closest to human (Chimp to squirrel monkey). **(c)** Training window selection. For each 128-bp window along the genome, conservation is computed as the 75^{th} percentile of phastCons. The top 5% conserved windows are chosen alongside a random 0.1% from the remaining windows. **(d)** Model architecture. The input is a 128-bp MSA window where certain positions in the human reference have been masked, and the goal is to predict the nucleotides at the masked positions, given the context across both columns (positions) and rows (species) of the MSA. During training, 15% of the positions are masked. During variant effect prediction, only the variant position is masked. The sequence of MSA columns is processed through a Transformer neural network resulting in a high-dimensional contextual embedding of each position. Then, a final layer outputs four nucleotide probabilities at each masked position. The model is trained with a weighted cross-entropy loss, designed to downweight repetitive elements and up-weight conserved elements (Materials and Methods). As data augmentation in non-conserved regions, prior to computing the loss, the reference is sometimes replaced by a random nucleotide (Materials and Methods). The GPN-MSA variant effect prediction score is defined as the log-likelihood ratio between the alternate and reference allele. REF: reference allele. ALT: alternate allele.

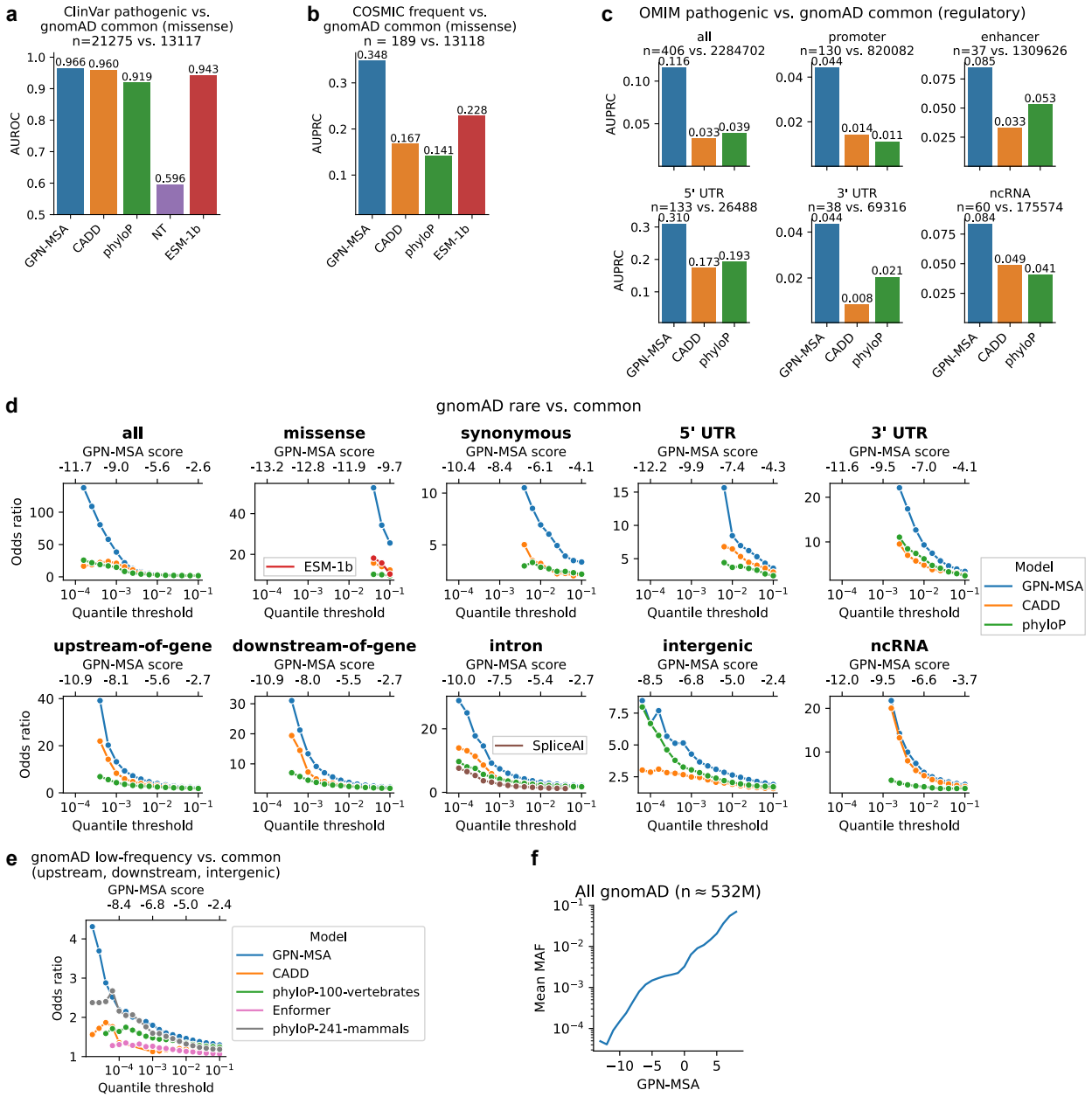


Figure 4.2: **Comparison of variant effect prediction results.** **(a)** Classification of ClinVar pathogenic vs. gnomAD common missense variants. NT: Nucleotide Transformer (version `2.5b-multi-species`). **(b)** Classification of COSMIC frequent (frequency $> 0.1\%$) vs. gnomAD common missense variants. **(c)** Classification of OMIM pathogenic vs. gnomAD common regulatory variants. We matched OMIM promoter variants with gnomAD upstream-of-gene variants, enhancer with intergenic, and "all" with the union of the matches of the specific categories, after removing any overlap with missense variants.

Figure 4.2 *(continued)*: **(d)** Enrichment of rare (singletons) vs. common (MAF > 5%) gnomAD variants (subset) in the tail of deleterious scores (defined using different threshold quantiles, e.g. the 10% most extreme scores are considered deleterious, or the 1% most extreme). The number of rare and common variants in each category is as follows. all: 4812825 vs. 4811795, missense: 37757 vs. 13118, synonymous: 18647 vs. 17566, 5′ UTR: 35538 vs. 26488, 3′ UTR: 82954 vs. 69316, upstream-of-gene: 849851 vs. 820082, downstream-of-gene: 869964 vs. 852410, intron: 1804469 vs. 1690417, intergenic: 1226643 vs. 1309626, ncRNA: 183960 vs. 175574. In categories other than "all" or "missense", we removed any overlap with missense variants. Odds ratios and p-values were computed using one-sided Fisher's exact test. All shown odds ratios have p-value < 0.05. The minimum threshold was chosen such that no score has less than 10 counts in the contingency table. **(e)** Comparison with Enformer. Enrichment of low-frequency ($0.5\% < \text{AF} < 5\%$, $n = 3539816$) vs. common (MAF > 5%, $n = 2125523$) gnomAD flanking and intergenic variants in the tail of deleterious scores. We removed any overlap with missense variants. Enformer scores were calculated as L^2 norm of delta predictions. We used the same odds ratio plotting considerations as in (d). **(f)** Mean MAF for different bins of GPN-MSA scores ($[-13.5, -12.5), [-12.5, -11.5), \ldots, [7.5, 9.5)$) in the full gnomAD set. AUROC: area under the receiving operating characteristic curve. AUPRC: area under the precision-recall curve. MAF: minor allele frequency. AF: allele frequency. "phyloP" refers to the statistic computed on the 100 vertebrates alignment.

4.4 Materials and Methods

MSA Processing

The `multiz` [15] whole-genome alignment of 100 vertebrates was downloaded from https://hgdownload.soe.ucsc.edu/goldenPath/hg38/multiz100way/maf/. Contiguous alignment blocks were stitched together using the `multiz` utility `maf2fasta` and any columns with gaps in human were removed. The 10 primate species closest to human were removed. We also downloaded associated conservation scores phastCons [122] https://hgdownload.soe.ucsc.edu/goldenPath/hg38/phastCons100way and phyloP [110] https://hgdownload.soe.ucsc.edu/goldenPath/hg38/phyloP100way.

Training Region Selection

Instead of training on the whole genome, we focused on the most conserved genomic windows, aiming to emphasize functionally-important regions such as exons, promoters and enhancers. The conservation of a genomic window was defined as the 75$^{\text{th}}$ percentile of phastCons scores in the window. We then chose a cutoff; in our current experiments we included the top 5% most conserved windows. We also included 0.1% of the remaining windows of the genome to ensure there is no extreme distribution shift when performing variant effect prediction in non-conserved regions. The reverse complement of each selected window was added as data augmentation. Chromosome 21 was held out for validation (early-stopping) and chromosome 22 was held out for possible testing (not actually used in this study).

Model Architecture

We adopt the general approach of masked language modeling [33]. As a general caveat, in this work we did not systematically tune hyperparameters, so they are likely far from optimal. The input is a 128-bp MSA window where certain positions in the human reference have been masked, and the goal is to predict the nucleotides at the masked positions, given its context across both columns (positions) and rows (species) of the MSA. During training, 15% of the positions are masked. During variant effect prediction, only the variant position is masked. The 1-hot encodings of nucleotides from different species at each position are first concatenated. Then, the sequence of MSA columns is processed through a Transformer neural network (RoFormer [127]) resulting in a high-dimensional contextual embedding of each position. Then, a final layer outputs four nucleotide probabilities at each masked position. The model is trained on the reference sequence with a weighted cross-entropy loss.

Our considerations for the loss weight were the following: downweighting repeats and upweighting conserved elements (so wrong predictions in neutral regions are penalized less). We introduce a smoothed version of phastCons, phastCons_M, as the max of phastCons over a window of 7 nucleotides. The goal was to not only give importance to conserved regions, but to regions immediately next to them. The loss weight w is defined as follows:

$$w \propto (0.1 \times \mathbb{1}\{\text{repeat}\} + \mathbb{1}\{\neg\text{repeat}\}) \times \max(\text{phyloP}, 1) \times (\text{phastCons}_M + 0.1)$$

which includes 10-fold downweighting on repetitive elements [10] plus upweighting based on both phyloP and phastCons_M.

As data augmentation in non-conserved regions, prior to computing the loss, the reference is replaced by a random nucleotide with a certain probability q:

$$q = 0.5 \times \mathbb{1}\{\text{phastCons}_M < 0.1\}$$

The intention is to guide the model to assign more neutral scores in non-conserved regions.

Our code is based on the Hugging Face Transformers library [158]. All models were trained with default hyperparameters [1] (e.g. 12 layers with 12 attention heads each) except for the ones listed in Table C.1. The total number of parameters is approximately 86 million. We performed early stopping based on validation loss. We manage to train the model in approximately 4.75 hours using 4 NVIDIA A100 GPUs.

The GPN-MSA variant effect prediction score is defined as the log-likelihood ratio between the alternate and reference allele. In our experiments, we average the predictions from the positive and negative strand. With our 4 NVIDIA A100 GPUs, we manage to score approximately 5 million variants per hour.

[1] https://huggingface.co/docs/transformers/model_doc/roformer#transformers.RoFormerConfig

Differences between GPN-MSA and MSA Transformer

While the MSA Transformer takes as input an arbitrary set of aligned sequences, GPN-MSA is trained on sequences from a fixed set of species. This allows simpler modeling of the MSA as a sequence of fixed-size alignment columns, reducing computation and memory requirements. Variant effect prediction, masking only the target sequence (in our case, human), is identical [97]. Since variant effect prediction is our main goal, during training we also only mask positions from the target sequence. The MSA Transformer, however, proposes masking MSA entries at random during training, based on results from structure prediction, their intended application.

Ablation Study

We performed an ablation study to understand the impact of each of our design choices on variant effect prediction when modified independently (Figure C.8). For each setting, three replicate models with different seeds were trained, where applicable. Since we hold the rest of the hyperparameters fixed, results should be interpreted as differences given a similar training procedure and compute budget.

- w/o MSA: the model is only trained on the human sequence, without access to other species.
- MSA frequency: variants are scored using the log-likelihood ratio of observed frequencies in the MSA column, with a pseudocount of 1.
- Train on 50% most conserved: expand the training region from the smaller 5% most conserved to a larger set with less overall conservation.
- Include closest primates: do not filter out from the MSA the 10 primates closest to human.
- Don't upweight conserved: do not upweight the loss function on conserved elements.
- Don't replace non-conserved: do not replace the reference in non-conserved positions with random nucleotides when computing the loss function.

Modeling the single human sequence instead of the MSA has by far the biggest impact. Using the column-wide MSA frequencies as predictor also shows a large decrease in performance. Including primate species close to human, or training on less conserved regions, have a moderate impact on performance. Finally, of relatively minor impact are removing the upweighting of conserved elements or removing the data augmentation procedure of replacing nucleotides in non-conserved positions.

Variant Effect Prediction (VEP) glossary

We summarize datasets and their provenance, metrics used to evaluate each dataset, and technical details in constructing VEP scores below.

VEP data sources:

- ClinVar [74]: downloaded release `20230730`.
- COSMIC [132]: downloaded `Cosmic_MutantCensus_v98_GRCh38.tsv.gz` and computed frequency as the proportion of samples containing the mutation, restricting to whole-genome or whole-exome samples.
- OMIM [123]: downloaded a set of curated pathogenic regulatory variants.
- gnomAD [28]: downloaded version `3.1.2` and filtered to autosomal variants with allele number of at least $2 \times 70\,000$, besides the official quality-control flags. In each autosomal chromosome, selected all common variants (minor allele frequency $> 5\%$) as well as an equally-sized subset of rare variants (singletons).

VEP metrics:

- ClinVar: area under the receiving operating characteristic curve (AUROC) for classification of ClinVar "Pathogenic" vs. gnomAD common missense variants.
- COSMIC: area under the precision-recall curve (AUPRC) for classifying COSMIC frequent (frequency $> 0.1\%$) vs. gnomAD common missense variants.
- OMIM: AUPRC for classification of OMIM pathogenic vs. gnomAD common regulatory variants. We matched OMIM promoter variants with gnomAD upstream-of-gene variants, enhancer with intergenic, and "all" with the union of the matches of the specific categories, after removing any overlap with missense variants.
- gnomAD: enrichment of rare vs. common gnomAD variants in the tail of deleterious scores (defined using different threshold quantiles, e.g. the 10% most extreme scores are considered deleterious, or the 1% most extreme). In categories other than "all" or "missense", we removed any overlap with missense variants.

VEP scores:

- GPN-MSA: log-likelihood ratio between alternate and reference allele. Predictions from both strands were averaged.
- CADD: raw scores, negated so lower means more deleterious.
- phyloP: computed on 100 vertebrate alignment, negated so lower means more deleterious.

- Nucleotide Transformer (NT): the center 6-mer was masked and the score was computed as the log-likelihood ratio between alternate and reference 6-mer. Predictions from both strands were averaged. Given the high computational requirements, we only scored variants for the ClinVar metric. The performance of the four different models can be seen in Figure C.10.
- ESM-1b: precomputed log-likelihood ratios between alternate and reference alleles were obtained in protein coordinates [21]. For variants affecting multiple isoforms, the minimum (most deleterious) score was considered.
- SpliceAI: precomputed scores recommended for variant effect prediction (`spliceai_scores.masked.snv.hg38.vcf.gz`) were downloaded from https://basespace.illumina.com/s/otSPW8hnhaZR. The authors do not recommend any specific way of computing a single deleteriousness score. We scored variants using minus the maximum absolute delta in splice acceptor or donor probability in any gene.
- Enformer: precomputed scores for variants with minor allele frequency (MAF) greater than 0.5% in any 1000 Genomes population [30] were downloaded from https://console.cloud.google.com/storage/browser/dm-enformer/variant-scores. These were intersected with upstream-of-gene, downstream-of-gene and intergenic variants with gnomAD MAF greater than 0.5%. The authors do not recommend any specific way of computing a single deleteriousness score. We scored variants using minus the norm of the 5 313 delta features (SNP Activity Difference or SAD). We found that the L^1 and L^2 norms seem to perform similarly, better than the L^∞ norm (Figure C.11).

GPN-MSA Captures Variant Functional Impact

A variant's impact on loss of fitness is mediated by genetic and functional pathways. To investigate whether GPN-MSA captures any functional impact of a variant, we performed functional enrichment analysis separately on four datasets curated across four public variant interpretation databases, ClinVar, COSMIC, OMIM and gnomAD. We used 18 functional annotations obtained from the FAVOR database [168] (accessed via Harvard Dataverse on April 10, 2023), which measure both impact of natural selection and gene regulatory activity of a variant (see Table C.2). For clarity, we collect computational details of the functional annotations and summarize them below.

- B Statistic [95], nucleotide diversity [46] and recombination rate [46] are mathematical quantities derived from evolutionary models, and are computed directly on the genomic position of the variant. They provide population-genetic interpretation of the impact of natural selection on the variant.
- Epigenetic tracks, RNA-seq, DNAse-seq, percent GC and percent CpG were all computed on genomic positions, to be included as training features in CADD [115]. Specifically, ENCODE track features are not gene-specific but are distributed as "bigWig"

value tracks along genomic coordinates. Values for each cell-type for which a track is available are summarized to create a new genome coordinate based track, which is subsequently assigned to the variant based on its genomic position. Whenever a variant is not annotatable for a track (e.g., RNA-seq level for a non-exonic variant), an `NA` value is assigned.

We found evidence of GPN-MSA capturing gene regulatory activity and impact of natural selection. Across all four datasets, significant negative correlations were observed between GPN-MSA and 8 histone mark levels (not including H3K9me3 and H3K27me3, which are recognized gene repressors; see Figure C.2). Additionally, GPN-MSA was positively correlated with nucleotide diversity and B statistic — for the both of which a smaller value indicates stronger impact of natural selection. In general, the strongest correlations of any annotation were observed in the dataset consisting of ClinVar pathogenic and gnomAD common missense variants.

Next, to investigate whether extreme values of GPN-MSA were associated with functional impact, we ran Mann-Whitney tests between the lowest (most deleterious) 1% GPN-MSA scoring ("target") variants and the remaining ("background") variants within each dataset, across all 18 annotations. Sample sizes were reasonably large between the target and background samples: the minimum sample size of any target set was 124. We found significant enrichment ($p < 0.05$ after controlling for FWER) of H4K20me1, a transcription activation mark, and RNA-seq levels in each dataset, and significant depletion of nucleotide diversity (Figure C.3). Interestingly, for H3K27me3, generally recognized as a gene repressor, all but the COSMIC pathogenic and gnomAD common missense dataset reported enrichment in the target variants. These results suggest that extremely negative GPN-MSA scores could potentially prioritize variants with impact on gene expression and regulation.

Code Availability

Code to reproduce all results is available at https://github.com/songlab-cal/gpn.

Data Availability

The processed whole-genome MSA is available at https://huggingface.co/datasets/songlab/multiz100way. The specific genomic windows used for training are available at https://huggingface.co/datasets/songlab/gpn-msa-sapiens-dataset. The variants used for benchmarking (including predictions) are available at https://huggingface.co/datasets/songlab/human_variants. Predictions for ∼530 million gnomAD variants are available at https://huggingface.co/datasets/songlab/gnomad. Predictions for all 9 billion possible single nucleotide variants in the human genome will be provided with the final, revised model upon publication. Predictions with the draft model can be performed with the Jupyter notebook at https://github.com/songlab-cal/gpn/blob/main/examples/msa/vep.ip

ynb. Sequence logos derived from GPN-MSA's predictions (currently available for chromosome 6 only) can be visualized at https://genome.ucsc.edu/s/gbenegas/gpn-msa-sapiens.

Model Availability

The pretrained model is available at https://huggingface.co/songlab/gpn-msa-sapiens.

Bibliography

[1] Carlos Alonso-Blanco et al. "1,135 genomes reveal the global pattern of polymorphism in *Arabidopsis thaliana*". In: *Cell* 166.2 (2016), pp. 481–491.

[2] Haley M Amemiya, Anshul Kundaje, and Alan P Boyle. "The ENCODE blacklist: identification of problematic regions of the genome". In: *Scientific Reports* 9.1 (2019), pp. 1–5.

[3] Simon Anders, Alejandro Reyes, and Wolfgang Huber. "Detecting differential usage of exons from RNA-seq data". In: *Genome Research* 22.10 (2012), pp. 2008–2017.

[4] Joel Armstrong et al. "Whole-genome alignment and comparative annotation". In: *Annual Review of Animal Biosciences* 7 (2019), pp. 41–64.

[5] Ángeles Arzalluz-Luque and Ana Conesa. "Single-cell RNAseq for the study of isoforms–how is that possible?" In: *Genome Biology* 19.1 (2018), p. 110.

[6] Aruna Asipu et al. "Properties of normal and mutant recombinant human ketohexokinases and implications for the pathogenesis of essential fructosuria". In: *Diabetes* 52.9 (2003), pp. 2426–2432.

[7] Žiga Avsec et al. "Effective gene expression prediction from sequence by integrating long-range interactions". In: *Nature Methods* 18.10 (2021), pp. 1196–1203.

[8] Zeheng Bai et al. "Identification of bacteriophage genome sequences with representation learning". In: *Bioinformatics* (Aug. 2022). btac509. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac509.

[9] Carlos Bas-Orth et al. "The calmodulin-binding transcription activator CAMTA1 is required for long-term memory formation in mice". In: *Learning & Memory* 23.6 (2016), pp. 313–321.

[10] Gonzalo Benegas, Sanjit Singh Batra, and Yun S. Song. "DNA language models are powerful predictors of genome-wide variant effects". In: *Proceedings of the National Academy of Sciences* 120.44 (2023), e2311219120.

[11] Gonzalo Benegas, Jonathan Fischer, and Yun S Song. "Robust and annotation-free analysis of alternative splicing across diverse cell types in mice". In: *Elife* 11 (2022), e73520.

[12] Gonzalo Benegas et al. "GPN-MSA: an alignment-based DNA language model for genome-wide variant effect prediction". In: *bioRxiv* (2023).

[13] Yoav Benjamini and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57.1 (1995), pp. 289–300.

[14] Volker Bergen et al. "Generalizing RNA velocity to transient cell states through dynamical modeling". In: *Nature Biotechnology* 38 (2020), pp. 1408–1414.

[15] Mathieu Blanchette et al. "Aligning multiple genomic sequences with the threaded blockset aligner". In: *Genome Research* 14.4 (2004), pp. 708–715.

[16] A Booeshaghi et al. "Isoform cell-type specificity in the mouse primary motor cortex". In: *Nature* 598.7879 (2021), pp. 195–199.

[17] S Borgeaud et al. "Improving language models by retrieving from trillions of tokens." In: *arXiv preprint arXiv:2112.04426* (2021).

[18] Claudia Bossen et al. "The chromatin remodeler Brg1 activates enhancer repertoires to establish B cell identity and modulate cell growth". In: *Nature Immunology* 16.7 (2015), pp. 775–784.

[19] Guillaume Bourque et al. "Ten things you should know about transposable elements". In: *Genome Biology* 19 (2018), pp. 1–12.

[20] Nadav Brandes, Omer Weissbrod, and Michal Linial. "Open problems in human trait genetics". In: *Genome Biology* 23.1 (2022), p. 131.

[21] Nadav Brandes et al. "Genome-wide prediction of disease variant effects with a deep protein language model". In: *Nature Genetics* (Oct. 2023). ISSN: 1546-1718. DOI: 10.1038/s41588-023-01465-0. URL: https://doi.org/10.1038/s41588-023-01465-0.

[22] Nicolas L Bray et al. "Near-optimal probabilistic RNA-seq quantification". In: *Nature Biotechnology* 34.5 (2016), pp. 525–527.

[23] Sébastien Bubeck et al. "Sparks of Artificial General Intelligence: Early experiments with GPT-4". In: *arXiv preprint arXiv:2303.12712* (2023).

[24] Carlos F Buen Abad Najar, Nir Yosef, and Liana F Lareau. "Coverage-dependent bias creates the appearance of binary splicing in single cells". In: *eLife* 9 (2020), e54603.

[25] Brendan K Bulik-Sullivan et al. "LD Score regression distinguishes confounding from polygenicity in genome-wide association studies". In: *Nature Genetics* 47.3 (2015), pp. 291–295.

[26] Ashley Byrne et al. "Nanopore long-read RNA-seq reveals widespread transcriptional variation among the surface receptors of individual B cells". In: *Nature Communications* 8.1 (2017), pp. 1–11.

[27] Kathleen M Chen et al. "A sequence-based global map of regulatory activity for deciphering human genetics". In: *Nature Genetics* 54.7 (2022), pp. 940–949.

[28] Siwei Chen et al. "A genome-wide mutational constraint map quantified from variation in 76,156 human genomes". In: *bioRxiv* (2022), pp. 2022–03.

[29] Jun Cheng et al. "MMSplice: modular modeling improves the predictions of genetic variant effects on splicing". In: *Genome Biology* 20.1 (2019), pp. 1–15.

[30] 1000 Genomes Project Consortium et al. "A global reference for human genetic variation". In: *Nature* 526.7571 (2015), p. 68.

[31] Richard J Cornall et al. "Role of Syk in B-cell development and antigen-receptor signaling". In: *Proceedings of the National Academy of Sciences* 97.4 (2000), pp. 1713–1718.

[32] Hugo Dalla-Torre et al. "The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics". In: *bioRxiv* (2023), pp. 2023–01.

[33] Jacob Devlin et al. "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[34] Alexander Dobin et al. "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1 (2013), pp. 15–21.

[35] Alexandre Evrard, Theogene Ndatimana, and Thomas Eulgem. "FORCA, a promoter element that responds to crosstalk between defense and light signaling". In: *BMC Plant Biology* 9 (2009), pp. 1–13.

[36] Huijuan Feng et al. "Complexity and graded regulation of neuronal cell-type–specific alternative splicing revealed by single-cell RNA sequencing". In: *Proceedings of the National Academy of Sciences* 118.10 (2021), e2013056118.

[37] Mikael Feracci et al. "Structural basis of RNA recognition and dimerization by the STAR proteins T-STAR and Sam68". In: *Nature communications* 7.1 (2016), pp. 1–12.

