Deep Learning for Branch Point Selection in RNA Splicing

Victoria Dean
MIT
vdean@mit.edu

Andrew Delong
Deep Genomics
andrew@deepgenomics.com

Brendan Frey
Deep Genomics
frey@deepgenomics.com

Abstract

Branch point selection is a key step in RNA splicing, yet many many popular splicing analysis tools do not model this mechanism. There were relatively few confirmed branch points until 2015, when a genome-wide map of experimental human branch points was released. This data facilitates, for the first time, modeling branch sites with more sophisticated methods. We used deep learning to model branch site selection, which improved significantly over position-weight matrix models. We show that our branch point model can be used to classify potential disease-causing variants, and can help to improve existing splicing models.

1 Overview

In complex organisms, RNA splicing is fundamental to building proteins from their corresponding genes. Figure 1 illustrates the two catalytic steps of splicing, the outcome of which is a messenger RNA (mRNA) molecule. Each mRNA molecule encodes the instructions for building a protein, so mis-splicing of these instructions can result in dramatically altered proteins, often damaging their function. It is estimated that anywhere from 15–60% of disease-causing mutations are caused by mis-splicing [8]. The splicing machinery recognizes local sequence elements, and so modeling the recognition of these elements is key to predicting how mutations affect splicing.

Branch point recognition (Figure 1) is a necessary sub-step for splicing to occur. If there is a mutation in proximity to an exon's primary branch point, that branch site can become unusable, causing the exon to be. Likewise a mutation can create a branch point for a cryptic splice site that would otherwise go unused, incurring a dramatic change on the resulting mRNA molecule.

In this work, we present results from our exploration of different model architectures for predicting branch site selection. We show that our best model substantially improves over a more traditional baseline model, achieving state-of-the-art performance at identifying functional branch points. We also show that this model improves splicing prediction in general, and can identify disease-causing mutations that disrupt splicing via the branch point mechanism.

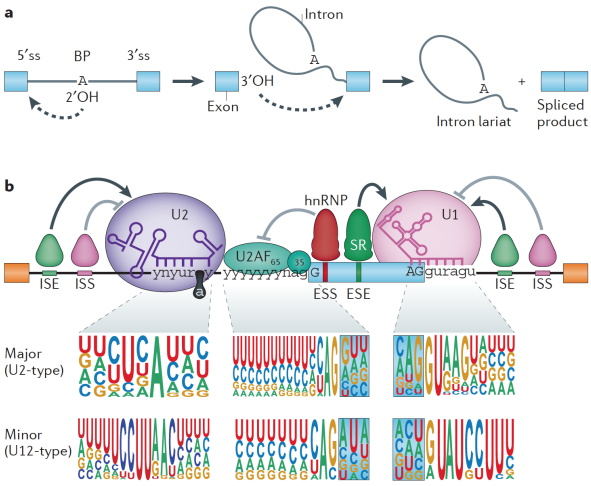

Figure 1: RNA splicing process. (A) The 5' end of the intron is cleaved and bonded to the branch point, forming a lariat. Then, the 3' end is cleaved and the exons are ligated, resulting in an mRNA molecule; the lariat is subsequently degraded and recycled. (B) The spliceosome assembles on the RNA by recognizing sequences for the branch point as well as the 3' and 5' splice sites. Figure from Scotti & Swanson [11].

2 Genome-Wide Branch Point Dataset (Mercer)

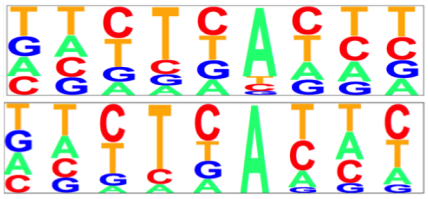

Figure 2: Mercer (top) and SpliceRack (bottom) motifs. The Mercer motif is more flexible, especially at the key adenine.

Research on the rules of branch point selection has been limited by a relatively low-throughput experiments and simple models. In principle, RNA-seq technology can characterize branch points by sequencing the lariats formed during splicing. However, lariats are degraded quickly and appear very rarely in sequencing experiments. Last year, Mercer *et al.* [9] proposed a way to enrich for lariats in RNA-seq, and release a genome-wide dataset of 59,359 human branch points. These branch points are high-confidence, but not exhaustive. The dataset suggests that branch point motifs are not as strict as previously suggested. In the Mercer dataset, the signature central adenine is only present in 78.4% of labeled branch points. Previous reports suggested that adenine was key, and the SpliceRack database for U2 branch points had adenine in 100% of cases [12]; see Figure 2.