[38] Veniamin Fishman et al. "GENA-LM: A Family of Open-Source Foundational Models for Long DNA Sequences". In: *bioRxiv* (2023), pp. 2023–06.

[39] Alyssa C Frazee et al. "Polyester: simulating RNA-seq datasets with differential transcript expression". In: *Bioinformatics* 31.17 (2015), pp. 2778–2784.

[40] Jonathan Frazer et al. "Disease variant prediction with deep generative models of evolutionary data". In: *Nature* 599.7883 (2021), pp. 91–95.

[41] Marc V Fuccillo et al. "Single-cell mRNA profiling reveals cell-type-specific expression of neurexin isoforms". In: *Neuron* 87.2 (2015), pp. 326–340.

[42] Geoff Fudenberg, David R Kelley, and Katherine S Pollard. "Predicting 3D genome folding from DNA sequence with Akita". In: *Nature Methods* 17.11 (2020), pp. 1111–1117.

[43] Azahara-Maria Garcia-Serna et al. "Dock10 regulates CD23 expression and sustains B-cell lymphopoiesis in secondary lymphoid tissue". In: *Immunobiology* 221.12 (2016), pp. 1343–1350.

[44] Diego Garrido-Martın et al. "ggsashimi: Sashimi plot revised for browser-and annotation-independent splicing visualization". In: *PLoS Computational Biology* 14.8 (2018), e1006360.

[45] Adam Gayoso et al. "scvi-tools: a library for deep probabilistic analysis of single-cell omics data". In: *bioRxiv preprint* (2021). URL: https://doi.org/10.1101/2021.04.28.441833.

[46] Steven Gazal et al. "Linkage disequilibrium–dependent architecture of human complex traits shows action of negative selection". In: *Nature Genetics* 49.10 (2017), pp. 1421–1427.

[47] Laura H. Goetz and Nicholas J. Schork. "Personalized medicine: motivation, challenges, and progress". In: *Fertility and Sterility* 109.6 (2018), pp. 952–963. ISSN: 0015-0282. DOI: https://doi.org/10.1016/j.fertnstert.2018.05.006. URL: https://www.sciencedirect.com/science/article/pii/S0015028218304072.

[48] Rachel L Goldfeder et al. "Human genome sequencing at the population scale: a primer on high-throughput DNA sequencing and analysis". In: *American Journal of Epidemiology* 186.8 (2017), pp. 1000–1009.

[49] Albert Gu, Karan Goel, and Christopher Re. "Efficiently Modeling Long Sequences with Structured State Spaces". In: *International Conference on Learning Representations*. 2021.

[50] Ishaan Gupta et al. "Single-cell isoform RNA sequencing characterizes isoforms in thousands of cerebellar cells". In: *Nature Biotechnology* 36.12 (2018), pp. 1197–1202.

[51] Shobhit Gupta et al. "Quantifying similarity between motifs". In: *Genome Biology* 8.2 (2007), pp. 1–9.

[52] Ho-Jin Gwak and Mina Rho. "ViBE: a hierarchical BERT model to identify eukaryotic viruses using metagenome sequencing data". In: *Briefings in Bioinformatics* 23.4 (June 2022). bbac204. ISSN: 1477-4054. DOI: 10.1093/bib/bbac204.

[53] Penelope R Haddrill. "Developments in forensic DNA analysis". In: *Emerging Topics in Life Sciences* 5.3 (2021), pp. 381–393.

[54] Michael Hagemann-Jensen et al. "Single-cell RNA counting at allele and isoform resolution using Smart-seq3". In: *Nature Biotechnology* 38.6 (2020), pp. 708–714.

[55] Bruce E Hayward and David T Bonthron. "Structure and alternative splicing of the ketohexokinase gene". In: *European journal of biochemistry* 257.1 (1998), pp. 85–91.

[56] A Hoarfrost et al. "Deep learning of a bacterial and archaeal universal language of life enables transfer learning and illuminates microbial dark matter". In: *Nature Communications* 13.1 (2022), pp. 1–12.

[57] Hui Hu et al. "Foxp1 is an essential transcriptional regulator of B cell development". In: *Nature Immunology* 7.8 (2006), pp. 819–826.

[58] Yu Hu, Kai Wang, and Mingyao Li. "Detecting differential alternative splicing events in scRNA-seq with or without Unique Molecular Identifiers". In: *PLOS Computational Biology* 16.6 (2020), e1007925.

[59] Yuanhua Huang and Guido Sanguinetti. "BRIE: transcriptome-wide splicing quantification in single cells". In: *Genome Biology* 18.1 (2017), p. 123.

[60] Yuanhua Huang and Guido Sanguinetti. "BRIE2: computational identification of splicing phenotypes from single-cell transcriptomic experiments". In: *Genome Biology* 22.1 (2021), pp. 1–15.

[61] Kishore Jaganathan et al. "Predicting splicing from primary sequence with deep learning". In: *Cell* 176.3 (2019), pp. 535–548.

[62] Milind Jagota et al. "Cross-protein transfer learning substantially improves disease variant prediction". In: *Genome Biology* 24.1 (2023), pp. 1–19.

[63] Yanrong Ji et al. "DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome". In: *Bioinformatics* 37.15 (2021), pp. 2112–2120.

[64] Anoushka Joglekar et al. "A spatially resolved brain region-and cell type-specific isoform atlas of the postnatal mouse brain". In: *Nature Communications* 12.1 (2021), pp. 1–16.

[65] Alan M Jones et al. "The impact of Arabidopsis on human health: diversifying our portfolio". In: *Cell* 133.6 (2008), pp. 939–943.

[66] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[67] Minghui Kang et al. "The pan-genome and local adaptation of Arabidopsis thaliana". In: *bioRxiv* (2022), pp. 2022–12.

[68] Jared Kaplan et al. "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361* (2020).

[69] W James Kent et al. "The human genome browser at UCSC". In: *Genome Research* 12.6 (2002), pp. 996–1006.

[70] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. 2015. URL: https://arxiv.org/abs/1412.6980.

[71] Diederik P Kingma and Max Welling. "Auto-encoding variational Bayes". In: *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. 2014. URL: https://arxiv.org/abs/1312.6114.

[72] Johannes Köster and Sven Rahmann. "Snakemake—a scalable bioinformatics workflow engine". In: *Bioinformatics* 28.19 (2012), pp. 2520–2522.

[73] Gioele La Manno et al. "RNA velocity of single cells". In: *Nature* 560.7719 (2018), pp. 494–498.

[74] Melissa J Landrum et al. "ClinVar: improvements to accessing data". In: *Nucleic Acids Research* 48.D1 (2020), pp. D835–D844.

[75] Derek Le Roith, Joseph Shiloach, and Jesse Roth. "Is there an earlier phylogenetic precursor that is common to both the nervous and endocrine systems?" In: *Peptides* 3.3 (1982), pp. 211–215.

[76] Kevin Lebrigand et al. "High throughput error corrected Nanopore single cell transcriptome sequencing". In: *Nature Communications* 11.1 (2020), pp. 1–8.

[77] Dongwon Lee et al. "A method to predict the impact of regulatory variants from DNA sequence". In: *Nature Genetics* 47.8 (2015), pp. 955–961.

[78] Seunggeun Lee et al. "Rare-variant association analysis: study designs and statistical tests". In: *The American Journal of Human Genetics* 95.1 (2014), pp. 5–23.

[79] Fernanda O Lemos, Mateus T Guerra, and M Fátima Leite. "Inositol 1, 4, 5 trisphosphate receptors in secretory epithelial cells of the gastrointestinal tract". In: *Current Opinion in Physiology* (2020).

[80] Bo Li and Colin N Dewey. "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome". In: *BMC bioinformatics* 12.1 (2011), pp. 1–16.

[81] Yang I Li et al. "Annotation-free quantification of RNA splicing using LeafCutter". In: *Nature Genetics* 50.1 (2018), p. 151.

[82] Yang Liao, Gordon K Smyth, and Wei Shi. "featureCounts: an efficient general purpose program for assigning sequence reads to genomic features". In: *Bioinformatics* 30.7 (2014), pp. 923–930.

[83] Zeming Lin et al. "Evolutionary-scale prediction of atomic-level protein structure with a language model". In: *Science* 379.6637 (2023), pp. 1123–1130.

[84] Jonathan P Ling et al. "ASCOT identifies key regulators of neuronal subtype-specific splicing". In: *Nature Communications* 11.1 (2020), pp. 1–12.

[85] Dong C Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45.1-3 (1989), pp. 503–528.

[86] Shang Liu et al. "Single-cell differential splicing analysis reveals high heterogeneity of liver tumor-infiltrating T cells". In: *Scientific Reports* 11.1 (2021), pp. 1–12.

[87] Romain Lopez, Adam Gayoso, and Nir Yosef. "Enhancing scientific discoveries in molecular biology with deep generative models". In: *Molecular Systems Biology* 16.9 (2020), e9198.

[88] Zhaogeng Lu et al. "Identification and characterization of novel lncRNAs in Arabidopsis thaliana". In: *Biochemical and Biophysical Research Communications* 488.2 (2017), pp. 348–354.

[89] Carla Márquez-Luna et al. "Incorporating functional priors improves polygenic prediction accuracy in UK Biobank and 23andMe data sets". In: *Nature Communications* 12.1 (2021), p. 6052.

[90] Marcel Martin. "Cutadapt removes adapter sequences from high-throughput sequencing reads". In: *EMBnet.journal* 17.1 (2011), pp. 10–12.

[91] Shruti Marwaha, Joshua W Knowles, and Euan A Ashley. "A guide for the diagnosis of rare and undiagnosed disease: beyond the exome". In: *Genome Medicine* 14.1 (2022), pp. 1–22.

[92] Hirotaka Matsumoto et al. "An NMF-based approach to discover overlooked differentially expressed gene regions from single-cell RNA-seq data". In: *NAR Genomics and Bioinformatics* 2.1 (2020), lqz020.

[93] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[94] William McLaren et al. "The Ensembl Variant Effect Predictor". In: *Genome Biology* 17.1 (2016), pp. 1–14.

[95] Graham McVicker et al. "Widespread genomic signatures of natural selection in hominid evolution". In: *PLoS Genetics* 5.5 (2009), e1000471.

[96] Colin Megill et al. "cell×gene: a performant, scalable exploration platform for high dimensional sparse matrices". In: *bioRxiv preprint* (2021). eprint: https://www.biorxiv.org/content/early/2021/04/06/2021.04.05.438318.full.pdf. URL: https://doi.org/10.1101/2021.04.05.438318.

[97] Joshua Meier et al. "Language models enable zero-shot prediction of the effects of mutations on protein function". In: *Advances in Neural Information Processing Systems* 34 (2021).

[98] Shentong Mo et al. "Multi-modal Self-supervised Pre-training for Large-scale Genome Data". In: *NeurIPS 2021 AI for Science Workshop*. 2021.

[99] Surag Nair et al. "The dynseq browser track shows context-specific features at nucleotide resolution". In: *Nature Genetics* 54.11 (2022), pp. 1581–1583.

[100] Eric Nguyen et al. "HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution". In: *arXiv preprint arXiv:2306.15794* (2023).

[101] Ka Ming Nip et al. "RNA-Bloom enables reference-free and reference-guided sequence assembly for single-cell transcriptomes". In: *Genome Research* 30.8 (2020), pp. 1191–1200.

[102] Vasilis Ntranos et al. "A discriminative learning approach to differential expression analysis for single-cell RNA-seq". In: *Nature methods* 16.2 (2019), pp. 163–166.

[103] Julia Eve Olivieri, Roozbeh Dehghannasiri, and Julia Salzman. "The SpliZ generalizes "Percent Spliced In" to reveal regulated splicing at single-cell resolution". In: *bioRxiv preprint* (2021). URL: https://doi.org/10.1101/2020.11.10.377572.

[104] Badri Padhukasahasram. "Inferring ancestry from population genomic data and its applications". In: *Frontiers in genetics* 5 (2014), p. 204.

[105] Adam Paszke et al. "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems*. 2019, pp. 8026–8037.

[106] Ralph Patrick et al. "Sierra: Discovery of differential transcript usage from polyA-captured single-cell RNA-seq data". In: *Genome Biology* 21.1 (2020), pp. 1–27.

[107] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[108] Simona Pedrotti et al. "The RNA-binding protein Rbfox1 regulates splicing required for skeletal muscle structure and function". In: *Human molecular genetics* 24.8 (2015), pp. 2360–2374.

[109] Simone Picelli et al. "Full-length RNA-seq from single cells using Smart-seq2". In: *Nature Protocols* 9.1 (2014), p. 171.

[110] Katherine S Pollard et al. "Detection of nonneutral substitution rates on mammalian phylogenies". In: *Genome Research* 20.1 (2010), pp. 110–121.

[111] Xiaojie Qiu et al. "Single-cell mRNA quantification and differential analysis with Census". In: *Nature Methods* 14.3 (2017), pp. 309–315.

[112] Daniel Quang, Yifei Chen, and Xiaohui Xie. "DANN: a deep learning approach for annotating the pathogenicity of genetic variants". In: *Bioinformatics* 31.5 (2015), pp. 761–763.

[113] Roshan M Rao et al. "MSA Transformer". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8844–8856.

[114] Philipp Rentzsch et al. "CADD-Splice—improving genome-wide variant effect prediction using deep learning-derived splice scores". In: *Genome Medicine* 13.1 (2021), pp. 1–12.

[115] Philipp Rentzsch et al. "CADD: predicting the deleteriousness of variants throughout the human genome". In: *Nucleic Acids Research* 47.D1 (2019), pp. D886–D894.

[116] Alexander Rives et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.

[117] Nicholas Schaum et al. "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris: The Tabula Muris Consortium". In: *Nature* 562.7727 (2018), p. 367.

[118] Thomas D Schneider and R Michael Stephens. "Sequence logos: a new way to display consensus sequences". In: *Nucleic Acids Research* 18.20 (1990), pp. 6097–6100.

[119] Martin Schüle et al. "mTOR driven gene transcription is required for cholesterol production in neurons of the developing cerebral cortex". In: *International Journal of Molecular Sciences* 22.11 (2021), p. 6034.

[120] Ameet S Sengar et al. "Control of long-term synaptic potentiation and learning by alternative splicing of the NMDA receptor subunit GluN1". In: *Cell Reports* 29.13 (2019), pp. 4285–4294.

[121] Avanti Shrikumar et al. "Technical note on transcription factor motif discovery from importance scores (TF-MoDISco) version 0.5. 6.5". In: *arXiv preprint arXiv:1811.00416* (2018).

[122] Adam Siepel et al. "Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes". In: *Genome Research* 15.8 (2005), pp. 1034–1050.

[123] Damian Smedley et al. "A whole-genome analysis framework for effective identification of pathogenic regulatory variants in Mendelian disease". In: *The American Journal of Human Genetics* 99.3 (2016), pp. 595–606.

[124] Yan Song et al. "Single-cell alternative splicing analysis with expedition reveals splicing dynamics during neuron differentiation". In: *Molecular Cell* 67.1 (2017), pp. 148–161.

[125] Merle Stein et al. "A defined metabolic state in pre B cells governs B-cell development and is counterbalanced by Swiprosin-2/EFhd1". In: *Cell Death & Differentiation* 24.7 (2017), pp. 1239–1252.

[126] Tim Stuart et al. "Comprehensive Integration of Single-Cell Data". In: *Cell* 177 (2019), pp. 1888–1902.

[127] Jianlin Su et al. "Roformer: Enhanced transformer with rotary position embedding". In: *arXiv preprint arXiv:2104.09864* (2021).

[128] Patrick F Sullivan et al. "Leveraging base-pair mammalian constraint to understand genetic variation and human disease". In: *Science* 380.6643 (2023), eabn2937.

[129] Valentine Svensson et al. "Interpretable factor models of single-cell RNA-seq via variational autoencoders". In: *Bioinformatics* 36.11 (2020), pp. 3418–3421.

[130] Karol Szlachta et al. "Alternative DNA secondary structure formation affects RNA polymerase II promoter-proximal pausing in human". In: *Genome Biology* 19 (2018), pp. 1–19.

[131] Tabula Muris Consortium. "A single-cell transcriptomic atlas characterizes ageing tissues in the mouse". In: *Nature* 583.7817 (2020), pp. 590–595.

[132] John G Tate et al. "COSMIC: the catalogue of somatic mutations in cancer". In: *Nucleic Acids Research* 47.D1 (2019), pp. D941–D947.

[133] Yi Tay et al. "Are Pretrained Convolutions Better than Pretrained Transformers?" In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4349–4359. DOI: 10.18653/v1/2021.acl-long.335.

[134] Tobias Tekath and Martin Dugas. "Differential transcript usage analysis of bulk and single-cell RNA-seq data with DTUrtle". In: *Bioinformatics* 37.21 (2021), pp. 3781–3787.

[135] Feng Tian et al. "PlantRegMap: charting functional regulatory maps in plants". In: *Nucleic Acids Research* 48.D1 (2020), pp. D1104–D1113.

[136] Laura Tibbs Cortes, Zhiwu Zhang, and Jianming Yu. "Status and prospects of genome-wide association studies in plants". In: *The Plant Genome* 14.1 (2021), e20077.

[137] Matteo Togninalli et al. "AraPheno and the AraGWAS Catalog 2020: a major database update including RNA-Seq and knockout mutation data for *Arabidopsis thaliana*". In: *Nucleic Acids Research* 48.D1 (2020), pp. D1063–D1068.

[138] Matteo Togninalli et al. "The AraGWAS Catalog: a curated and standardized Arabidopsis thaliana GWAS catalog". In: *Nucleic Acids Research* 46.D1 (2018), pp. D1150–D1156.

[139] Katerina Trajanoska et al. "From target discovery to clinical drug development with human genetics". In: *Nature* 620.7975 (Oct. 2023), pp. 737–745. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06388-8. URL: https://doi.org/10.1038/s41586-023-06388-8.

[140] Lisa Traunmüller et al. "Control of neuronal synapse specification by a highly dedicated alternative splicing program". In: *Science* 352.6288 (2016), pp. 982–986.

[141] Ana Urzainqui et al. "Relevance of PSGL-1 expression in B cell development and activation". In: *Frontiers in Immunology* 11 (2020), p. 2900.

[142] Jorge Vaquero-Garcia et al. "A new view of transcriptome complexity and regulation through the lens of local splicing variations". In: *eLife* 5 (2016), e11752.

[143] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[144] Peter M Visscher et al. "10 years of GWAS discovery: biology, function, and translation". In: *The American Journal of Human Genetics* 101.1 (2017), pp. 5–22.

[145] Roger Volden and Christopher Vollmers. "Highly multiplexed single-cell full-length CDNA sequencing of human immune cells with 10X genomics and R2C2". In: *bioRxiv preprint* (2020). URL: https://doi.org/10.1101/2020.01.10.902361.

[146] Jakob Von Engelhardt et al. "CKAMP44: a brain-specific protein attenuating short-term synaptic plasticity in the dentate gyrus". In: *Science* 327.5972 (2010), pp. 1518–1522.

[147] Celine K Vuong, Douglas L Black, and Sika Zheng. "The neurogenetics of alternative splicing". In: *Nature Reviews Neuroscience* 17.5 (2016), pp. 265–281.

[148] Brie Wamsley et al. "Rbfox1 mediates cell-type-specific splicing in cortical interneurons". In: *Neuron* 100.4 (2018), pp. 846–859.

[149] Eric T Wang et al. "Alternative isoform regulation in human tissue transcriptomes". In: *Nature* 456.7221 (2008), pp. 470–476.

[150] Qingqing Wang and Donald C Rio. "JUM is a computational method for comprehensive annotation-free analysis of alternative pre-mRNA splicing patterns". In: *Proceedings of the National Academy of Sciences* 115.35 (2018), E8181–E8190.

[151] Xiliang Wang et al. "Direct comparative analyses of 10X Genomics Chromium and smart-seq2". In: *Genomics, Proteomics & Bioinformatics* (2021).

[152] Daniel J Weiner et al. "Polygenic architecture of rare coding variation across 394,783 exomes". In: *Nature* 614.7948 (2023), pp. 492–499.

[153] Omer Weissbrod et al. "Functionally informed fine-mapping and polygenic localization of complex trait heritability". In: *Nature Genetics* 52.12 (2020), pp. 1355–1363.

[154] Joshua D Welch, Yin Hu, and Jan F Prins. "Robust detection of alternative splicing in a population of single cells". In: *Nucleic Acids Research* 44.8 (2016), e73–e73.

[155] Wei Xiong Wen, Adam J Mead, and Supat Thongjuea. "VALERIE: Visual-based inspection of alternative splicing events at single-cell resolution". In: *PLOS Computational Biology* 16.9 (2020), e1008195.

[156] Jennifer Westoby et al. "Obstacles to detecting isoforms using full-length scRNA-seq data". In: *Genome Biology* 21.1 (2020), pp. 1–19.

[157] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome Biology* 19.1 (2018), pp. 1–5.

[158] Thomas Wolf et al. "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: *arXiv preprint arXiv:1910.03771* (2019).

[159] Qinghong Yan et al. "Systematic discovery of regulated and conserved alternative exons in the mammalian brain reveals NMD modulating chromatin regulators". In: *Proceedings of the National Academy of Sciences* 112.11 (2015), pp. 3445–3450.

[160] Kevin K Yang, Alex Xijie Lu, and Nicolo Fusi. "Convolutions are competitive with transformers for protein sequence pretraining". In: *ICLR2022 Machine Learning for Drug Discovery*. 2022. URL: https://openreview.net/forum?id=3i7WPak2sCx.

[161] Meng Yang et al. "Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution". In: *Nucleic Acids Research* 50.14 (2022), e81–e81.

[162] Zizhen Yao et al. "A transcriptomic and epigenomic cell atlas of the mouse primary motor cortex". In: *Nature* 598.7879 (2021), pp. 103–110.

[163] Gene Yeo et al. "Variation in alternative splicing across human tissues". In: *Genome Biology* 5.10 (2004), pp. 1–15.

[164] Manzil Zaheer et al. "Big Bird: Transformers for longer sequences". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17283–17297.

[165] Syed Shan-e-Ali Zaidi et al. "Engineering crops of the future: CRISPR approaches to develop climate-resilient and disease-resistant plants". In: *Genome biology* 21.1 (2020), pp. 1–19.

[166] Haoyang Zeng and David K Gifford. "Predicting the impact of non-coding variants on DNA methylation". In: *Nucleic Acids Research* 45.11 (2017), e99–e99.

[167] Hannah Zhou, Avanti Shrikumar, and Anshul Kundaje. "Towards a better understanding of reverse-complement equivariance for deep learning models in genomics". In: *Machine Learning in Computational Biology*. PMLR. 2022, pp. 1–33.

[168] Hufeng Zhou et al. "FAVOR: functional annotation of variants online resource and annotator for variation across the human genome". In: *Nucleic Acids Research* 51.D1 (2023), pp. D1300–D1311.

[169] Jian Zhou and Olga G Troyanskaya. "Predicting effects of noncoding variants with deep learning–based sequence model". In: *Nature Methods* 12.10 (2015), pp. 931–934.

[170] Wanding Zhou et al. "DNA methylation enables transposable element-driven genome expansion". In: *Proceedings of the National Academy of Sciences* 117.32 (2020), pp. 19359–19366.

Appendix A

Supplementary Information for Chapter 2

Overview of available methods for alternative splicing analysis in full-length scRNA-seq data

Due to experimental considerations, the analysis of transcript variation in 10x Chromium data is mostly restricted to the 3' end of genes; in contrast, Smart-seq2 and other full-length, short-read protocols theoretically enable characterization of transcript variation along the whole gene. Nevertheless, numerous challenges impede such analyses in practice. For example, low transcript capture efficiency introduces additional technical noise into transcript quantification [5, 156, 24], and incomplete transcriptome annotations result in discarded reads and reduced sensitivity to cross-cell differences [156]. Some authors have even recommended avoiding the analysis of alternative splicing in single-cell RNA sequencing (scRNA-seq) data until such obstacles can be suitably overcome [156]. Despite these difficulties, several methods (summarized in Table A.1) have sought to analyze transcript variation in short-read, full-length scRNA-seq. Many methods, including kallisto [22], Census [111], BRIE [59], SCATS [58], Quantas [159], VALERIE (meant only for visualization) [155], DESJ [86], BRIE2 [60] and DTUrtle [134], depend on transcript annotations and consequently cannot easily identify unannotated alternative splicing events, which may comprise a sizable fraction of events. Currently available annotation-free methods, such as ODEGR-NMF [92], Expedition [124], ASCOT [84], SingleSplice [154] and RNA-Bloom [101], do not provide a statistical test for differential transcript usage across conditions. Table A.1 summarizes this information and makes the comparison of different methods easier.

Table A.1: **Summary of methods available to analyze transcript variation in short-read full-length scRNA-seq.** *Annotation-free*: Does quantification require an accurate transcriptome reference? *Differential transcript usage*: Does the method provide a two-sample test for differences in transcript proportions? Some methods, denoted by (*), provide other statistical tests. Quantas requires cells to be aggregated into known subgroups of each group and therefore does not perform a test at the single-cell level. SingleSplice tests for alternative splicing within a single population. kallisto and ODEGR-NMF test for differential transcript expression, i.e., changes in absolute transcript expression rather than their proportions. Census tests for differential transcript usage along a pseudotime trajectory.

Method	Annotation-free	Differential transcript usage
Quantas [159]		*
SingleSplice [154]	✓	*
kallisto [22]		*
Census [111]		*
BRIE [59]		✓
Expedition [124]	✓	
ODEGR-NMF [92]	✓	*
SCATS [58]		✓
RNA-Bloom [101]	✓	
ASCOT [84]	✓	
DESJ [86]		✓
BRIE2 [60]		✓
DTUrtle [134]		✓
scQuint	✓	✓

Supplementary Figures

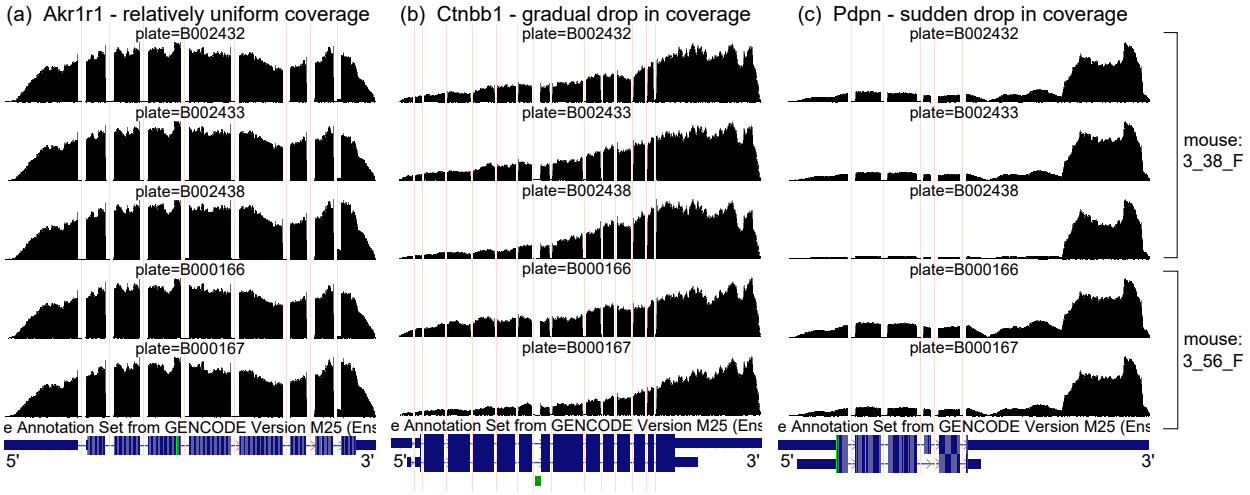


Figure A.1: **Coverage artifacts in mammary gland basal cells from *Tabula Muris*.** Aggregate read coverage of basal cells is shown for three genes in two female mice: *3_38_F*, processed in three different plates, and *3_56_F*, processed in two different plates. Visualization on the UCSC Genome Browser. **(a)** *Akr1r1*, with relatively uniform coverage, what we expect. **(b)** *Ctnbb1*, with a gradual drop in coverage away from the 3' end. The rate of coverage decay varies across plates. **(c)** *Pdpn*, with a sudden drop in coverage halfway through the 3' UTR. The magnitude of the drop varies across plates.

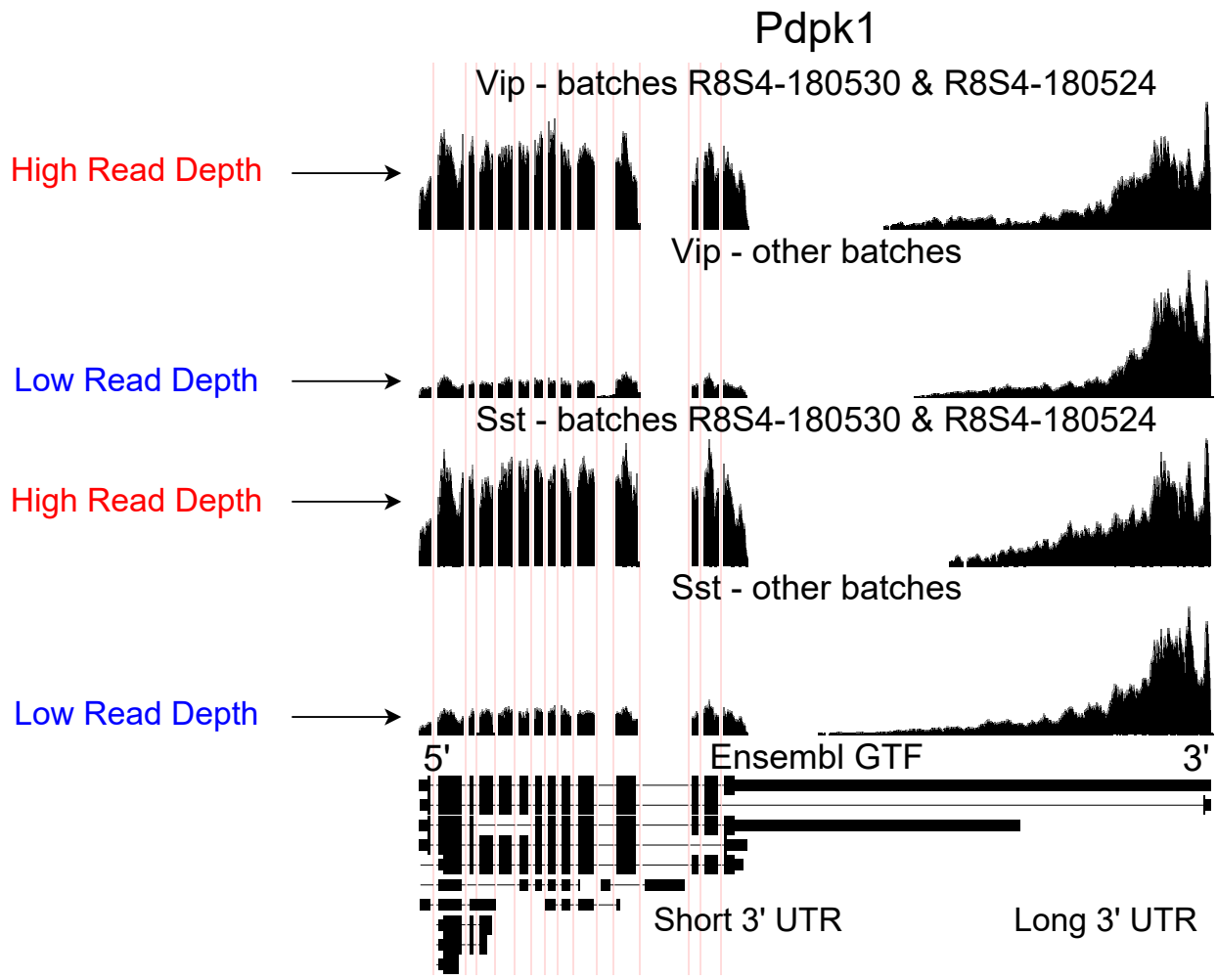


Figure A.2: **Technical artifacts in *BICCN Cortex*.** Aggregate read coverage in *Pdpk1* in two cell types, Vip and Sst, further separated according to the "batch" metadata label into two groups. The first group contains cells from batches *R8S4-180530* and *R8S4-180524*, while the second group contains the remaining batches. Cells from different batches belong to different mice and were processed on different dates. In all groups, coverage decreases rapidly in the 3' UTR of the isoform with the longest 3' UTR, eventually reaching zero. Additionally, the relative coverage in this region compared to the rest of the gene (seeming to originate from a different isoform with a short 3' UTR) varies drastically across batches, and consistently in different cell types. In principle, this could be due to biological differences between mice from different batches, but an explanation based on technical factors such as amplification bias may be more plausible.

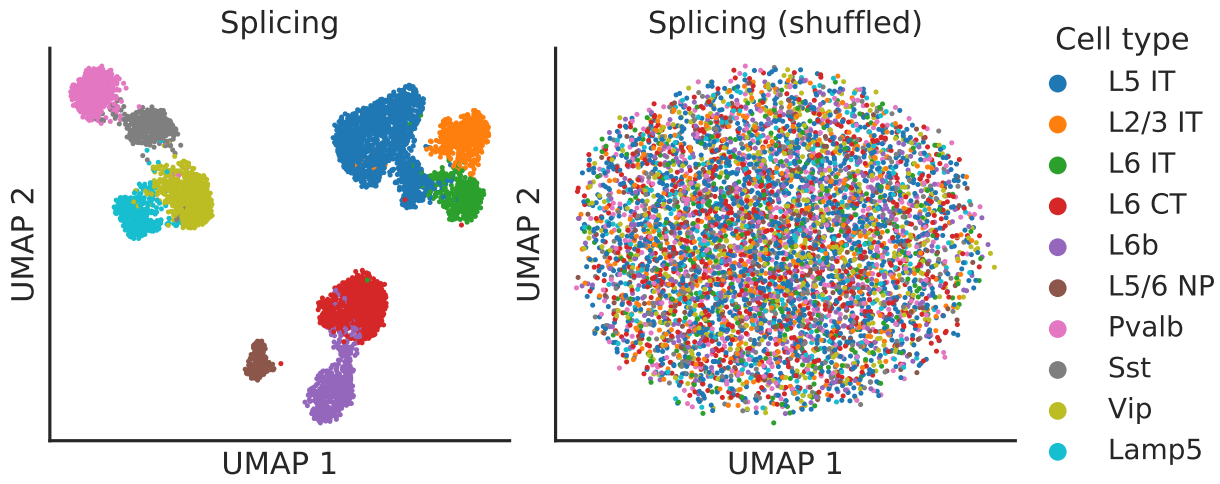


Figure A.3: **Splicing latent space when alternative intron counts are shuffled.** To verify that absolute gene expression does not affect the splicing latent space, we perturbed the *BICCN Cortex* data set by resampling alternative intron counts with a fixed proportion in all cells (the proportions in different alternative intron groups varied and were sampled from a uniform Dirichlet distribution). In this scenario, different cell types still vary in their gene expression levels but not in their splicing patterns. As hoped, the splicing latent space does not distinguish between cell types, indicating it is only capturing differences in splicing proportions rather than changes in absolute gene expression.

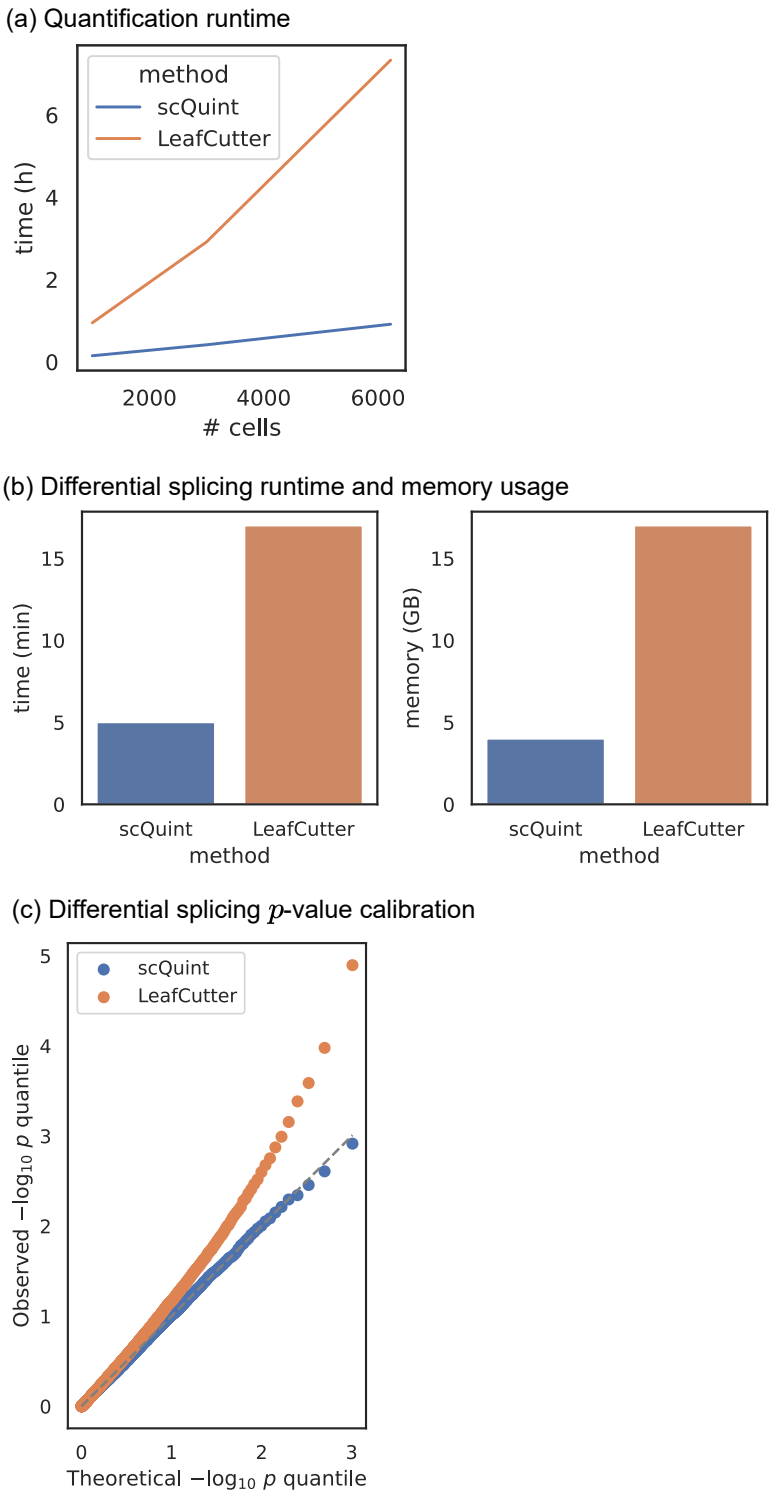


Figure A.4: **Comparison with LeafCutter.**

Figure A.4 *(continued)*: (a) Quantification runtime. Time to perform intron quantification on *BICCN Cortex* dataset, including cell subsampling to understand effect of number of cells.
(b) Differential splicing runtime and memory usage. We randomly split all 6220 *BICCN Cortex* cells into two equally sized groups and performed differential splicing between them. Runtime (left) and memory usage (right) are displayed.
(c) Differential splicing p-value calibration. In the same random split of (b), the null hypothesis of no difference in splicing proportions holds, and we expect the distribution of p-values to be uniform. The quantile-quantile plot of p-values obtained with scQuint shows their distribution is indeed uniform, suggesting that the model is well-calibrated under the null; this is not true for p-values obtained by LeafCutter.

All experiments were performed on a Skylake processor (2x16 cores @ 2.1 GHz) with 96 GB of RAM.

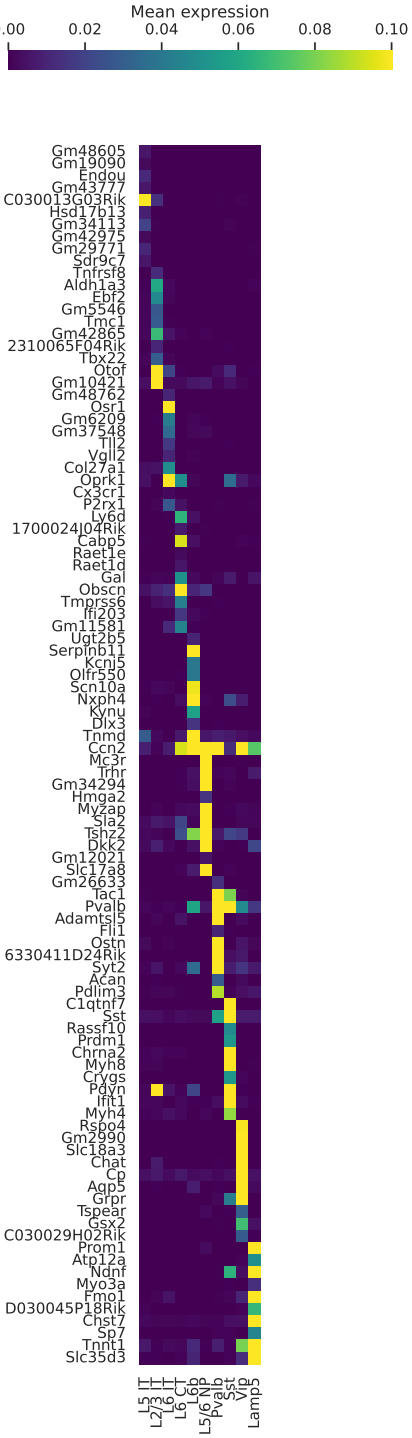


Figure A.5: **Marker genes for cell types in *BICCN Cortex*.** Mean (log-transformed) expression for some of the top differentially expressed genes in each cell type.

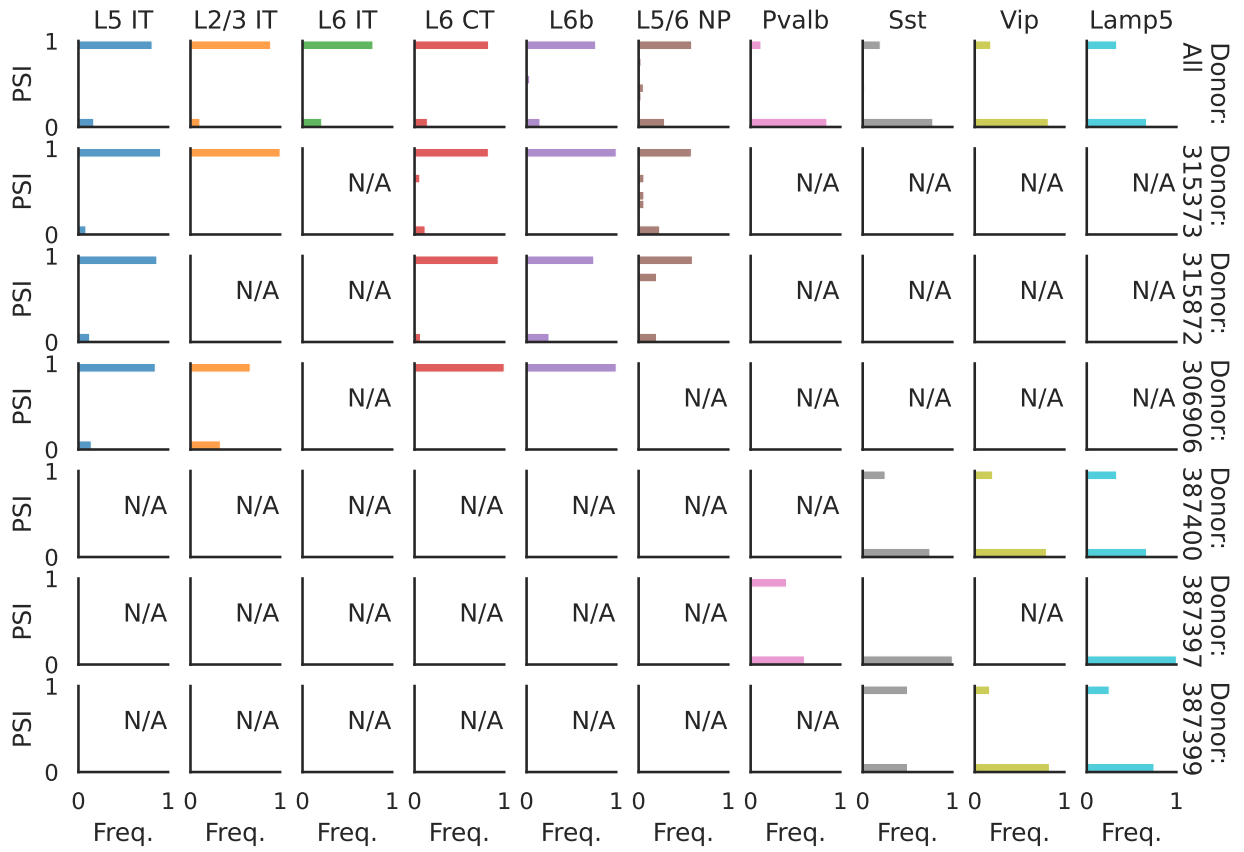


Figure A.6: **PSI distribution of `Pgm2_32951`.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells. Per the experimental design of this dataset, the top 3 individuals have only Glutamatergic cell types sequenced, while the bottom 3 have only GABAergic.

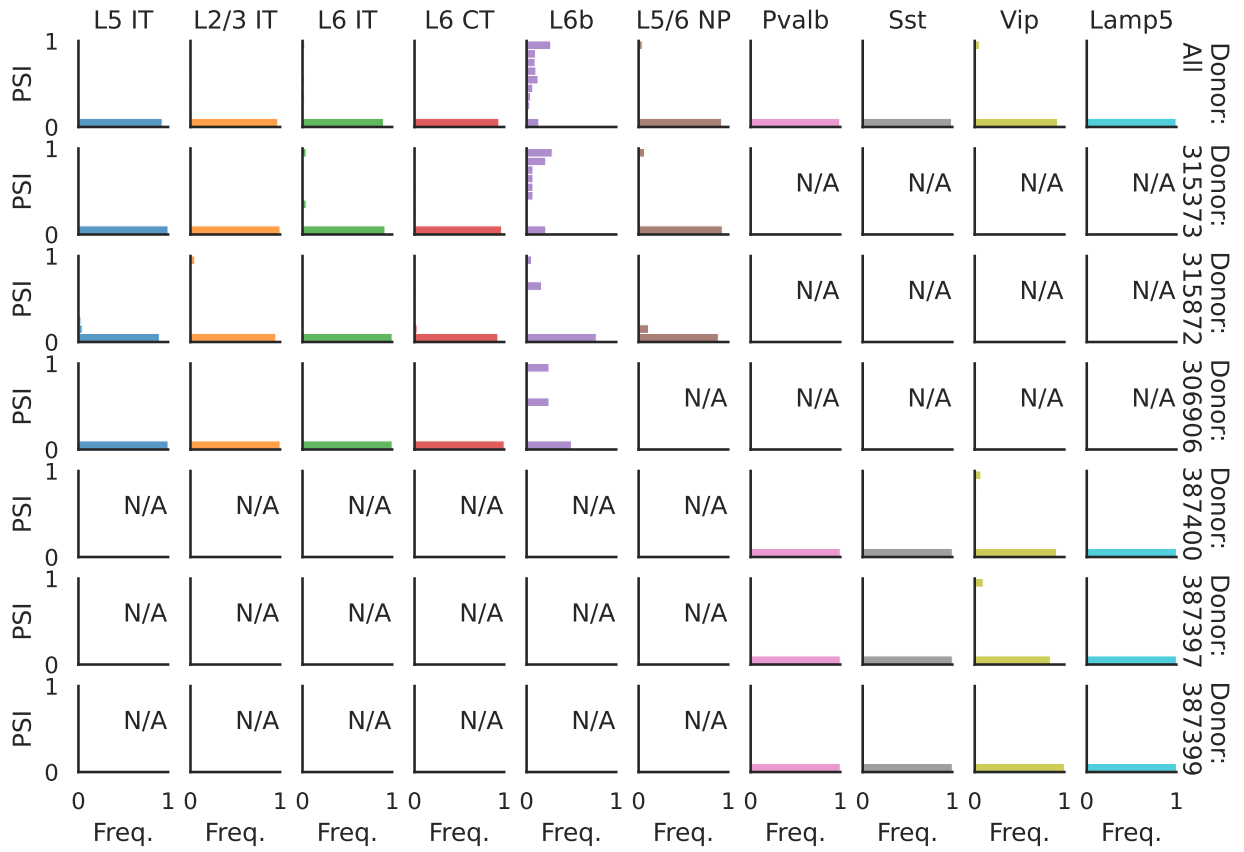


Figure A.7: **PSI distribution of `Rbfox1_26172`.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells. Per the experimental design of this dataset, the top 3 individuals have only Glutamatergic cell types sequenced, while the bottom 3 have only GABAergic.

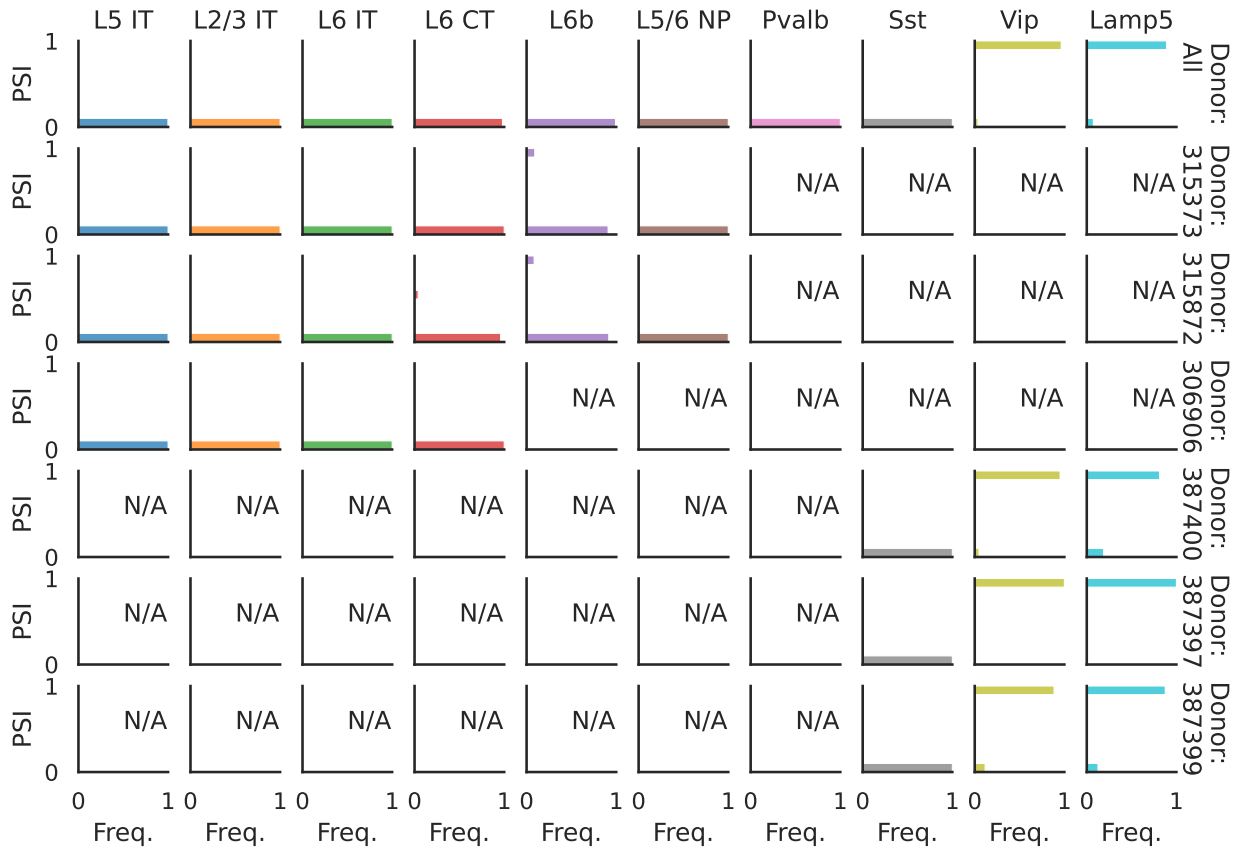


Figure A.8: **PSI distribution of `Nrxn1_8067`.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells. Per the experimental design of this dataset, the top 3 individuals have only Glutamatergic cell types sequenced, while the bottom 3 have only GABAergic.