The dataset also gives a clearer picture of where branch points are in relation to the 3'SS. Figure 3 shows the distribution of branch points by distance to the splice site. Since the Mercer dataset did not include splice site information, the 3' splice sites were paired with branch points by taking the nearest high-confidence downstream annotated splice site. 80% of all branch points found were between 18 and 38 nucleotides from the splice site.

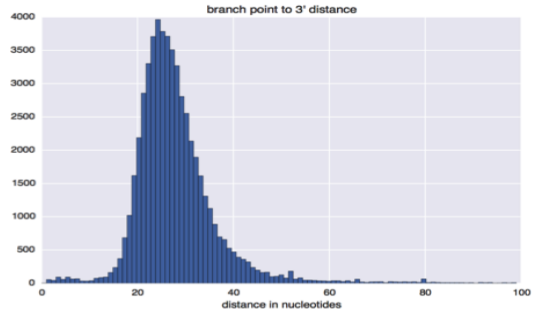

Figure 3: The distribution of Mercer branch points relative to nearest annotated 3'SS. The Mercer dataset did not contain splice site information, so the 3'SS was selected by taking the nearest downstream high-confidence splice site.

3 Branch Point Model

3.1 Position-weight matrix

Our baseline model comprises two parts: (1) a position-weight matrix (PWM) is first trained by aligning experimentally-verified branch points, and (2) a linear model is then trained to combine the PWM score with a distance feature to the downstream acceptor site.

The PWM weights are determined by aligning the 9 nucleotide sequences surrounding each branch point in the Mercer dataset. The nucleotide frequencies are used to create a position-frequency matrix (PFM) with a pseudocount of 1, which is then converted to a PWM in the standard way. PWMs are standard in bioinformatics [13], and they resemble a kind of normalized convolutional filter. A PWM assumes the contribution of each position is independent.

After the PWM baseline is determined, we also use its scores as a feature track in an additional "PWM+dist" baseline. Distance from the branch point to the 3' splice site (3'SS) is an important feature: 80% of branch points in the dataset are between 18 and 38 nucleotides upstream of the 3'SS. So, the PWM+dist baseline combines the distance feature with the PWM score using a neural network. This neural network was trained via 3-fold cross validation on the same training set as our CNN architecture, and was akin to replacing the first convolutional layer with a PWM score track.

3.2 Convolutional neural network

Our best model is a convolutional neural network (CNN) [7] which takes the sequence flanking a 3'SS as input features, along with a feature track encoding the distance from the splice site. The sequence is encoded as a one-hot (e.g. `C` is encoded as $[0, 1, 0, 0]$). We call this model BRANCHR. This configuration outputs independent scores, so it is equivalent to a fully-connected neural network applied at each of the 100 positions. Besides providing faster training and inference on GPUs, handling all positions with a CNN architecture facili-

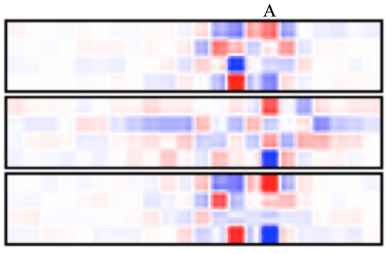

Figure 4: Example filters from the first convolutional layer, sitting on top of the RNA sequence. Nucleotides, from top to bottom, are ACGT.