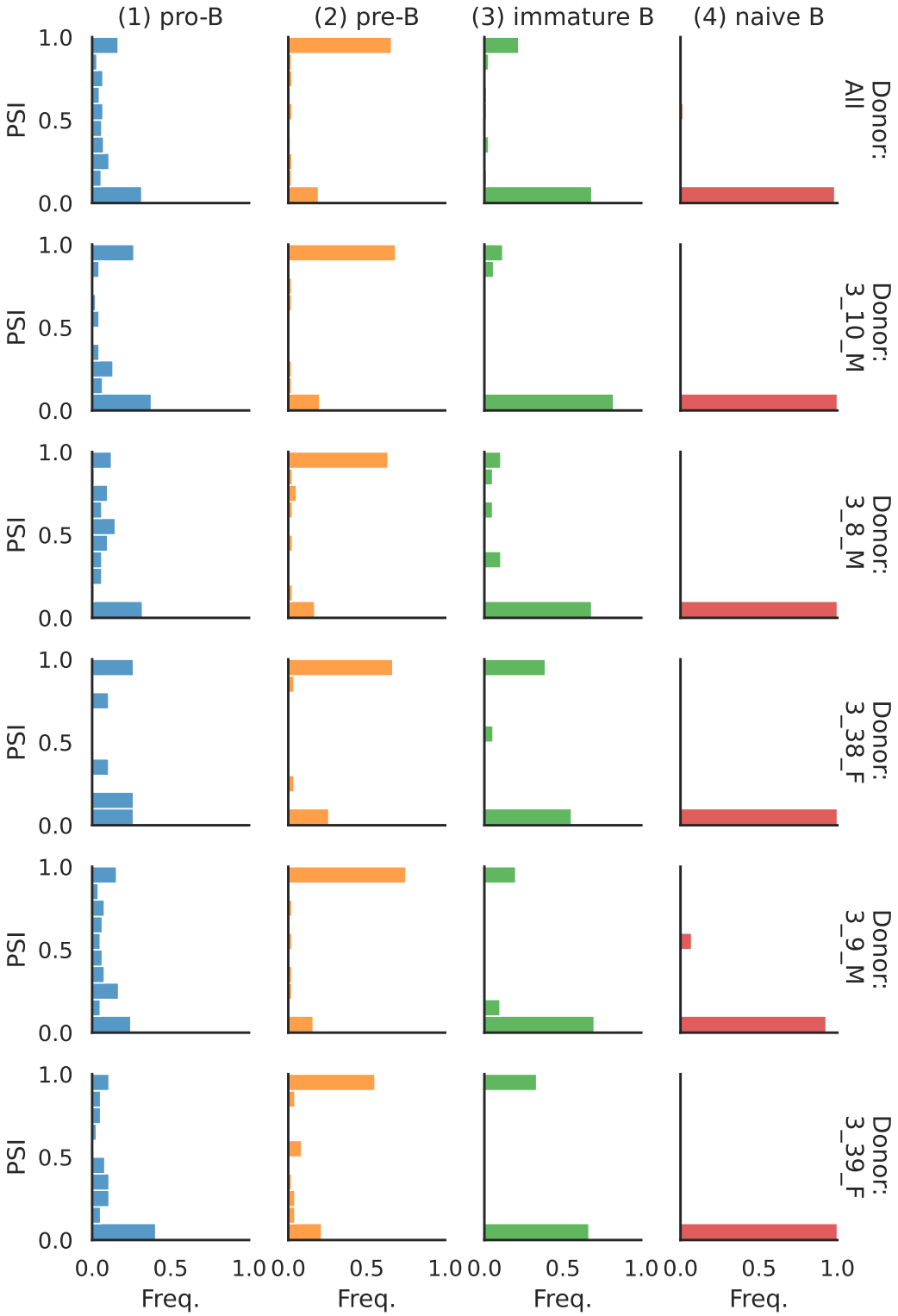


Figure A.9: **PSI distribution of** `Smarca4_28720`**.**

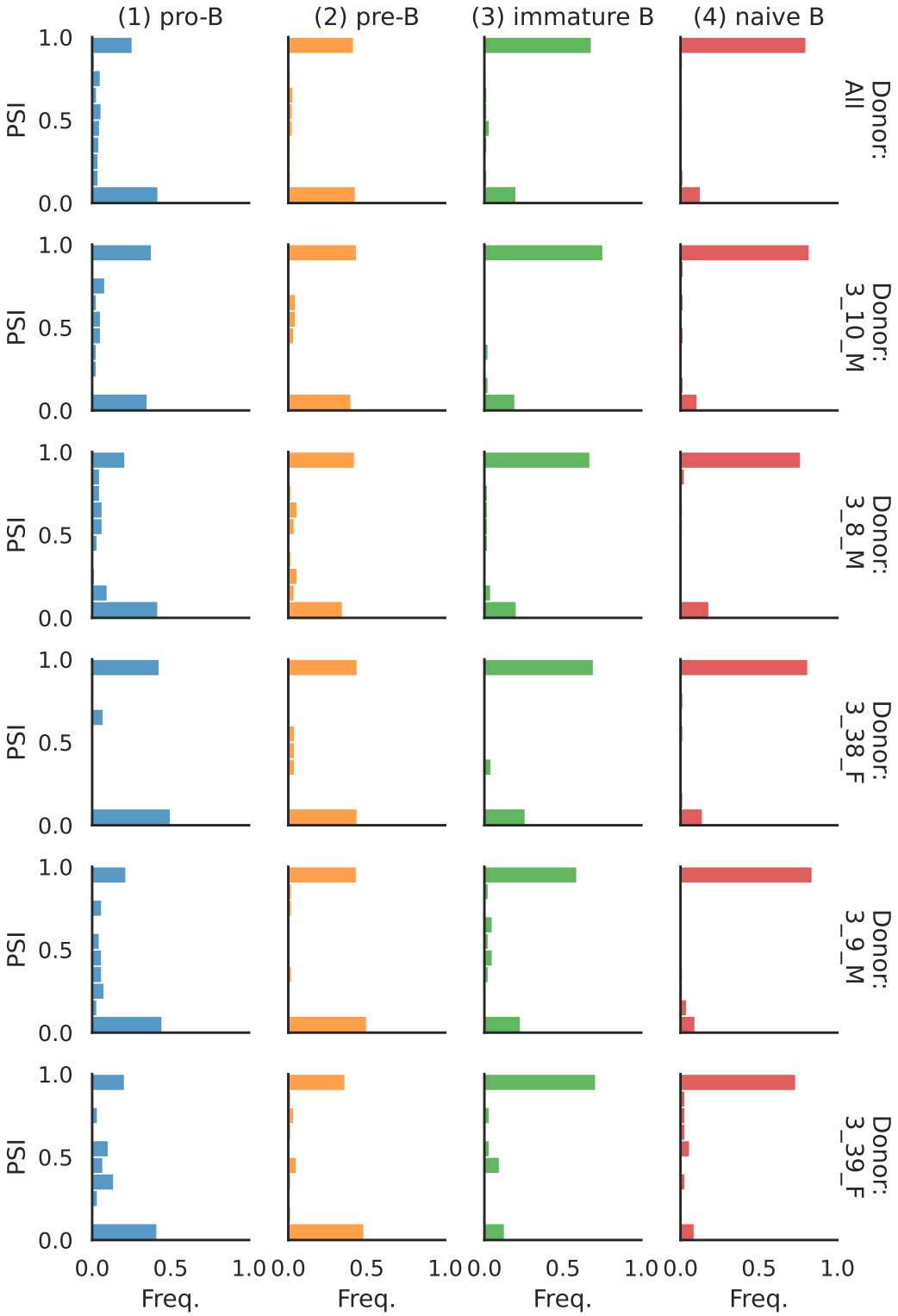


Figure A.10: **PSI distribution of** `Foxp1_11076`**.**

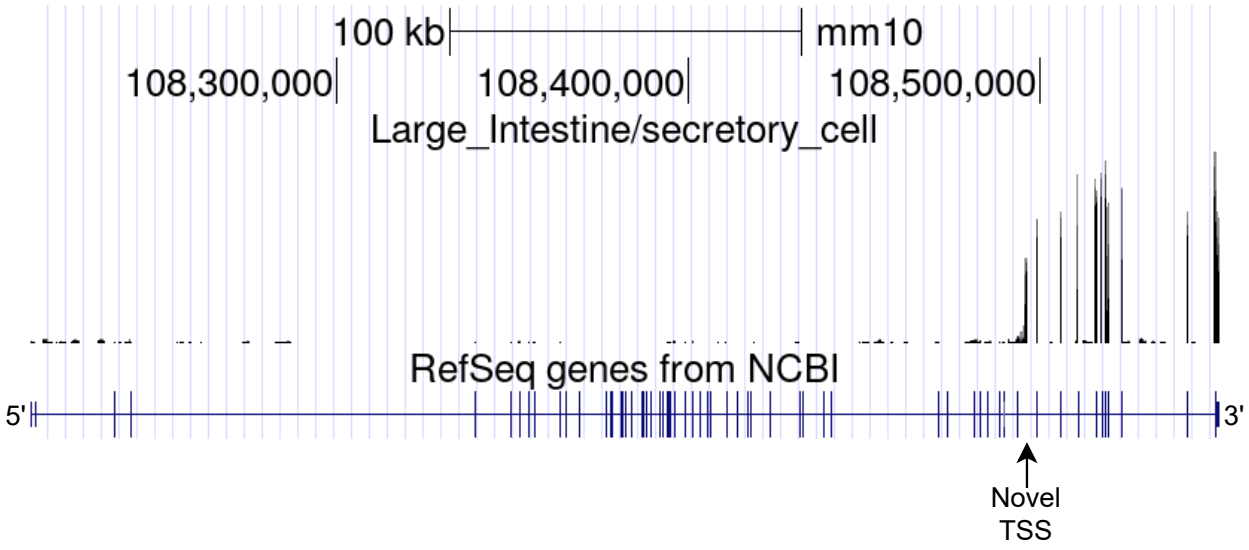


Figure A.11: **Full-gene view of novel alternative TSS in *Itpr1*.** Large intestine secretory cells aggregate read coverage visualized in the UCSC Genome Browser.

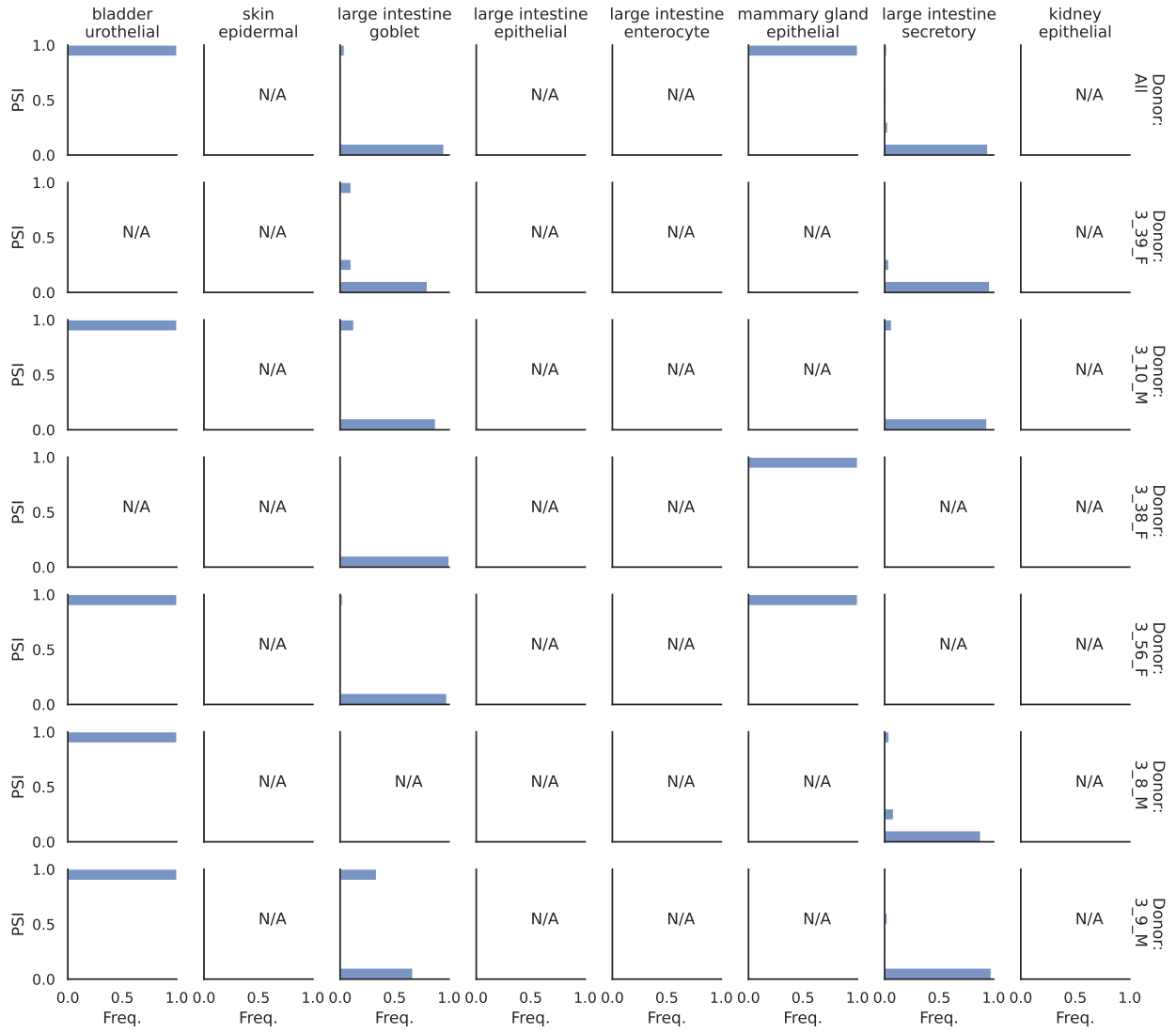


Figure A.12: **PSI distribution of `Itpr1_26257`.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells.

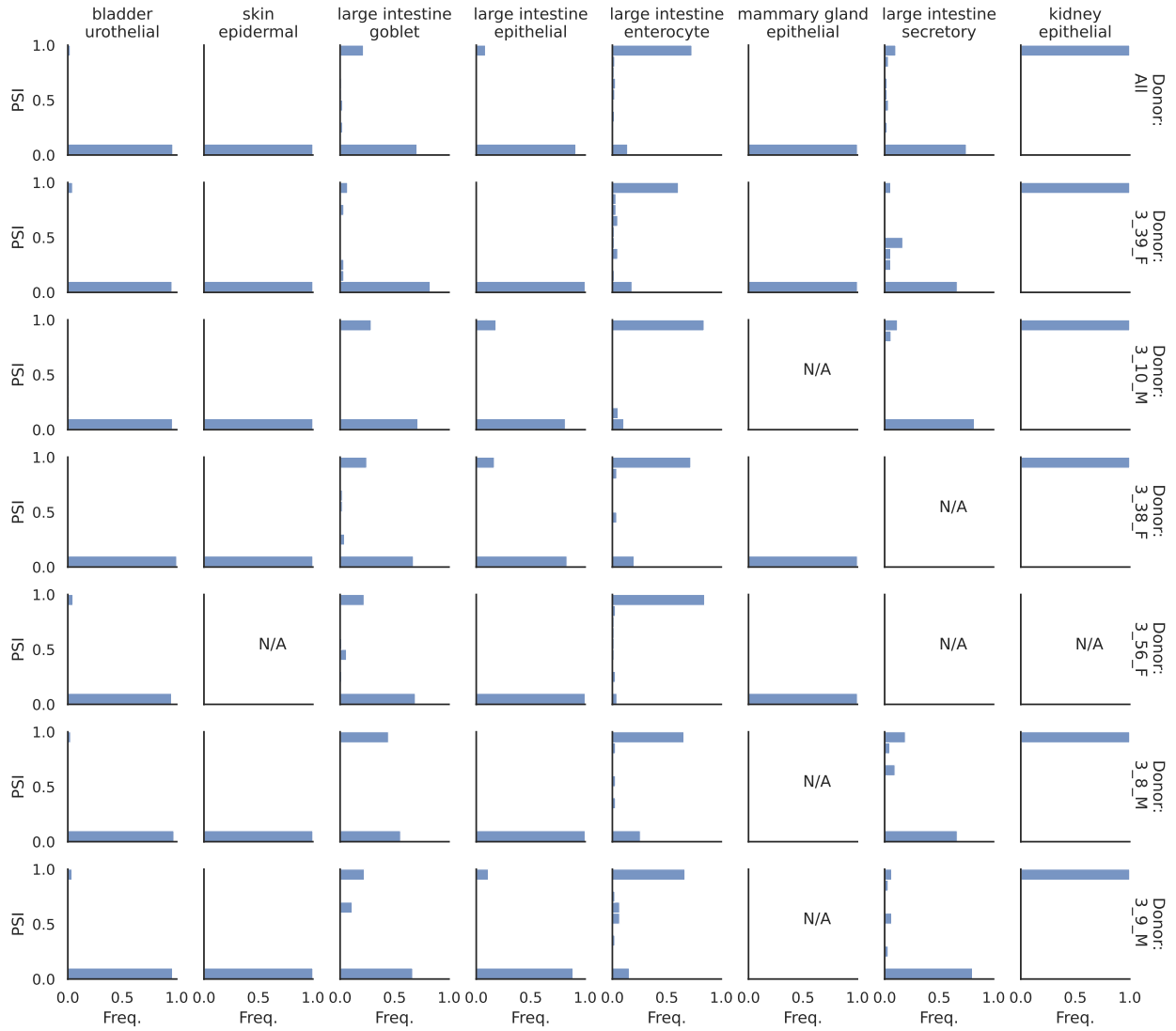


Figure A.13: **PSI distribution of** `Khk_24896`**.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells.

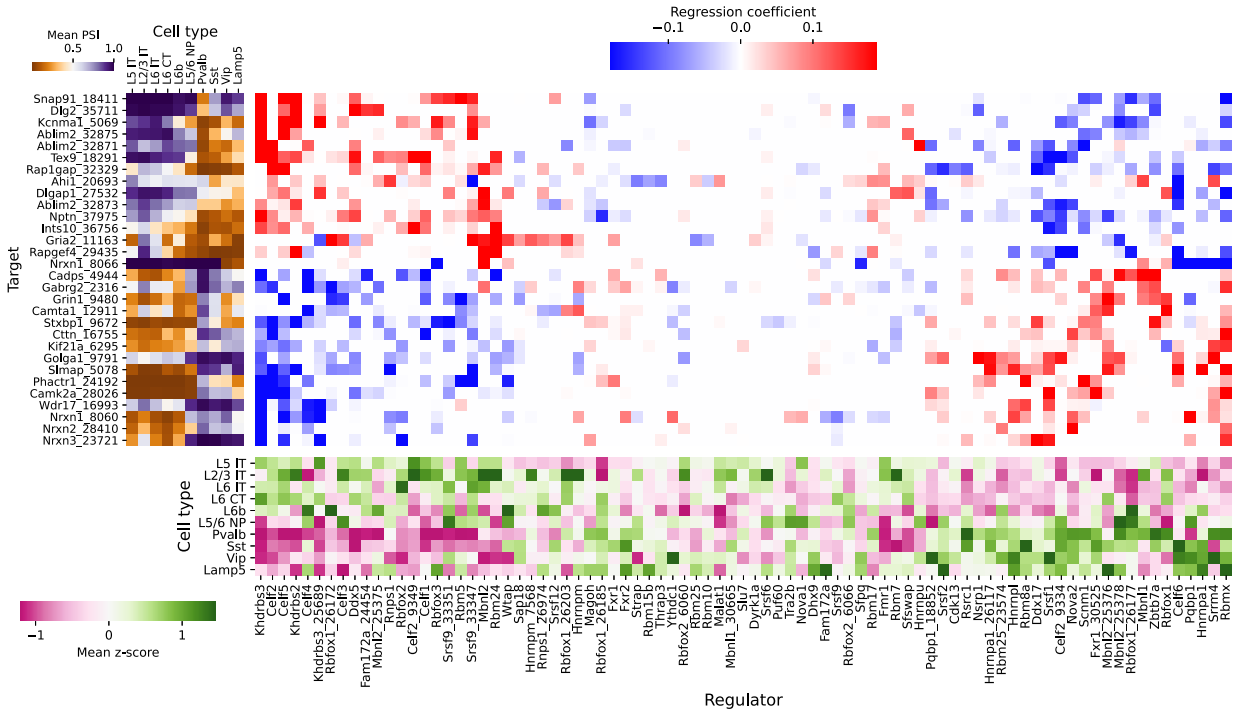


Figure A.14: **Full plot of associations between splicing factors and alternative splicing.** Regression analysis of exon skipping based on expression and splicing of splicing factors, using the BICCN mouse primary motor cortex dataset. Left panel: mean PSI of skipped exons across cell types. Bottom panel: mean z-scores of selected splicing factor features across cell types, including whole-gene expression (gene name) and PSI of alternative introns (gene name and numerical identifier). Center panel: regression coefficients (log-odds) of each splicing factor feature used to predict skipped exon PSI in our sparse Dirichlet-Multinomial linear model.

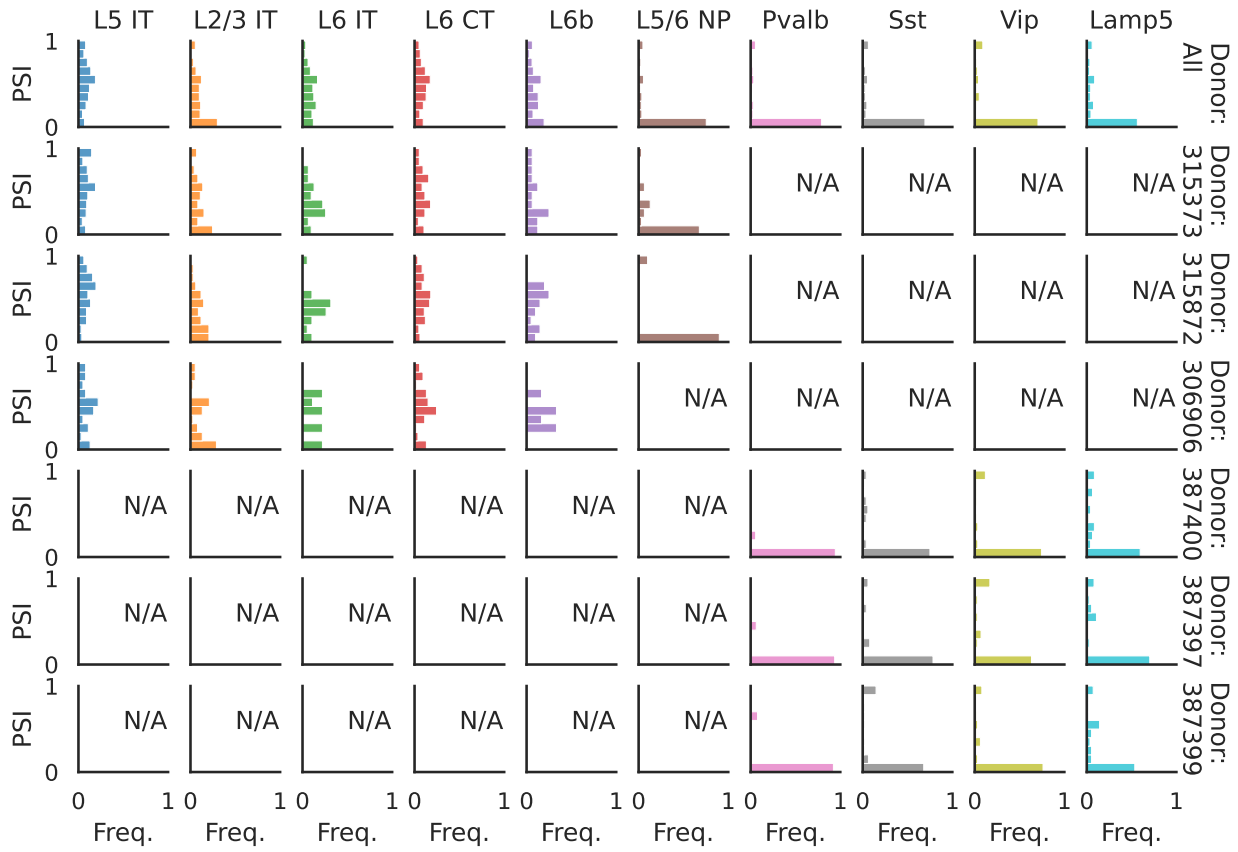


Figure A.15: **PSI distribution of `Khdrbs3_25689`.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells. Per the experimental design of this dataset, the top 3 individuals have only Glutamatergic cell types sequenced, while the bottom 3 have only GABAergic.

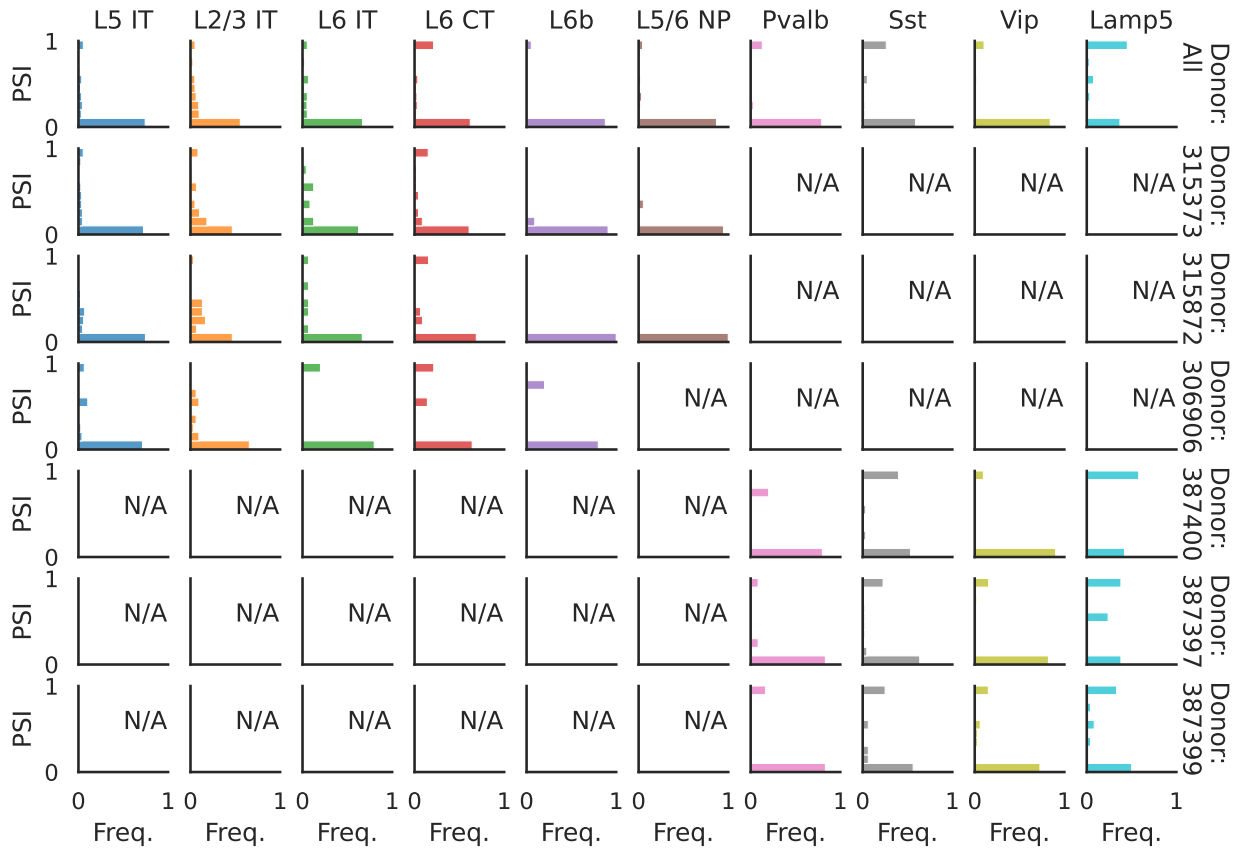


Figure A.16: **PSI distribution of `Mbnl2_25376`.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells. Per the experimental design of this dataset, the top 3 individuals have only Glutamatergic cell types sequenced, while the bottom 3 have only GABAergic.

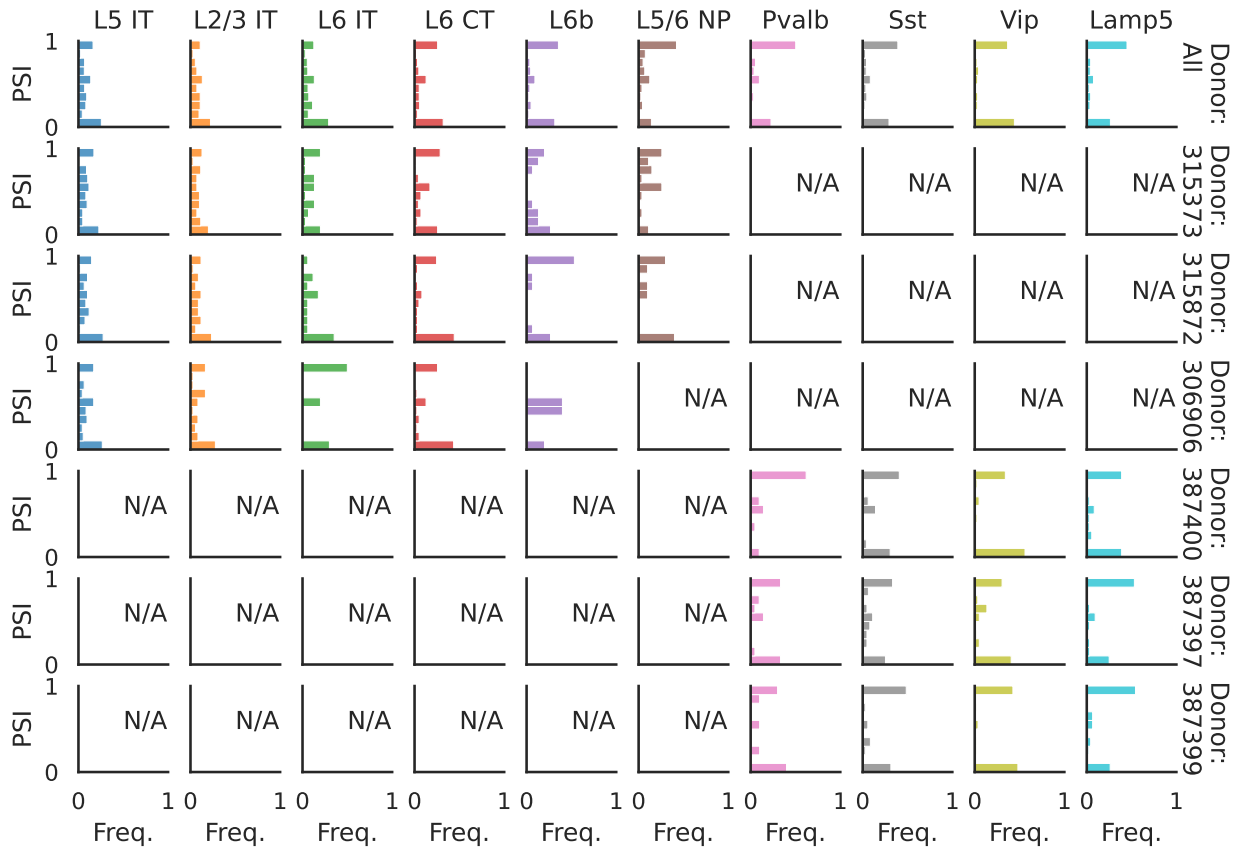


Figure A.17: **PSI distribution of** `Mbnl2_25378`**.** Only six individuals with highest number of cells are displayed. Marked N/A are cell types where the individuals have PSI defined in fewer than 3 cells. Per the experimental design of this dataset, the top 3 individuals have only Glutamatergic cell types sequenced, while the bottom 3 have only GABAergic.

Appendix B

Supplementary Information for Chapter 3

Supplementary Tables

Table B.1: Genome assemblies used for training

Assembly Accession	Assembly Name	Organism Name
GCF_000001735.4	TAIR10.1	*Arabidopsis thaliana*
GCF_000309985.2	CAAS_Brap_v3.01	*Brassica rapa*
GCF_000633955.1	Cs	*Camelina sativa*
GCF_000375325.1	Caprub1_0	*Capsella rubella*
GCF_000150535.2	Papaya1.0	*Carica papaya*
GCF_000478725.1	Eutsalg1_0	*Eutrema salsugineum*
GCF_000801105.1	Rs1.0	*Raphanus sativus*
GCF_000463585.1	ASM46358v1	*Tarenaya hassleriana*

Table B.2: Test perplexity. Perplexity, defined as the exponentiation of the cross-entropy loss, is equivalent to 1 over the probability given to the correct nucleotide. *Arabidopsis thaliana* chromosomes 4 and 5 were used for validation and testing, respectively. Note that reducing the repeat weight leads to improved test perplexity in non-repetitive regions, which are often of greater interest. Compared to full down-weighting, moderate down-weighting results in a similar improvement in perplexity for non-repetitive regions without sacrificing genome-wide perplexity as much.

Model	Chromosome-wide	Non-repeat regions
Repeat weight 1	2.88	2.99
Repeat weight 0.1	2.90	2.92
Repeat weight 0	3.03	2.92

Table B.3: GPN training hyperparameters

Window size (L)	512
Repeat weight	0.1
Embedding dimension (D)	512
Convolutional blocks	25
Convolutional kernel size	9
Convolutional dilation schedule	$1, 2, 4, 8, 16, 32, 1, 2, 4, 8, 16, 32, \ldots$
Optimizer	AdamW
Weight decay	0.01
Batch size	2048
Learning rate	10^{-3} for 120 K steps + decaying (cosine) for 30 K steps
Learning rate warmup	1 K steps

Supplementary Figures

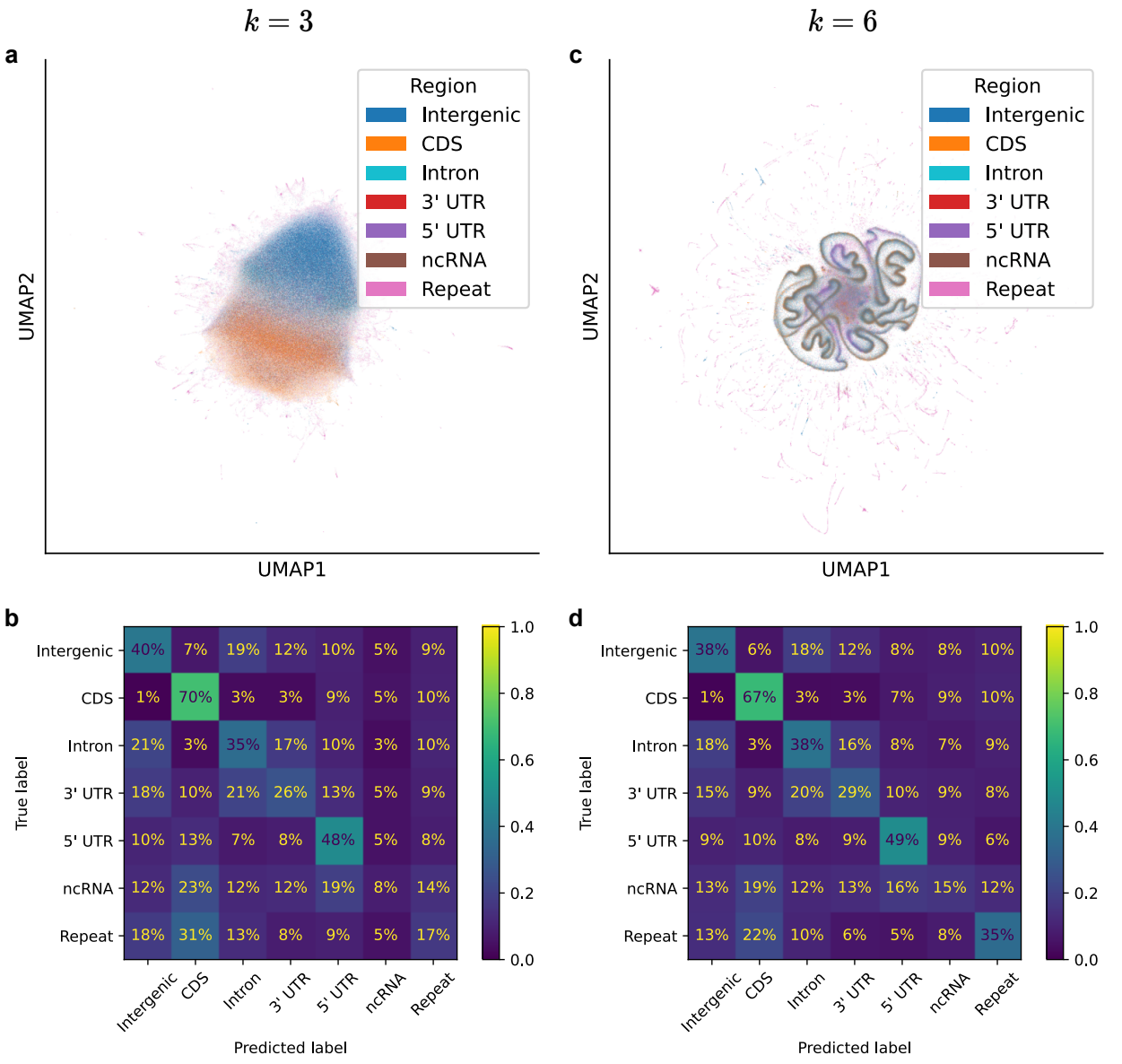


Figure B.1: **UMAP visualization of k-mer spectrum of different windows, as in Figure 3.2, annotated with gene region.** (a,b) $k = 3$. (c,d) $k = 6$.

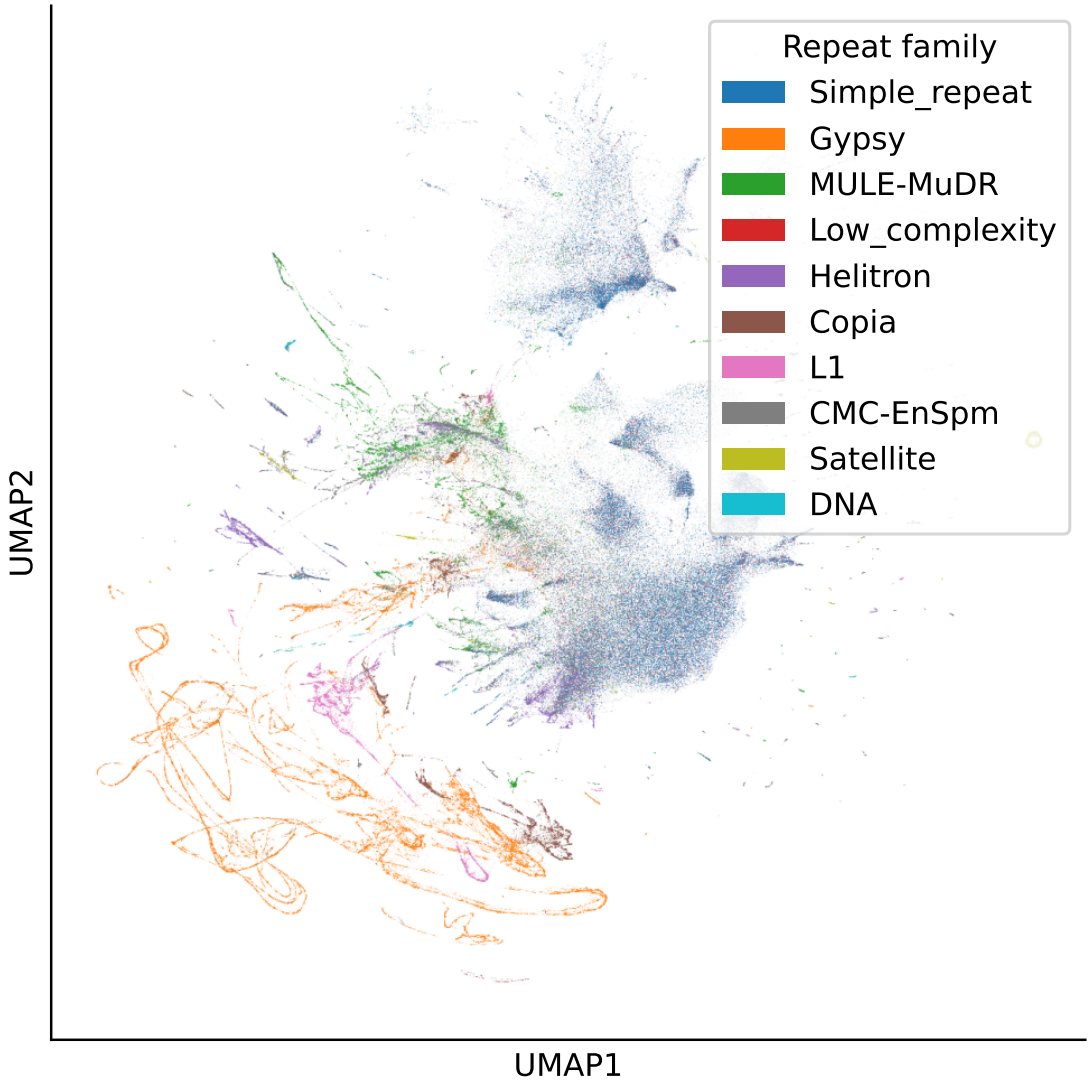


Figure B.2: **UMAP visualization of GPN embeddings, as in Figure 3.2, annotated by repeat family.**

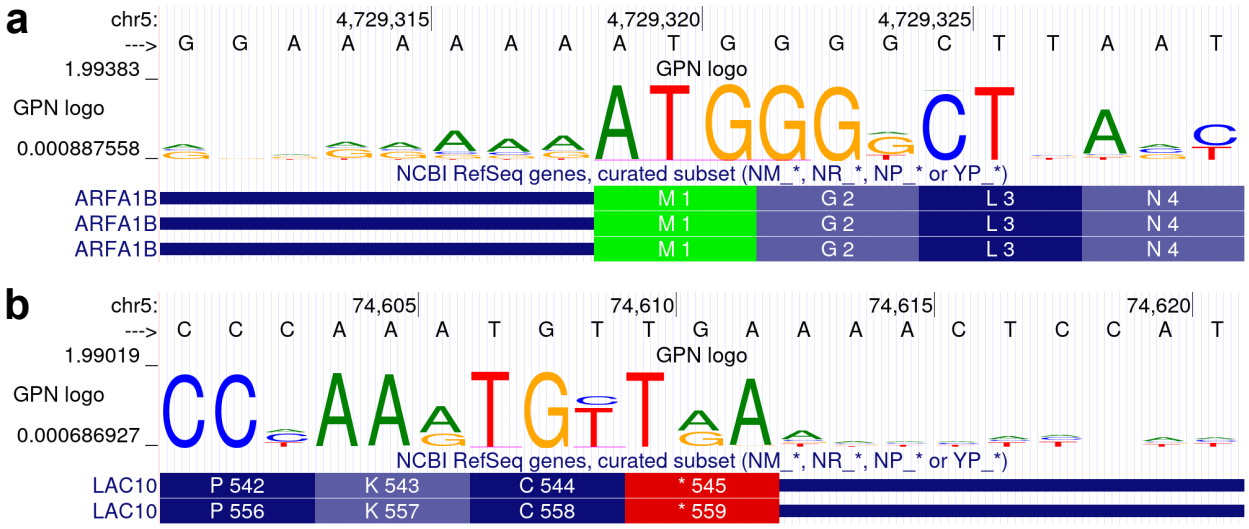


Figure B.3: **Additional GPN sequence logos.** (a) Start codon. (b) Stop codon.

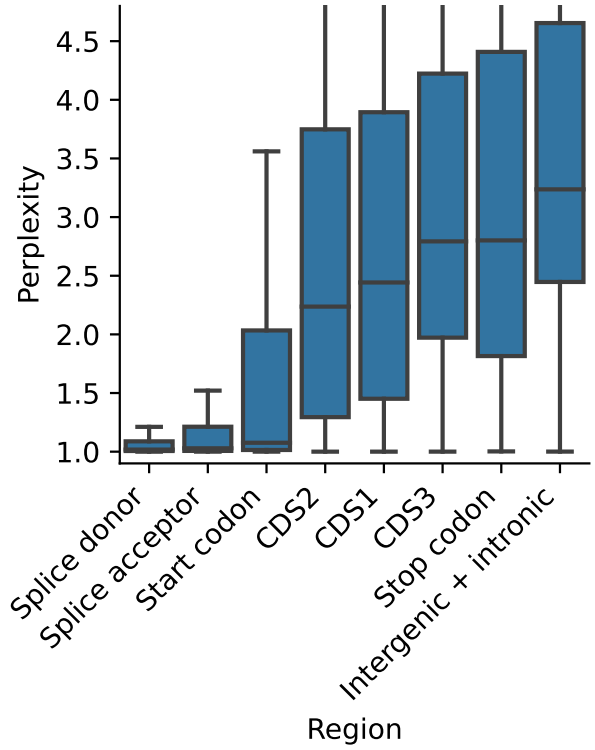


Figure B.4: **Perplexity on select positions from the 1 Mb region** `Chr5:3,500,000-4,500,000` **(test chromosome).** CDS1-3: frame within the coding sequence.

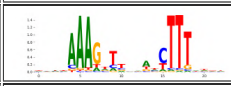
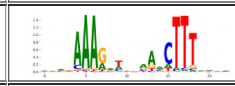
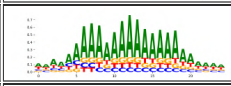
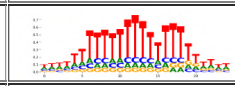
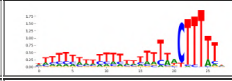
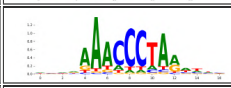
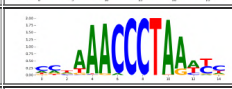
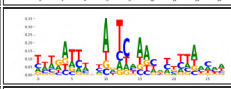
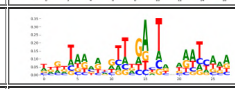
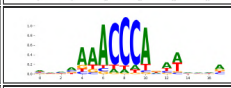
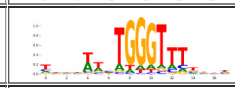
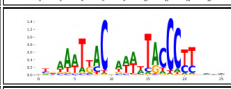
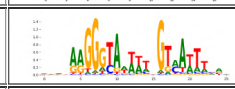
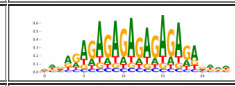
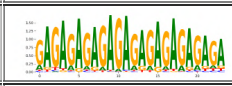
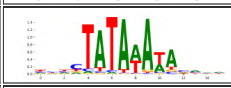
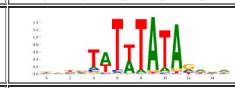
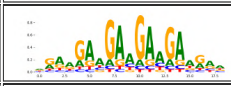
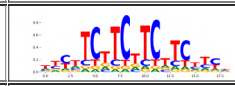
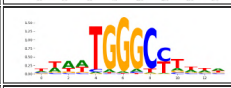
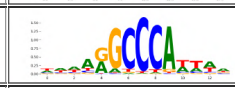
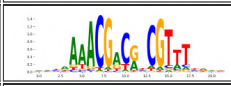
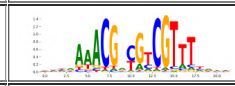
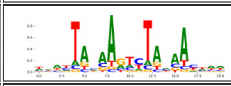
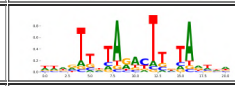
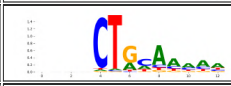
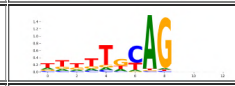
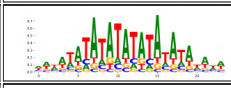
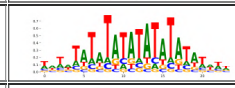
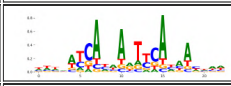
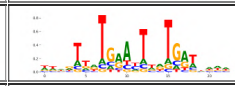
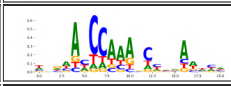
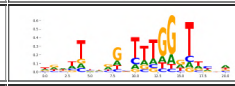
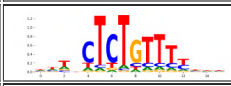
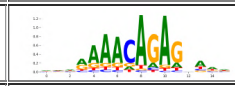
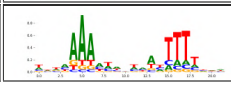
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_0	5028					
pos_patterns.pattern_1	4509			AT4G38000	0.0	
pos_patterns.pattern_2	3386			AT1G72740	0.000117	
pos_patterns.pattern_3	1658					
pos_patterns.pattern_4	1611					
pos_patterns.pattern_5	1556					
pos_patterns.pattern_6	1490			AT2G01930	0.0	
pos_patterns.pattern_7	1424					
pos_patterns.pattern_8	1391			AT5G18090	0.044918	
pos_patterns.pattern_9	1385					
pos_patterns.pattern_10	1057					
pos_patterns.pattern_11	1052					
pos_patterns.pattern_12	928					
pos_patterns.pattern_13	921					
pos_patterns.pattern_14	844					
pos_patterns.pattern_15	837					
pos_patterns.pattern_16	836			AT3G48430	0.000584	
pos_patterns.pattern_17	828					

Figure B.5: **Promoter motifs predicted by GPN and matching motifs in PlantTFDB.**

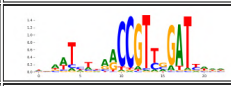
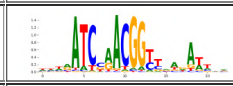
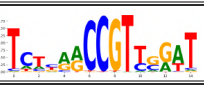
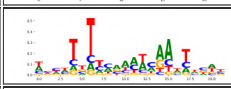
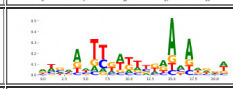
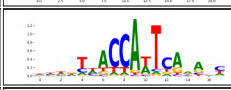
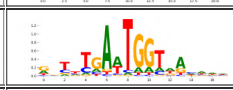
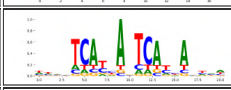
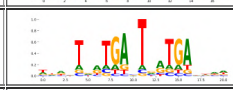
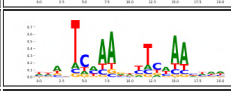
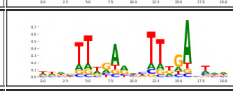
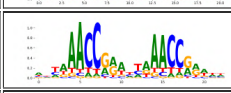
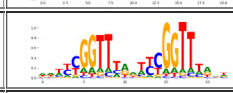
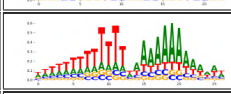
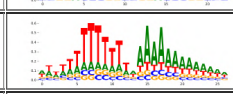
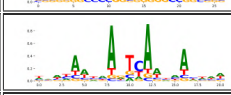
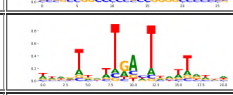
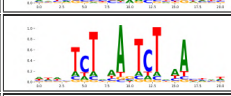
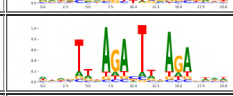
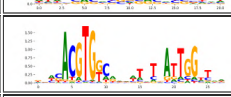
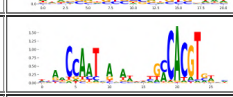
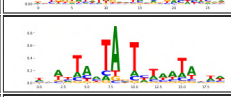
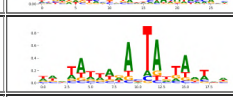
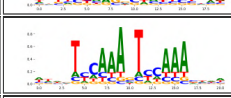
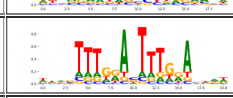
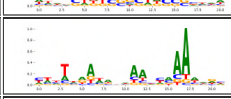
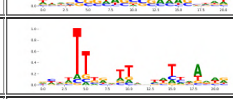
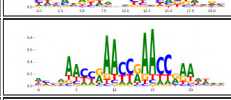
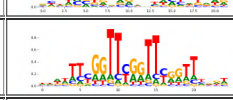
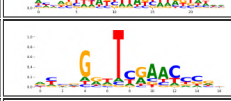
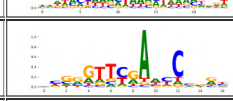
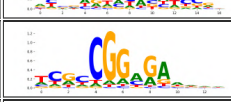
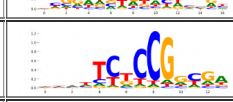
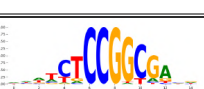
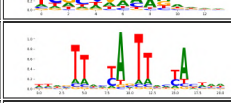
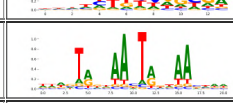
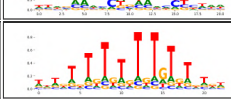
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_18	780			AT4G24470	0.0	
pos_patterns.pattern_19	775			AT2G41835	0.016114	
pos_patterns.pattern_20	756					
pos_patterns.pattern_21	743					
pos_patterns.pattern_22	729			AT1G49560	0.029145	
pos_patterns.pattern_23	708					
pos_patterns.pattern_24	702					
pos_patterns.pattern_25	696					
pos_patterns.pattern_26	688					
pos_patterns.pattern_27	674					
pos_patterns.pattern_28	643			AT2G36610	0.026016	
pos_patterns.pattern_29	640					
pos_patterns.pattern_30	631					
pos_patterns.pattern_31	631					
pos_patterns.pattern_32	630					
pos_patterns.pattern_33	604			AT3G58630	0.029229	
pos_patterns.pattern_34	595					
pos_patterns.pattern_35	595					

Figure B.5: (Continued)

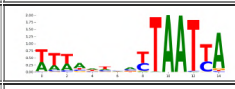
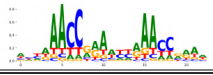
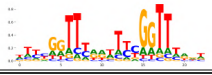
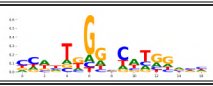
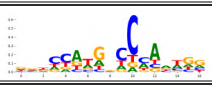
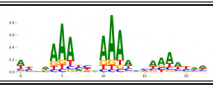
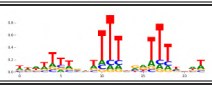
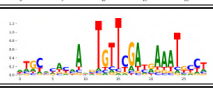
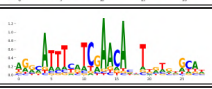
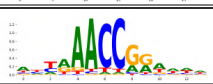
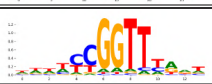
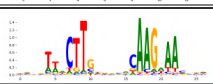
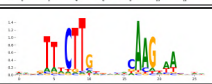
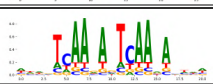
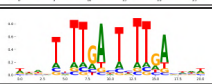
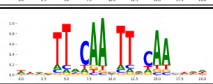
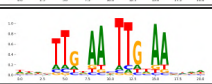
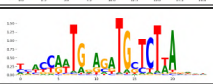
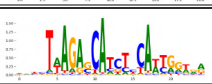
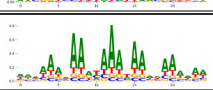
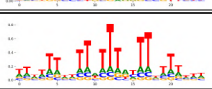
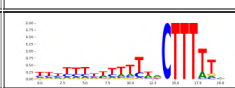
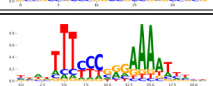
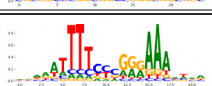
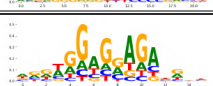
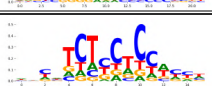
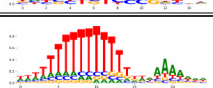
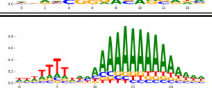
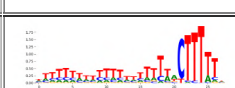
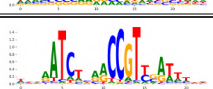
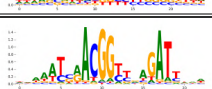
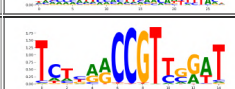
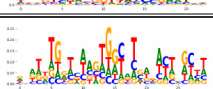
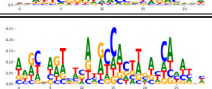
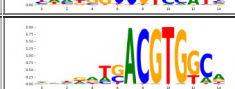
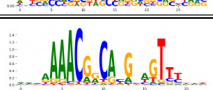
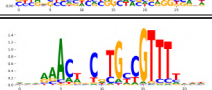
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_36	590			AT3G28920	0.000035	
pos_patterns.pattern_37	580					
pos_patterns.pattern_38	566					
pos_patterns.pattern_39	558					
pos_patterns.pattern_40	544					
pos_patterns.pattern_41	523					
pos_patterns.pattern_42	512			AT3G10480	0.000001	
pos_patterns.pattern_43	511					
pos_patterns.pattern_44	508					
pos_patterns.pattern_45	506					
pos_patterns.pattern_46	492			AT2G28810	0.000187	
pos_patterns.pattern_47	480					
pos_patterns.pattern_48	480					
pos_patterns.pattern_49	477			AT4G38000	0.002972	
pos_patterns.pattern_50	454			AT4G24470	0.000033	
pos_patterns.pattern_51	451			AT3G10800	0.025896	
pos_patterns.pattern_52	433					
pos_patterns.pattern_53	426					

Figure B.5: (Continued)

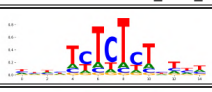
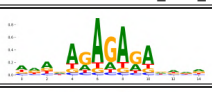
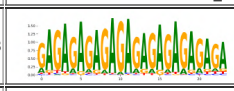
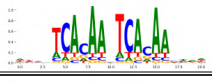
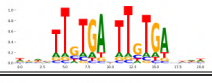
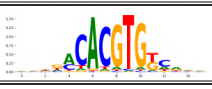
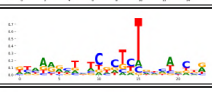
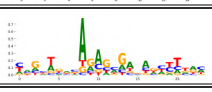
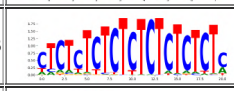
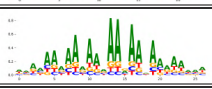
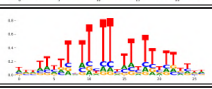
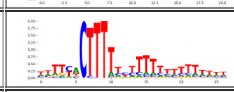
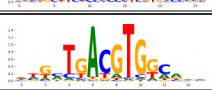
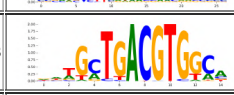
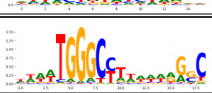
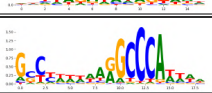
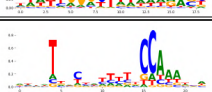
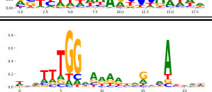
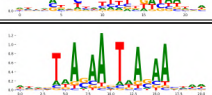
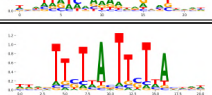
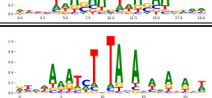
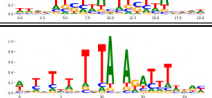
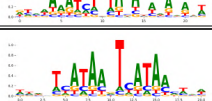
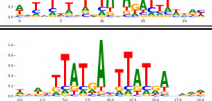
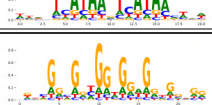
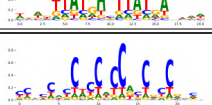
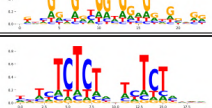
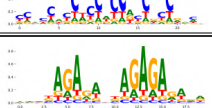
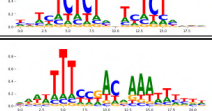
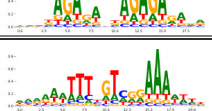
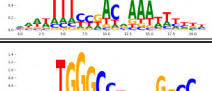
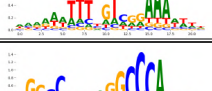
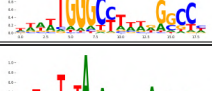
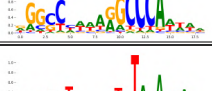
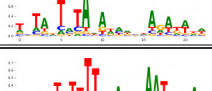
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_54	407			AT2G01930	0.002768	
pos_patterns.pattern_55	404					
pos_patterns.pattern_56	387			AT4G34000	0.000455	
pos_patterns.pattern_57	357			AT5G42520	0.037446	
pos_patterns.pattern_58	354			AT1G69570	0.0	
pos_patterns.pattern_59	352			AT3G62420	0.000053	
pos_patterns.pattern_60	322					
pos_patterns.pattern_61	319					
pos_patterns.pattern_62	314					
pos_patterns.pattern_63	311					
pos_patterns.pattern_64	309					
pos_patterns.pattern_65	302			AT1G21910	0.0	
pos_patterns.pattern_66	293					
pos_patterns.pattern_67	291			AT2G33860	0.005094	
pos_patterns.pattern_68	285					
pos_patterns.pattern_69	282					
pos_patterns.pattern_70	280					
pos_patterns.pattern_71	280			AT1G53170	0.001163	

Figure B.5: (Continued)

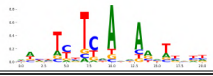
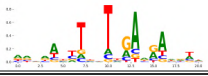
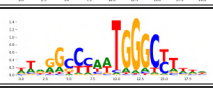
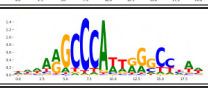
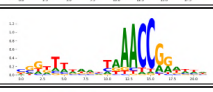
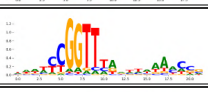
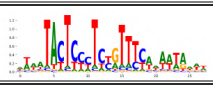
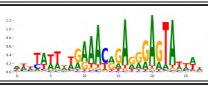
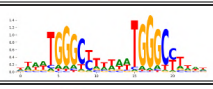
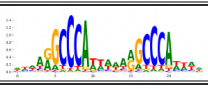
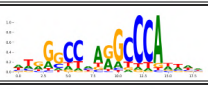
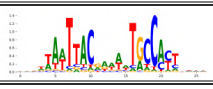
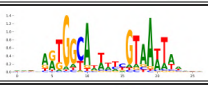
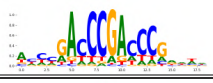
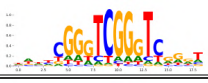
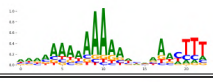
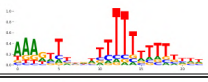
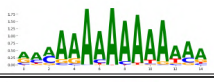
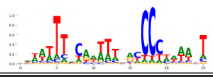
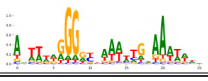
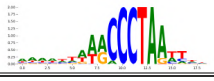
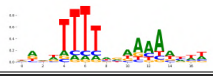
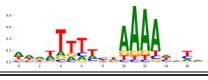
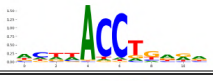
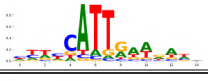
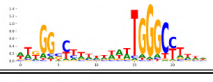
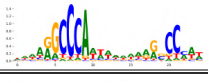
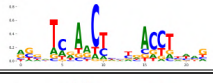
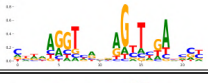
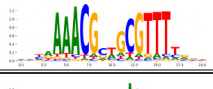
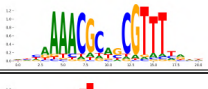
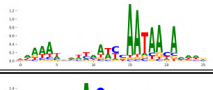
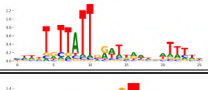
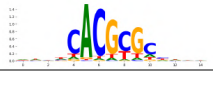
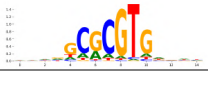
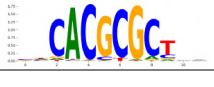
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_72	275					
pos_patterns.pattern_73	265					
pos_patterns.pattern_74	263					
pos_patterns.pattern_75	262					
pos_patterns.pattern_76	259					
pos_patterns.pattern_77	253					
pos_patterns.pattern_78	249					
pos_patterns.pattern_79	248					
pos_patterns.pattern_80	242			AT1G49480	0.000331	
pos_patterns.pattern_81	228			AT5G67580	0.016528	
pos_patterns.pattern_82	225					
pos_patterns.pattern_83	221					
pos_patterns.pattern_84	215					
pos_patterns.pattern_85	209					
pos_patterns.pattern_86	206					
pos_patterns.pattern_87	199					
pos_patterns.pattern_88	198					
pos_patterns.pattern_89	194			AT3G22170	0.002435	

Figure B.5: (Continued)

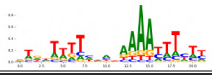
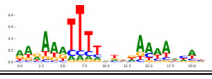
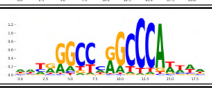
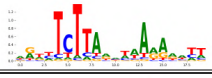
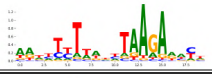
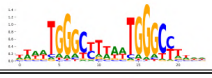
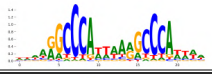
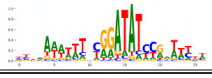
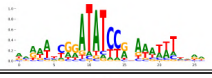
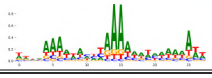
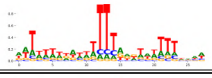
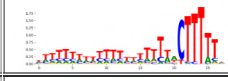
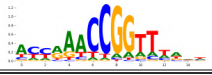
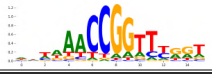
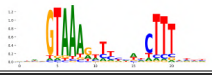
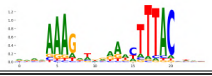
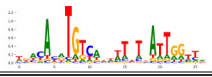
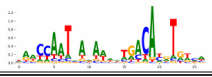
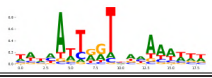
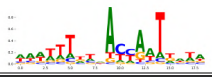
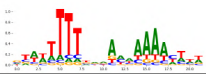
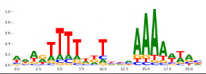
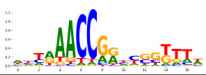
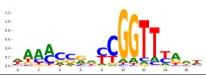
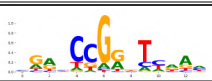
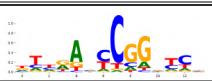
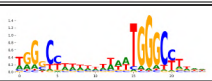
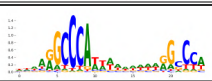
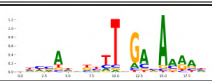
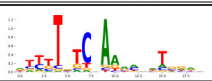
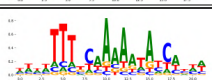
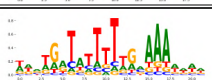
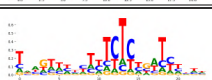
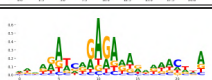
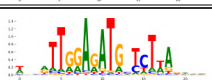
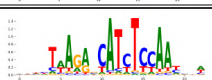
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_90	186					
pos_patterns.pattern_91	169					
pos_patterns.pattern_92	167					
pos_patterns.pattern_93	167					
pos_patterns.pattern_94	167					
pos_patterns.pattern_95	159			AT4G38000	0.005238	
pos_patterns.pattern_96	158					
pos_patterns.pattern_97	157					
pos_patterns.pattern_98	157					
pos_patterns.pattern_99	156					
pos_patterns.pattern_100	154					
pos_patterns.pattern_101	151					
pos_patterns.pattern_102	146					
pos_patterns.pattern_103	136					
pos_patterns.pattern_104	135					
pos_patterns.pattern_105	135			AT5G02840	0.041659	
pos_patterns.pattern_106	131					
pos_patterns.pattern_107	131					

Figure B.5: (Continued)

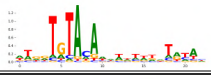
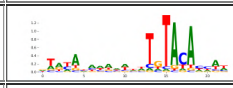
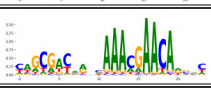
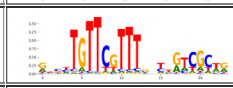
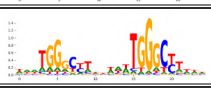
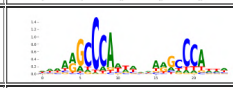
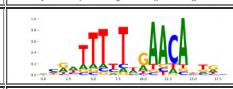
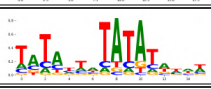
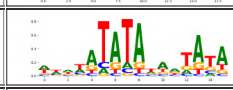
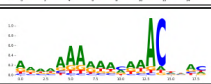
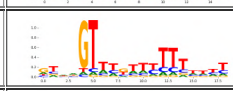
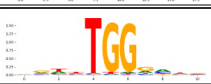
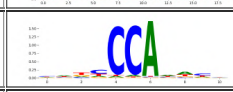
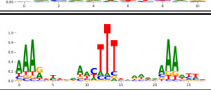
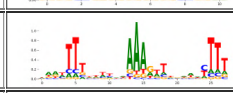
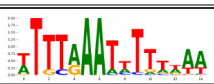
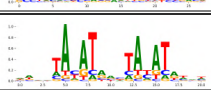
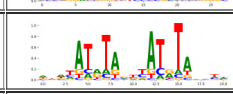
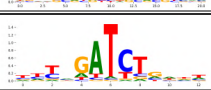
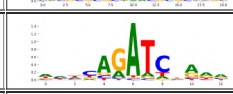
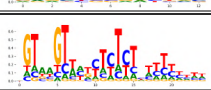
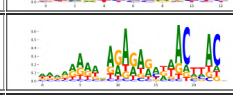
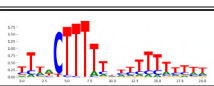
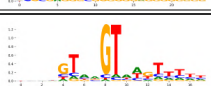
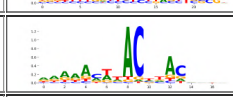
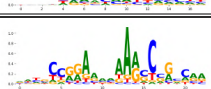
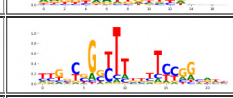
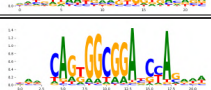
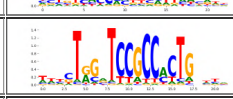
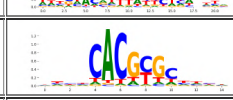
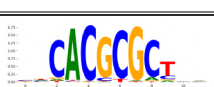
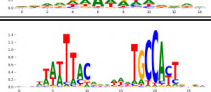
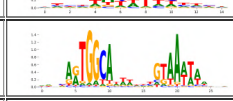
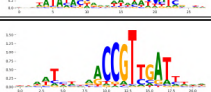
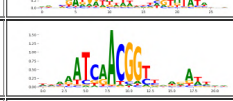
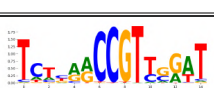
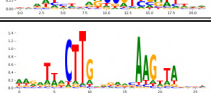
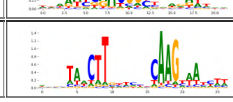
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_108	130					
pos_patterns.pattern_109	128					
pos_patterns.pattern_110	126					
pos_patterns.pattern_111	125					
pos_patterns.pattern_112	121					
pos_patterns.pattern_113	117					
pos_patterns.pattern_114	115					
pos_patterns.pattern_115	107			AT2G20110	0.049803	
pos_patterns.pattern_116	106					
pos_patterns.pattern_117	105					
pos_patterns.pattern_118	105			AT3G55370	0.025597	
pos_patterns.pattern_119	104					
pos_patterns.pattern_120	103					
pos_patterns.pattern_121	102					
pos_patterns.pattern_122	100			AT3G22170	0.007859	
pos_patterns.pattern_123	96					
pos_patterns.pattern_124	93			AT4G24470	0.000435	
pos_patterns.pattern_125	91			AT3G10500	0.000013	

Figure B.5: (Continued)

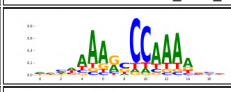
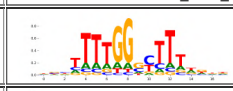
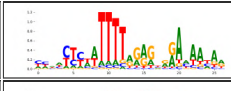
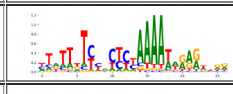
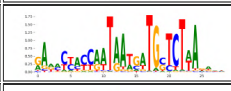
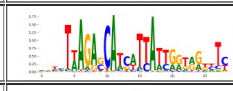
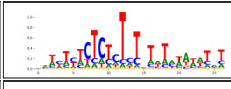
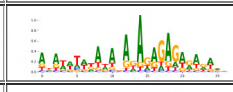
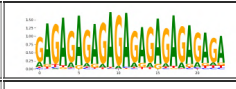
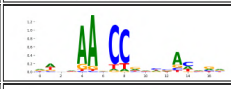
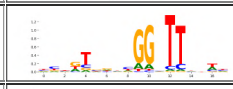
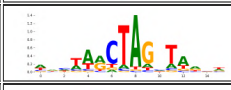
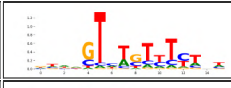
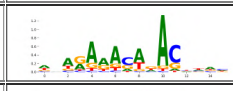
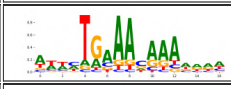
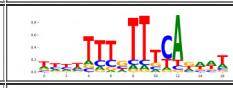
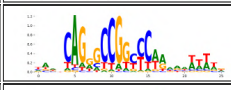
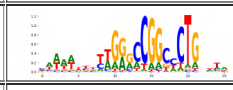
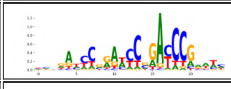
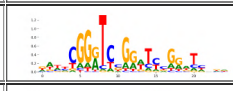
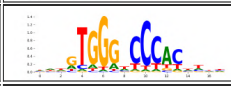
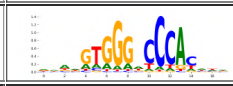
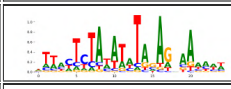
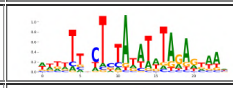
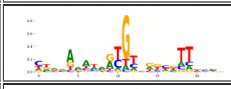
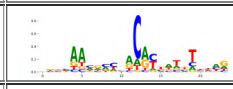
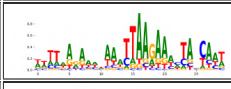
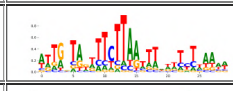
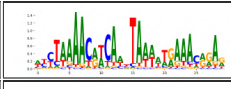
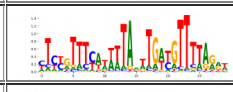
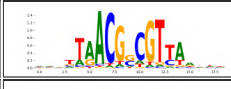
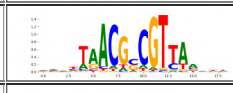
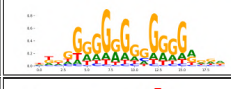
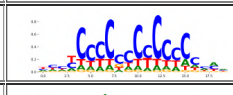
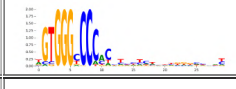
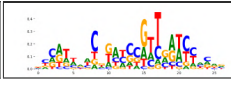
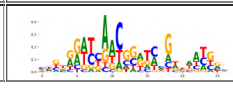
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_126	88					
pos_patterns.pattern_127	88					
pos_patterns.pattern_128	87					
pos_patterns.pattern_129	86			AT2G01930	0.007347	
pos_patterns.pattern_130	80					
pos_patterns.pattern_131	79					
pos_patterns.pattern_132	75					
pos_patterns.pattern_133	72					
pos_patterns.pattern_134	67					
pos_patterns.pattern_135	63					
pos_patterns.pattern_136	61			AT5G23280	0.000262	
pos_patterns.pattern_137	61					
pos_patterns.pattern_138	61					
pos_patterns.pattern_139	56					
pos_patterns.pattern_140	55					
pos_patterns.pattern_141	54			AT3G10030	0.000039	
pos_patterns.pattern_142	52			AT1G67260	0.031614	
pos_patterns.pattern_143	52					

Figure B.5: (Continued)

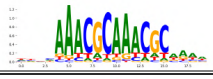
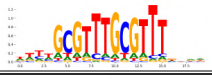
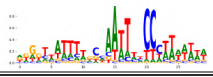
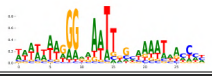
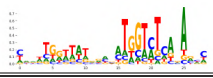
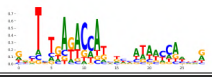
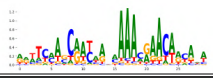
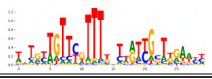
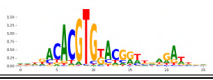
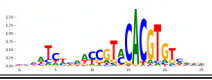
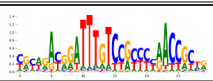
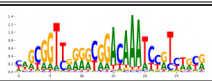
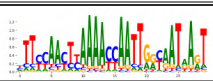
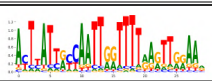
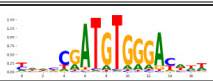
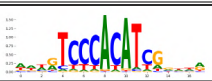
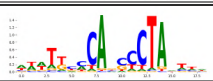
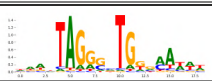
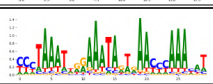
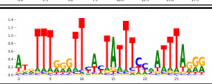
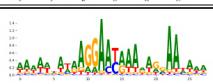
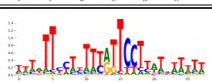
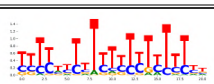
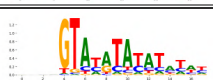
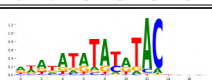
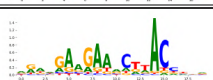
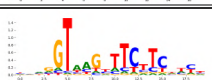
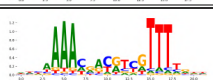
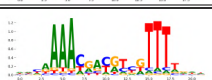
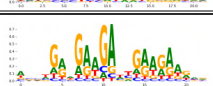
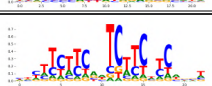
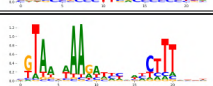
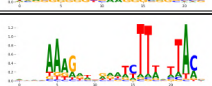
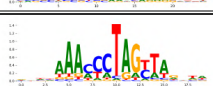
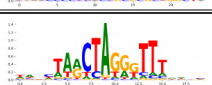
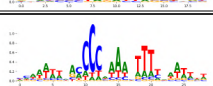
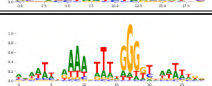
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_144	49					
pos_patterns.pattern_145	49					
pos_patterns.pattern_146	45					
pos_patterns.pattern_147	41					
pos_patterns.pattern_148	40			AT4G34000	0.002944	
pos_patterns.pattern_149	39					
pos_patterns.pattern_150	35					
pos_patterns.pattern_151	35					
pos_patterns.pattern_152	32					
pos_patterns.pattern_153	29					
pos_patterns.pattern_154	28			AT2G45660	0.02262	
pos_patterns.pattern_155	28					
pos_patterns.pattern_156	28					
pos_patterns.pattern_157	27					
pos_patterns.pattern_158	27					
pos_patterns.pattern_159	26					
pos_patterns.pattern_160	26					
pos_patterns.pattern_161	23					

Figure B.5: (Continued)

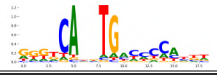
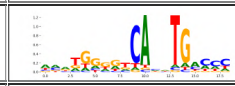
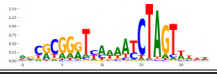
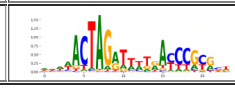
pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev	match0	qval0	match0_logo
pos_patterns.pattern_162	23					
pos_patterns.pattern_163	22					

Figure B.5: (Continued)

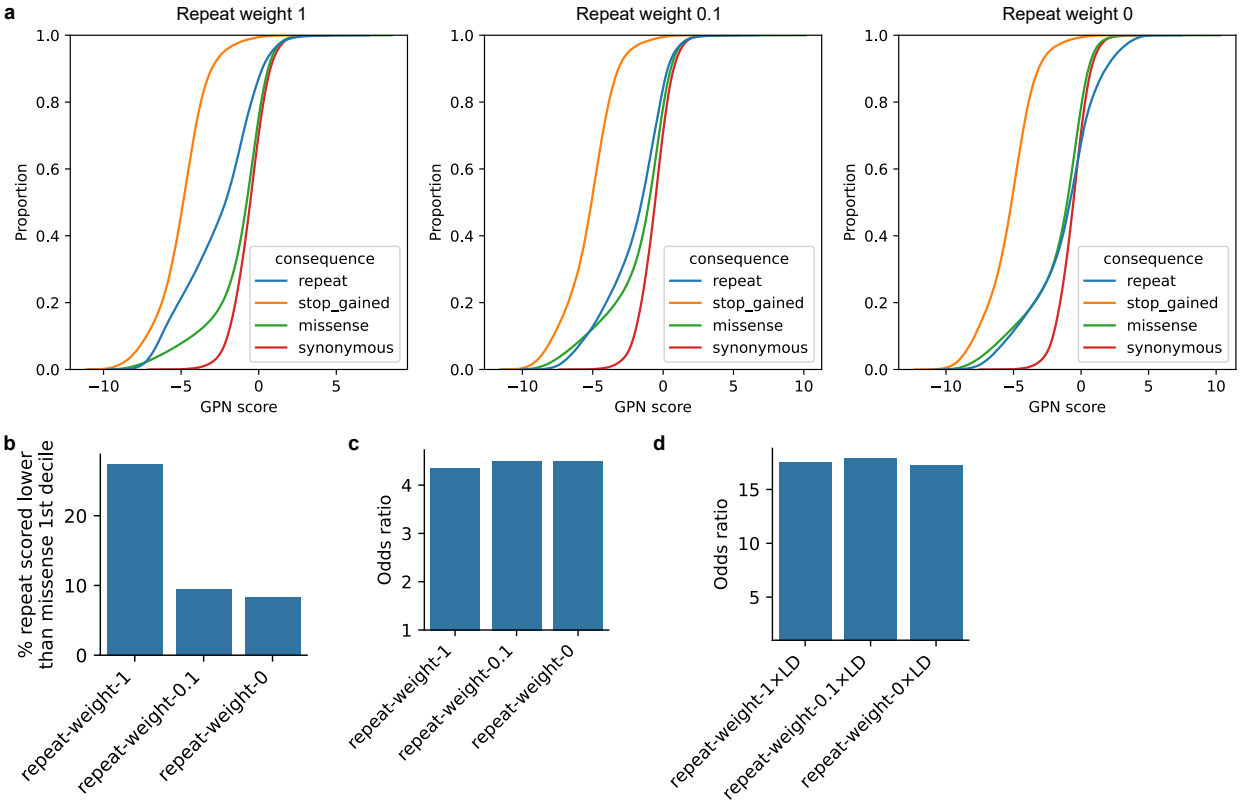


Figure B.6: **Comparison of GPN models trained with different loss weights on repeats.** (a) Cumulative distribution function of GPN scores for simulated variants in specific categories, as described in Figure 3.4. (b) Percentage of simulated repeat variants scored lower than the first decile of simulated missense variants. (c) Odds ratios for rare (AC = 1) vs. common (AF $\geq$ 5%) variants, as described in Figure 3.5c. AC: allele count. AF: allele frequency. (d) Odds ratios for GWAS hits, as described in Figure 3.6c.

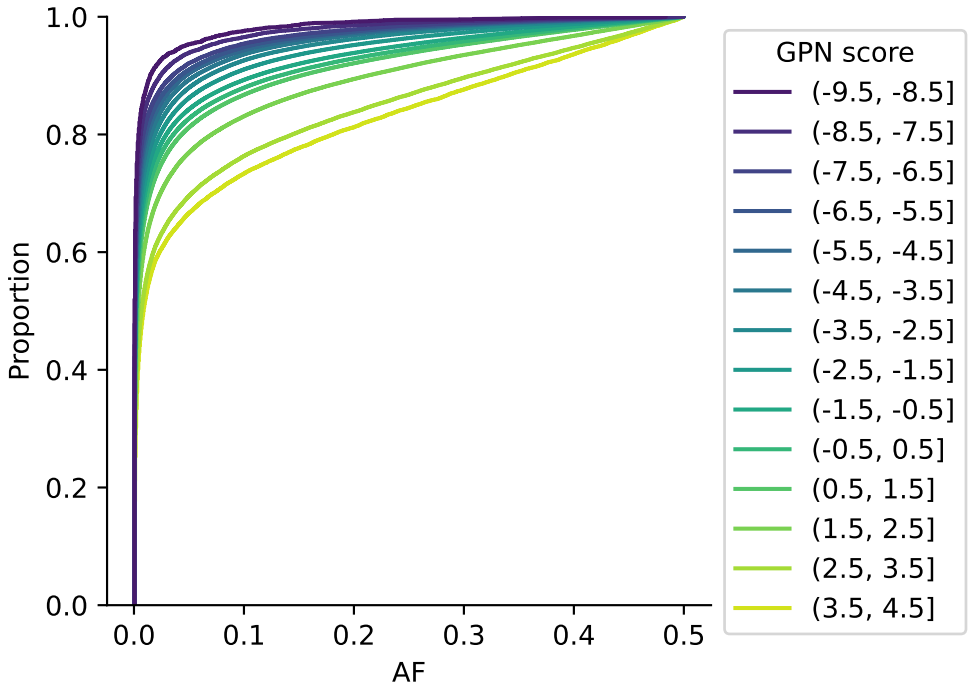


Figure B.7: **Cumulative distribution function of allele frequency (AF) for variants in different GPN score bins, as described in Figure 3.5b.**

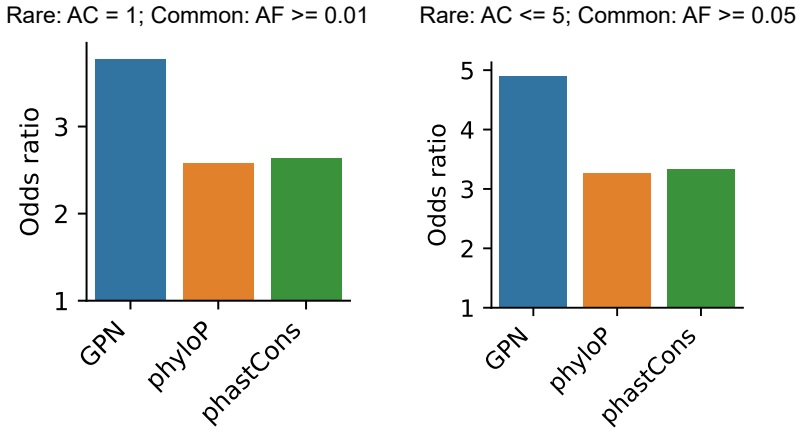


Figure B.8: **Rare vs. common odds ratios for different thresholds for defining rare and common variants.** Odds ratios (OR) were calculated as described in Figure 3.5c.

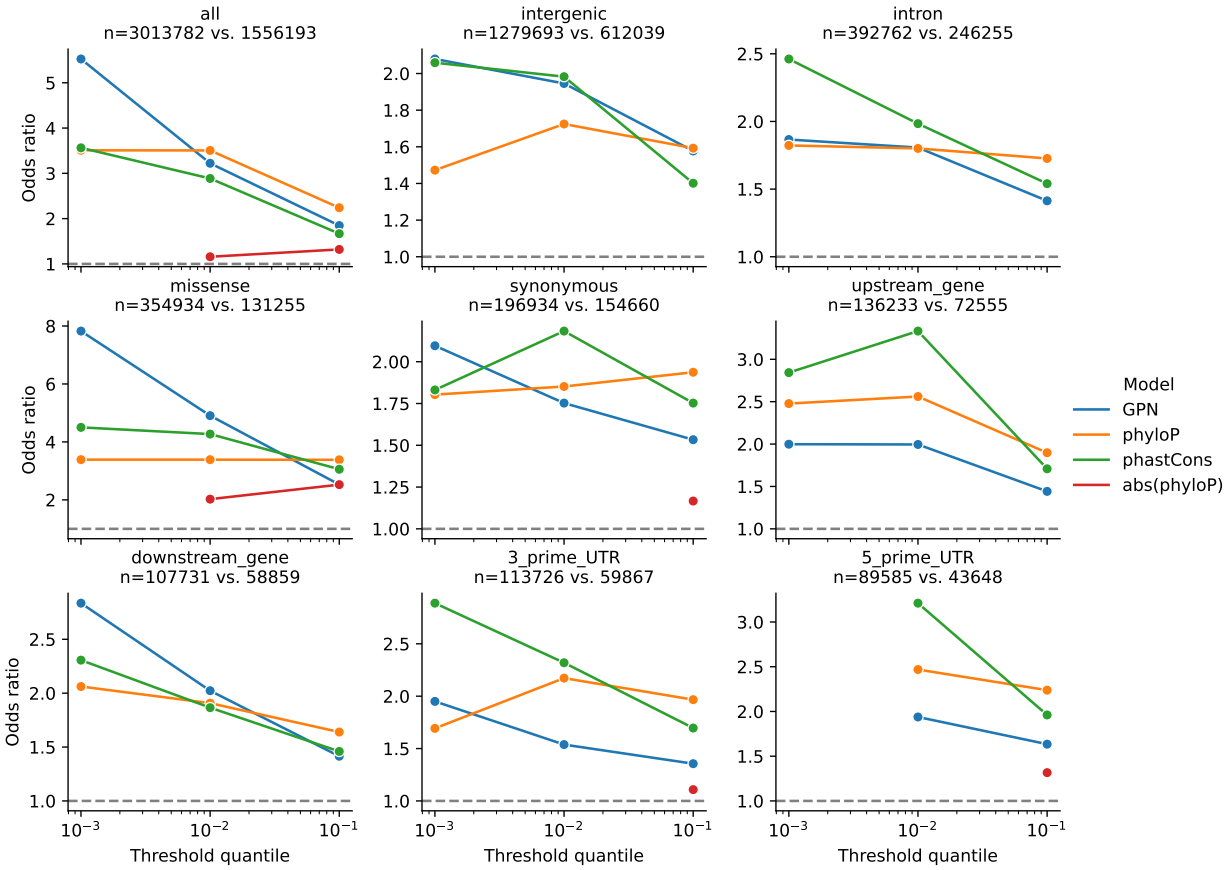


Figure B.9: **Rare vs. common odds ratios for specific variant categories and different thresholds for defining functional scores.** Odds ratios (OR) were calculated as described in Figure 3.5c. Only significant odds ratios are shown. The most stringent threshold in 5' UTR was excluded due to certain models having less than 10 counts in an entry of the contingency table.

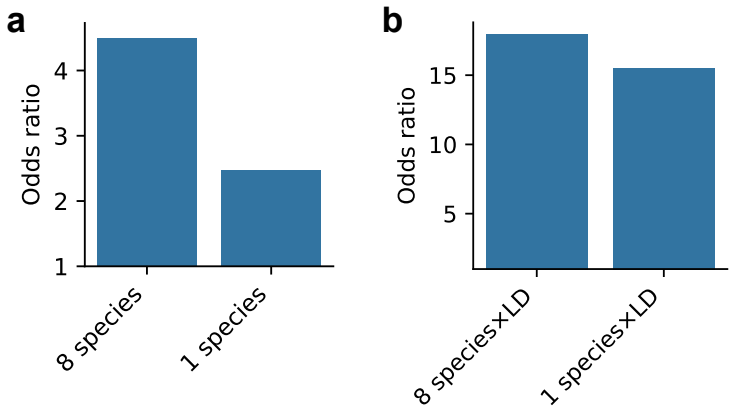


Figure B.10: **Comparison of GPN models trained on a different number of species.** (a) Odds ratios for rare (AC = 1) vs. common (AF $\geq$ 5%), as described in Figure 3.5c. AC: allele count. AF: allele frequency. (b) Odds ratios for GWAS hits, as described in Figure 3.6c.

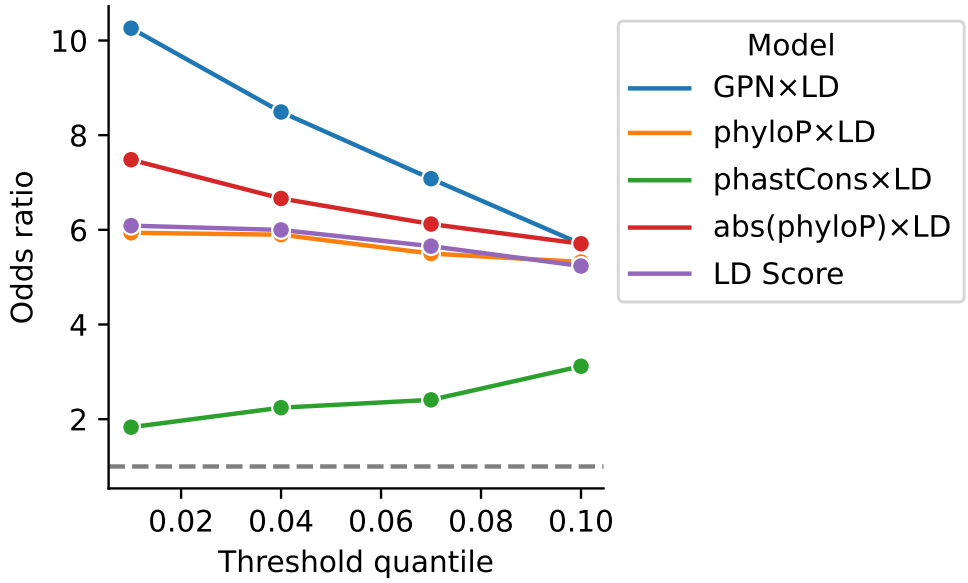


Figure B.11: **GWAS hit odds ratios for different thresholds for defining functional-tagged scores.** Odds ratios (OR) were calculated as described in Figure 3.6c.

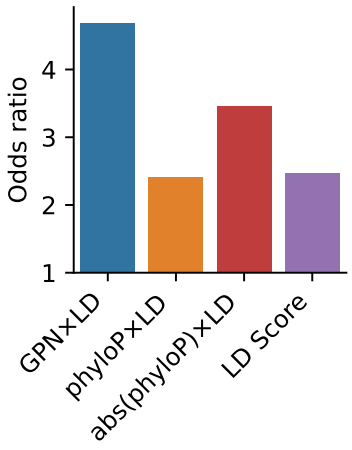


Figure B.12: **Odds ratios for GWAS hits, using the Bonferroni correction instead of permutation-based significance threshold, as described in Figure 3.6c.** Only significant odds ratios are shown.

Appendix C

Supplementary Information for Chapter 4

Supplementary Tables

Table C.1: GPN-MSA training hyperparameters

Weight decay	0.01
Max steps	30 K
Batch size	2048
Learning rate	10^{-4}
Learning rate schedule	Cosine
Learning rate warmup steps	1 K

Table C.2: Functional annotations considered in our analysis of functional enrichment. In particular, annotations relying on predictive models are not considered.

Functional Annotation	**Category**	**Definition / Interpretation**
Recombination Rate	Local Nucleotide Diversity	How likely a region tends to undergo recombination
Nuclear Diversity	Local Nucleotide Diversity	How likely the region diversifies
B Statistic	Local Nucleotide Diversity	Population-genetic quantity. Lower value means greater impact of selection on removing diversity
Percent GC	Epigenetics	Percent GC in +/− 75bp window
Percent CpG	Epigenetics	Percent CpG in +/− 75bp window
RNA-seq	Epigenetics	Max level over 10 cell lines (ENCODE)
DNase-seq	Epigenetics	Max level over 12 cell lines (ENCODE)
H3K4me1	Epigenetics	Max level over 13 cell lines (ENCODE)
H3K4me2	Epigenetics	Max level over 14 cell lines (ENCODE)
H3K4me3	Epigenetics	Max level over 14 cell lines (ENCODE)
H3K9ac	Epigenetics	Max level over 13 cell lines (ENCODE)
H3K9me3	Epigenetics	Max level over 14 cell lines (ENCODE)
H3K27ac	Epigenetics	Max level over 14 cell lines (ENCODE)
H3K27me3	Epigenetics	Max level over 14 cell lines (ENCODE)
H3K36me3	Epigenetics	Max level over 10 cell lines (ENCODE)
H3K79me2	Epigenetics	Max level over 13 cell lines (ENCODE)
H4K20me1	Epigenetics	Max level over 11 cell lines (ENCODE)
H2AFZ	Epigenetics	Max level over 13 cell lines (ENCODE)

Supplementary Figures

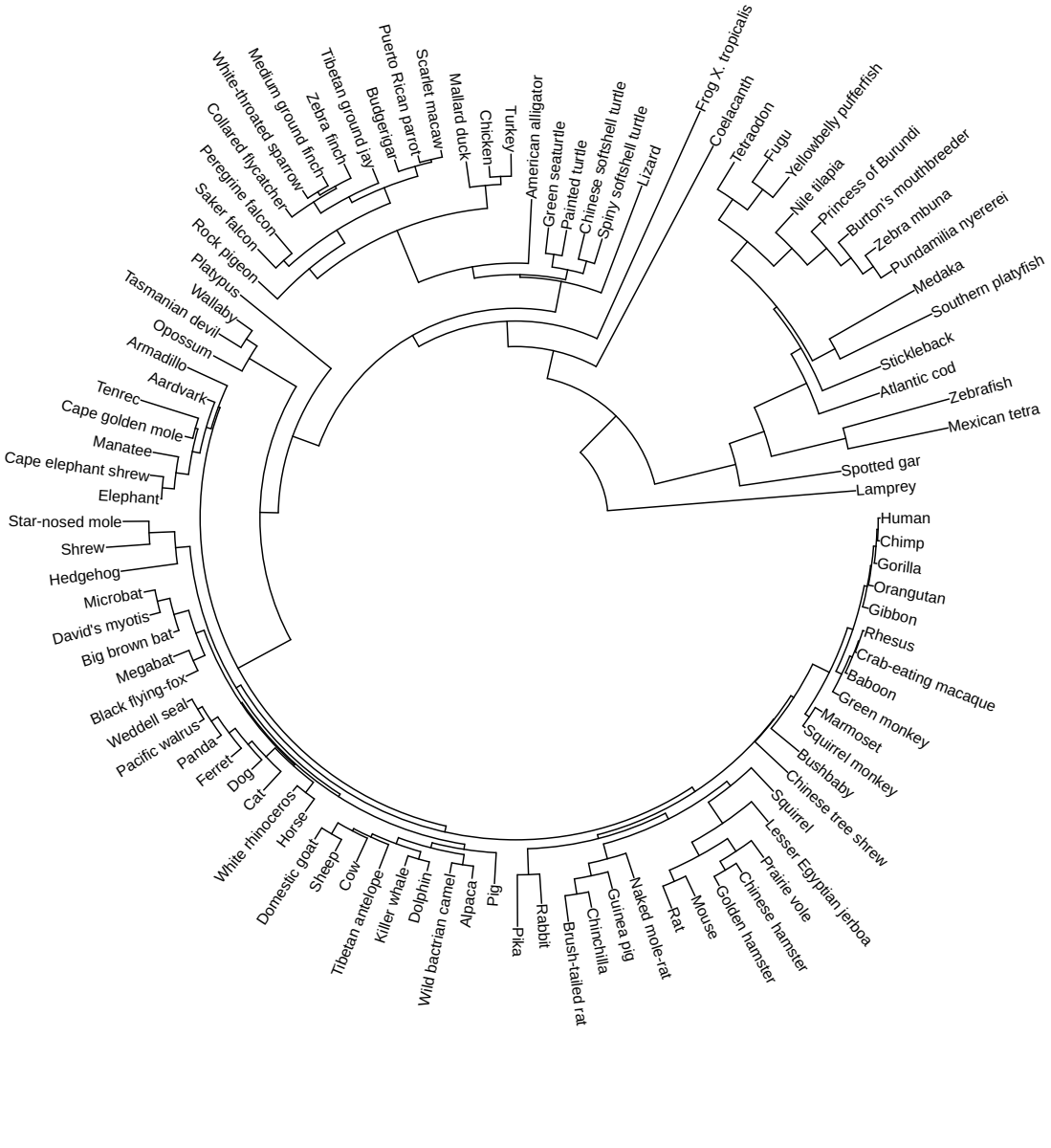


Figure C.1: **Phylogenetic tree of 100 vertebrates.**

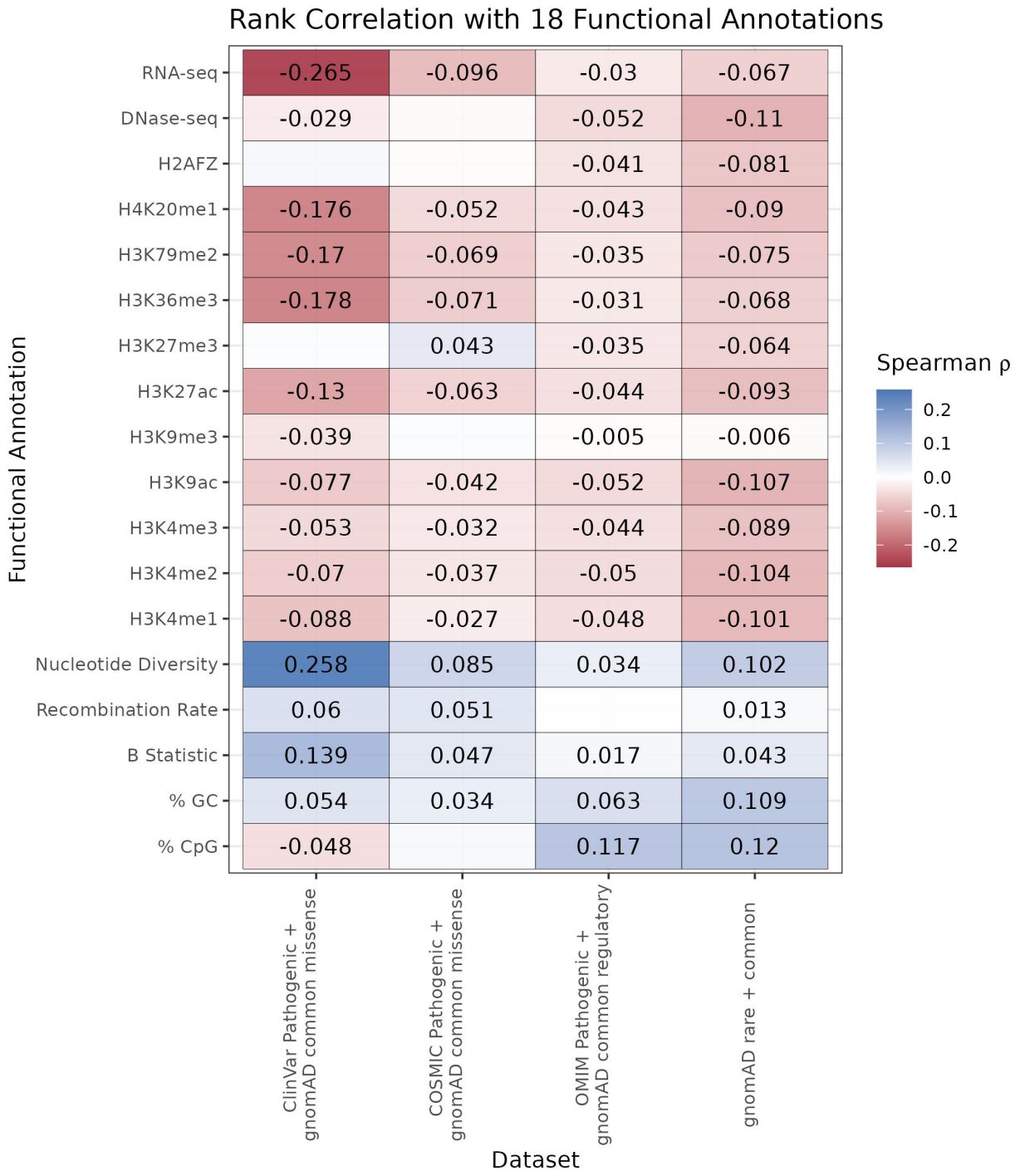


Figure C.2: **Functional impact of GPN-MSA.** Rank correlation between GPN-MSA and 18 assay-based or biologically interpretable functional annotations, across four datasets. Only significant (with FWER controlled at 0.05) correlations are shown. For tracks like RNA-seq, which require the variant to be exonic, variants without the annotation are not included in correlation computations.

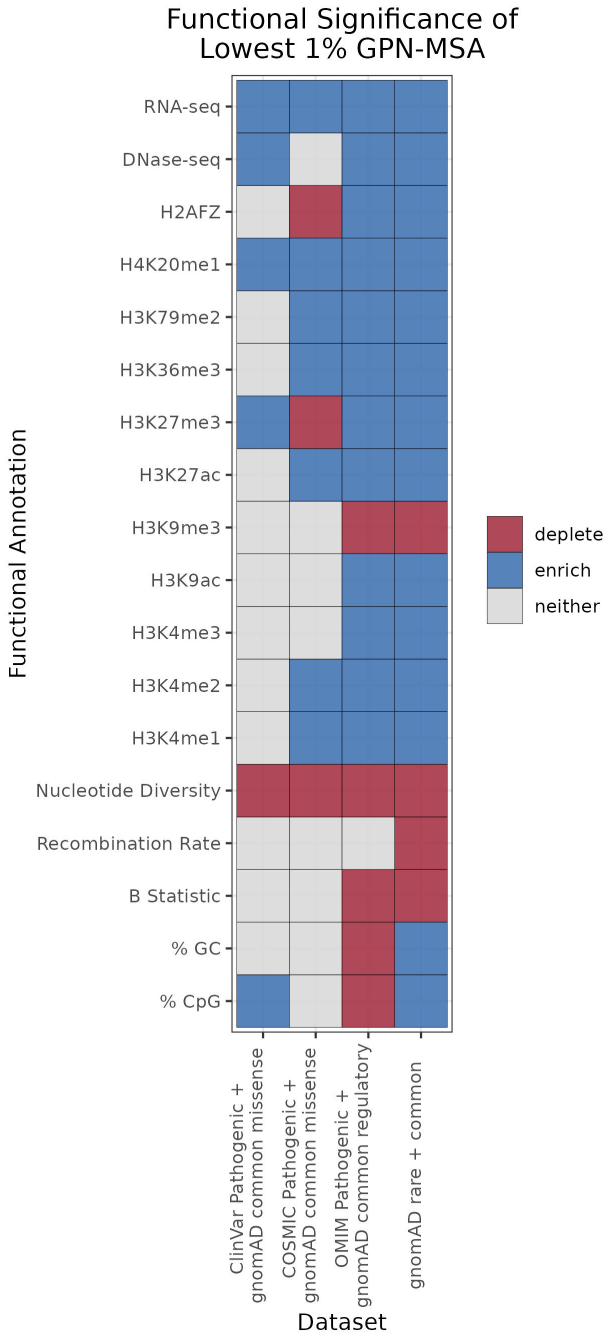


Figure C.3: **Functional enrichment and depletion of deleterious tail.** Significant (Wilcoxon-Mann-Whitney test with FWER controlled at 0.05) enrichments and depletions of functional annotations between the deleterious GPN-MSA tail set of variants and background variants, across four datasets. For tracks like RNA-seq, which require the variant to be exonic, variants without the annotation are not included in the two-sample test.

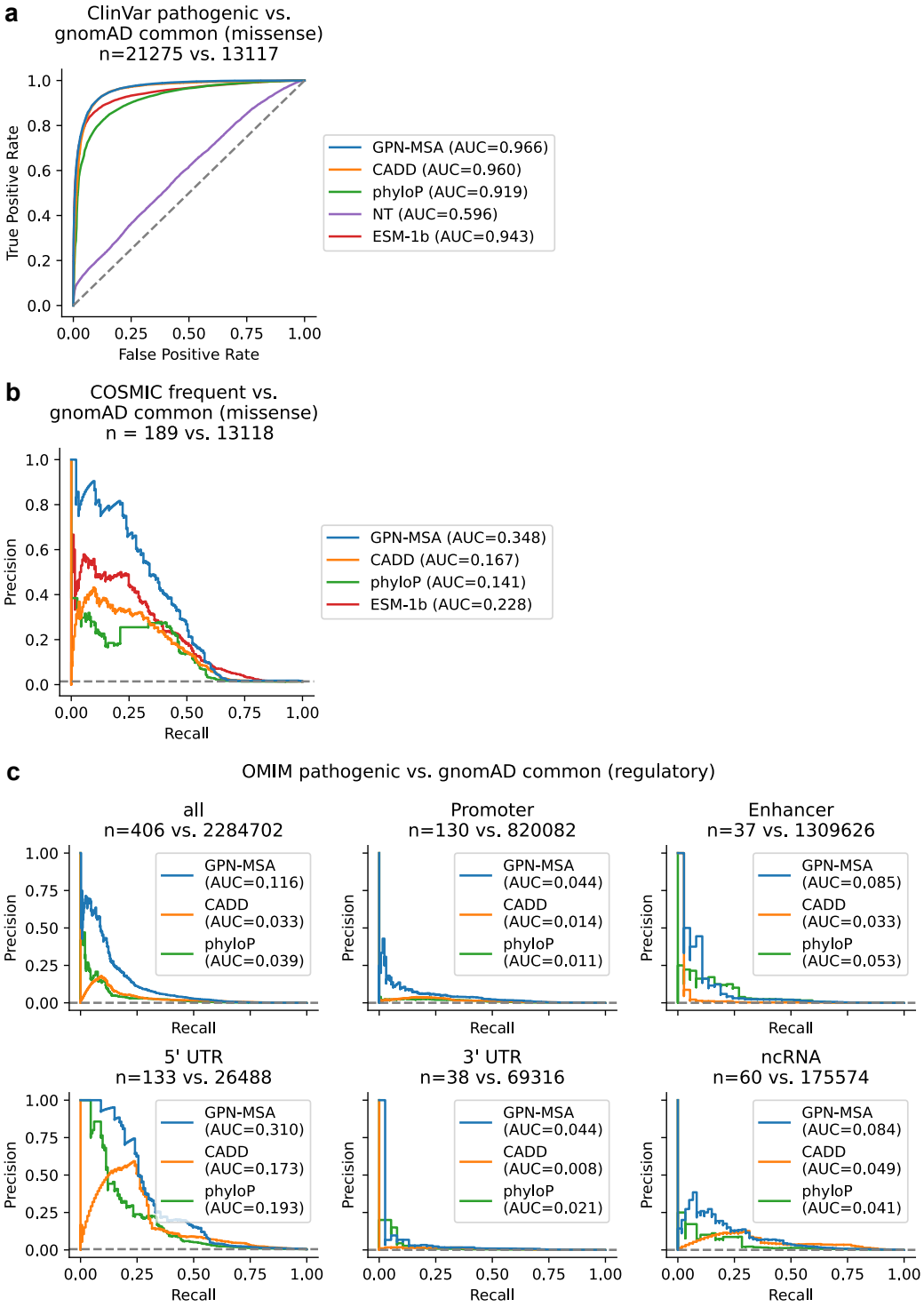


Figure C.4: **Receiver Operating Characteristic and Precision-Recall curves for variant effect prediction.** **(a)** Same setting as Figure 4.2a. **(b)** Same setting as Figure 4.2b. **(c)** Same setting as Figure 4.2c.

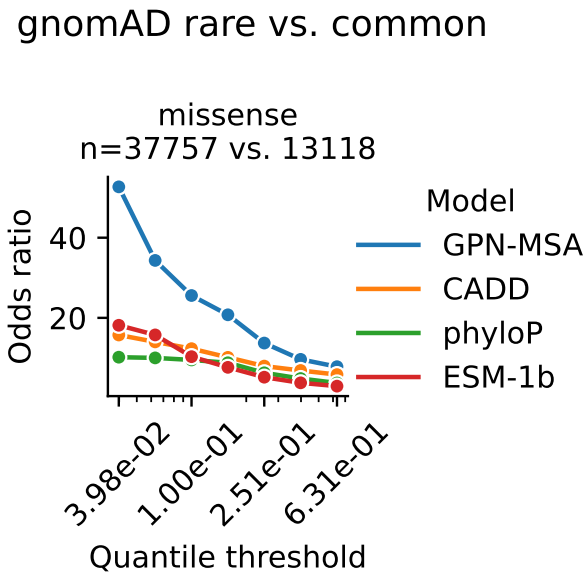


Figure C.5: **Enrichment in rare vs. common gnomAD missense variants.** The same setting as Figure 4.2d with higher quantile thresholds.

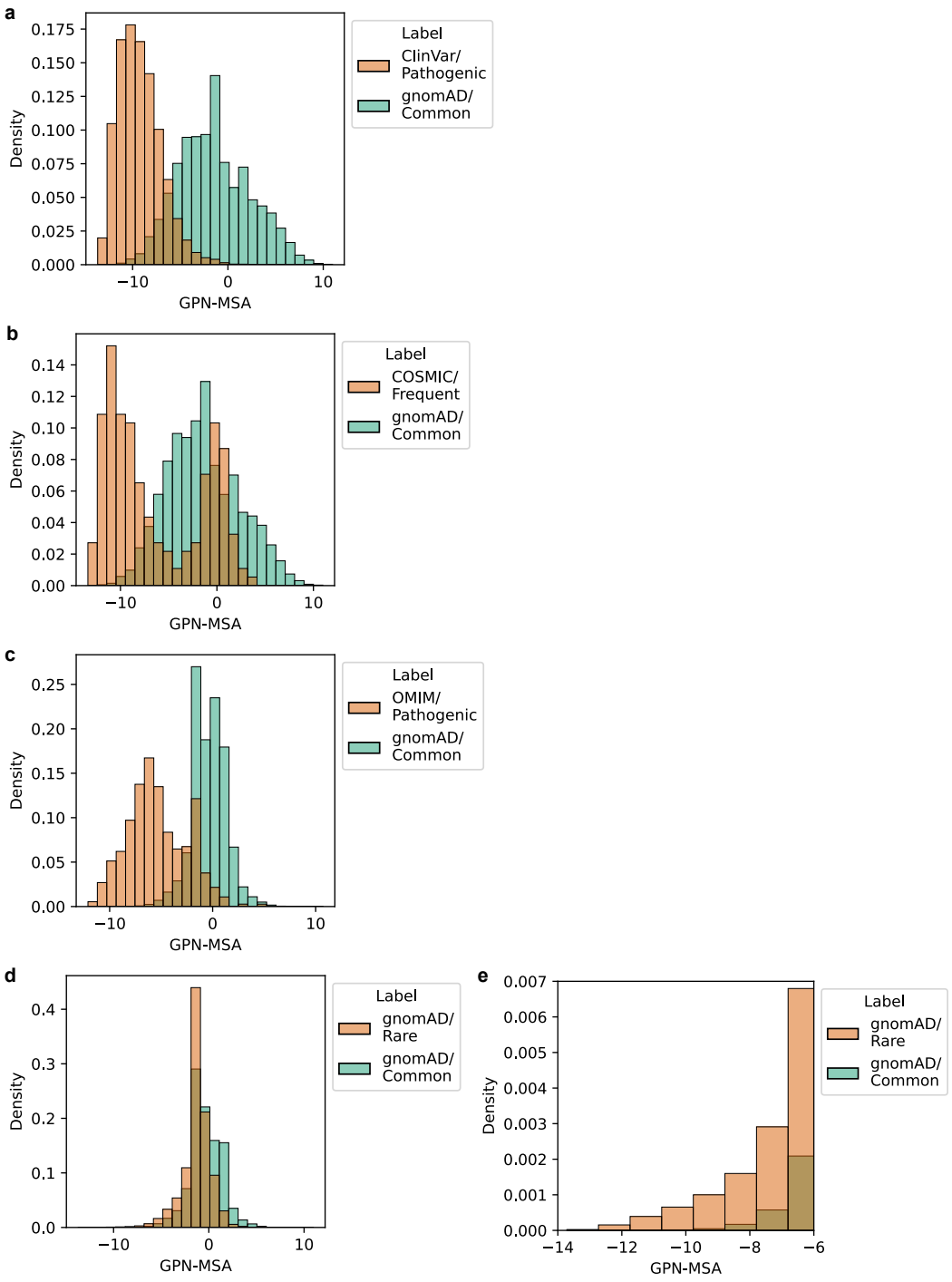


Figure C.6: **Histogram of GPN-MSA scores.** **(a)** Scores for Figure 4.2a. **(b)** Scores for Figure 4.2b. **(c)** Scores for Figure 4.2c. **(d)** Scores for Figure 4.2d. **(e)** A zoomed-in version of (d) highlighting the left tail.

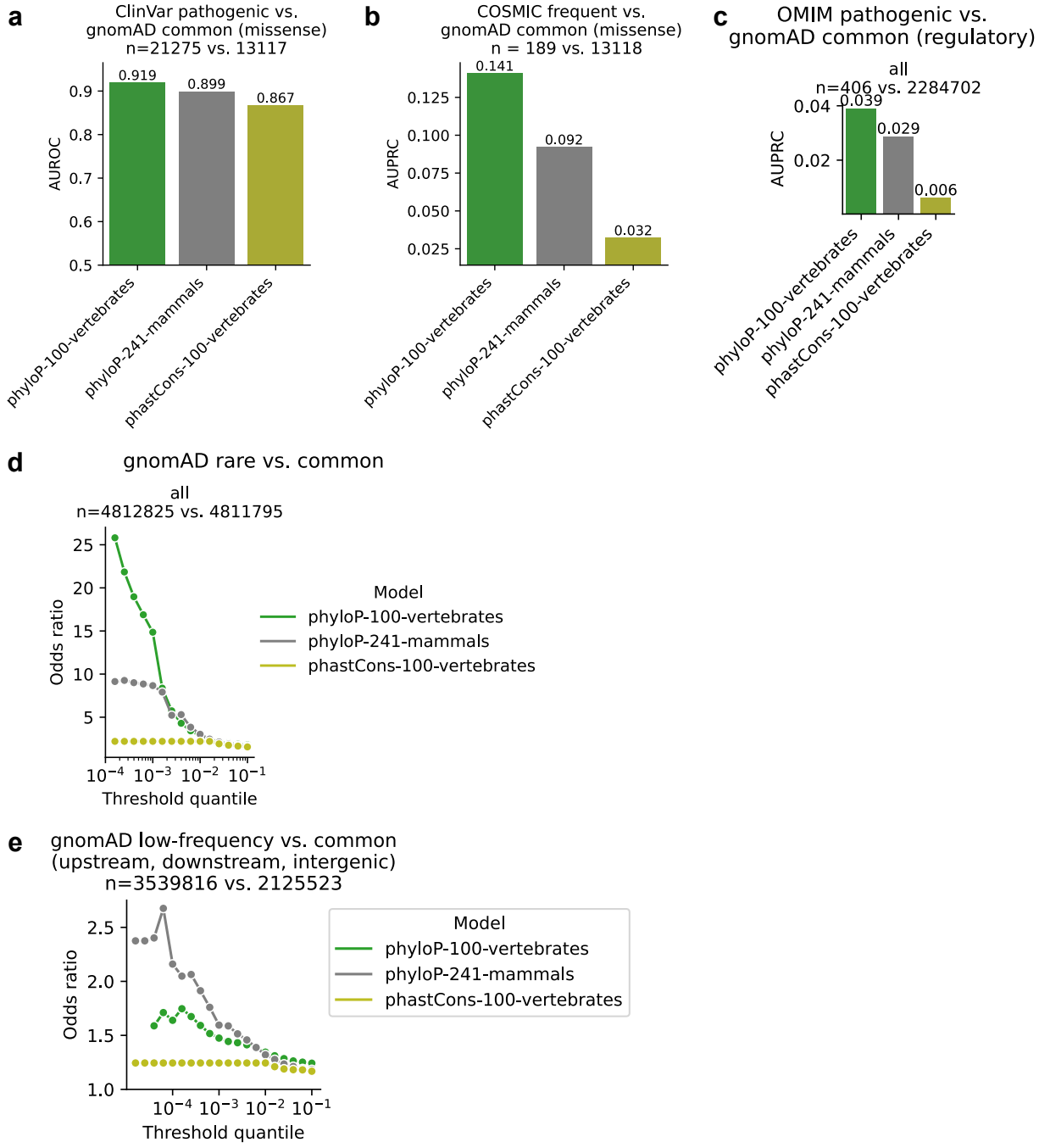


Figure C.7: **Variant effect prediction with conservation scores.** **(a)** Same setting as Figure 4.2a. **(b)** Same setting as Figure 4.2b. **(c)** Same setting as Figure 4.2c. **(d)** Same setting as Figure 4.2d. **(e)** Same setting as Figure 4.2e.

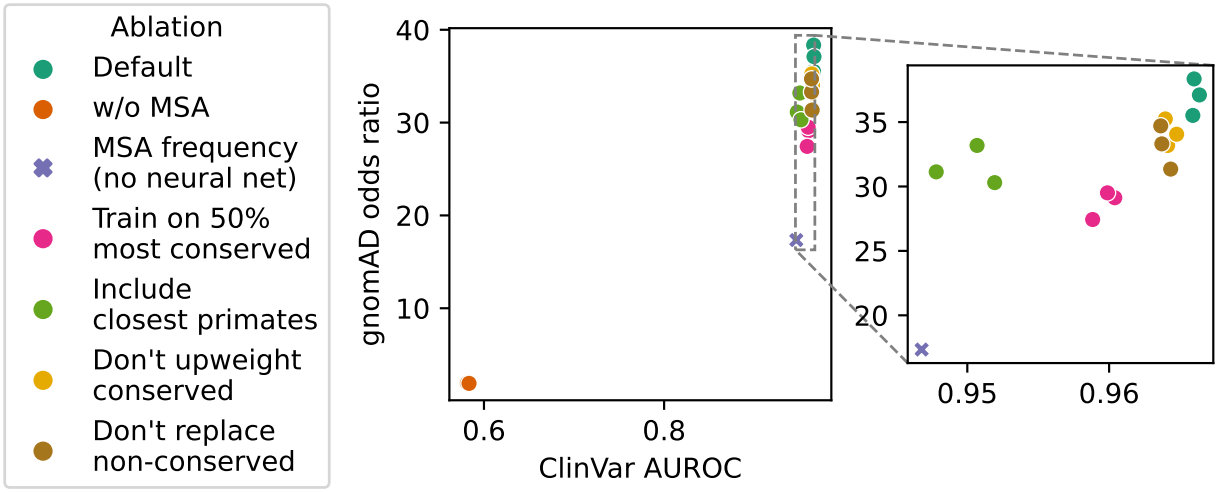


Figure C.8: **Ablation study.** Performance of three random seeds of each independent ablation on two variant effect prediction metrics. ClinVar AUROC: same setting as Figure 4.2a. gnomAD odds ratio: same setting as Figure 4.2d (threshold quantile = 10^{-3}). Ablations include: training solely on the human sequence (w/o MSA), scoring variants based on MSA column frequencies (MSA frequency), expanding training to include 50% most conserved regions, including nearest primates in MSA, not upweighting conserved elements, and not replacing non-conserved positions when calculating loss. Further details in Materials and Methods.

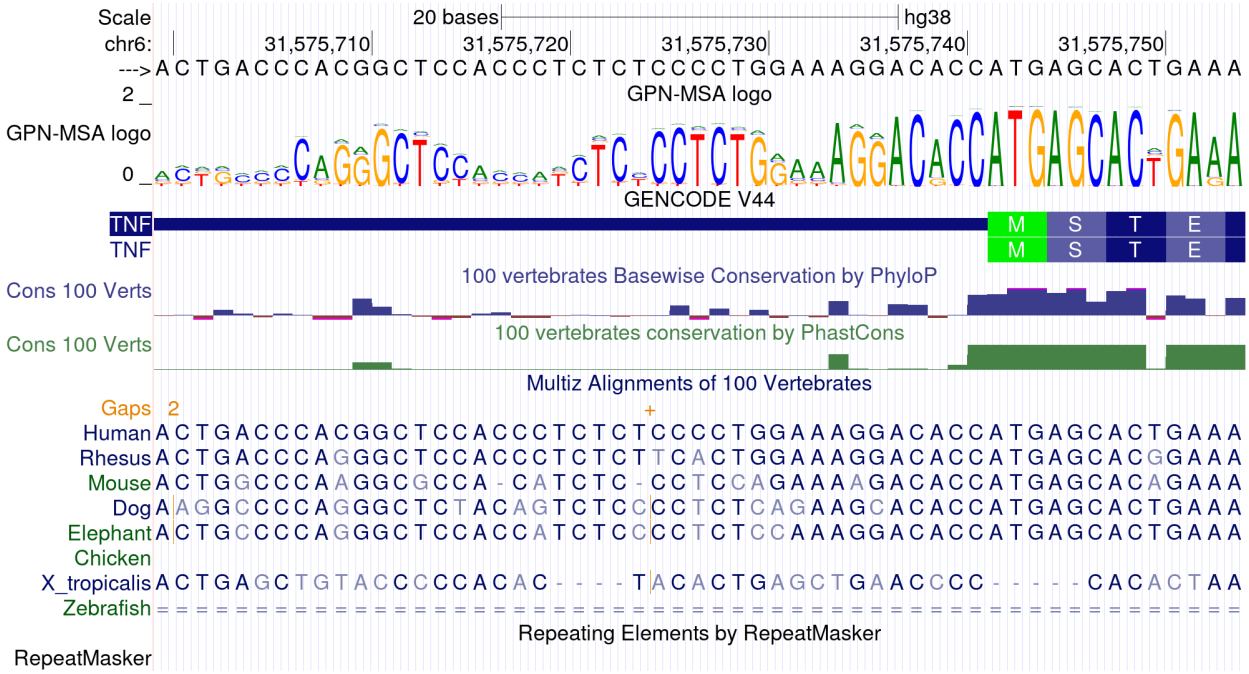


Figure C.9: **GPN-MSA logo track on the UCSC Genome Browser.** Shown region: `chr6:31,575,700-31,575,754`.

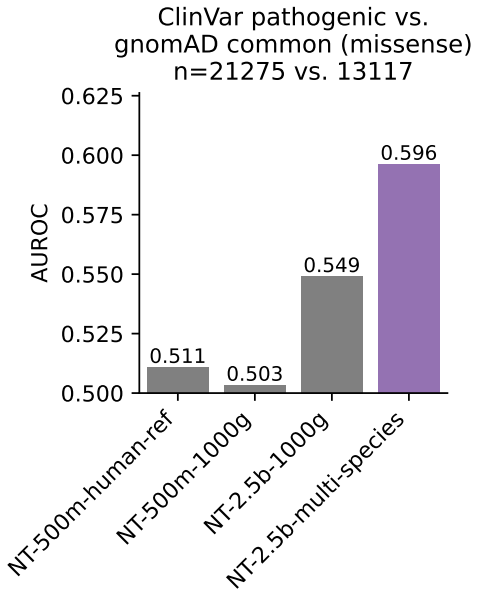


Figure C.10: **Variant effect prediction with Nucleotide Transformer models.** Same setting as Figure 4.2a.

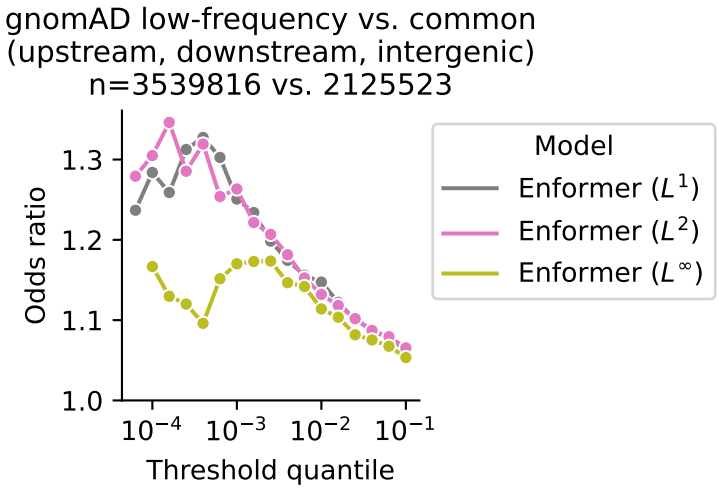


Figure C.11: **Variant effect prediction with different norms of Enformer delta predictions.** Same setting as Figure 4.2e.