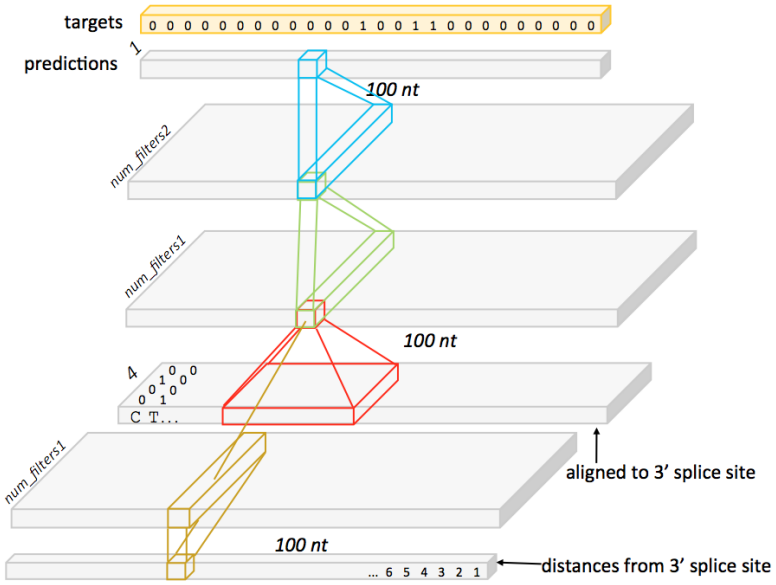

Figure 5: Model architecture. The BRANCHR model takes 100 nucleotides of DNA sequence and a separate distance to 3'SS feature as inputs. The model contains 4 convolutional layers. The goal is binary classification of whether the given location is a branch point.

tates competition among candidate branch points in the late stages of the network; for simplicity, in this work they are treated independently.

To ensure our model did not conflate polypyrimidine tract strength with branch point strength, we configured the network so that the receptive field of each prediction only included 6nt of downstream sequence, precluding any sensitivity to the polypyrimidine tract.

4 Experimental Setup

All Mercer branch point candidates from chromosomes 1 and 2 were held out for the test set. The test set also included a small previous branch point dataset from Gao *et al.* [3]. The Gao dataset is similar to Mercer, but uses an earlier and much lower-throughput branch point detection experiment. The test set contained 42 introns with Gao candidates, and 6,988 introns with Mercer candidates.

The remaining Mercer candidates were split into four folds. The folds were partitioned on an array of branch points sorted by position, so most chromosomes are represented in only a single fold. Three folds were used for training, while the fourth was used for cross-validation.

Models were trained with TensorFlow [1] using the ADAM optimizer [5].

5 Experimental results

5.1 Baseline Comparison

Figure 6 shows how the CNN model compares to the baseline models. The CNN, PWM, and PWM+dist models were all trained on the same data. The CNN achieves an AUC of 0.95 and the PWM+distance baseline achieves 0.93 AUC. However, there are many more non-

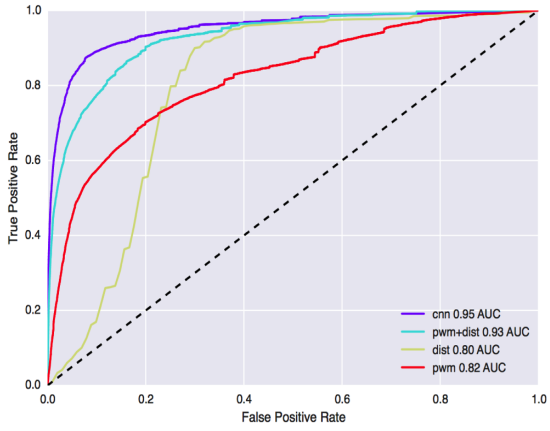

Figure 6: ROC curves show that the CNN performs best on the task of functional branch point prediction. Given the imbalanced nature of this task, the important region on the curve is approximately 5% FPR.

branch points even within the 'sweet spot' (Figure 3). As such, the 5% false positive rate (FPR) is a more relevant operating point than the overall AUC. Here, the CNN gets a True Positive Rate (TPR) of 82%, while the PWM+distance model gets 67.4%—a 1.2x improvement in sensitivity.

5.2 Visualizing effects of disease-causing mutations

The model can also help to classify variants of unknown significance in the intronic region near the 3' splice site, and even re-classify variants previously thought to be benign or pathogenic. BRANCHR scores confirm many pathogenic mutations from literature and ClinVar [6], and show how on how these variants could affect the branch point. We present a visualization similar to [2,4].

Figures 7 and 8 show 2 examples. In Figure 7, BRANCHR predicts "no change" for SNV labeled Benign in ClinVar, whereas applying the Zhang PWM [16] in the 'sweet spot' predicts "modest loss".

In Figure 8, BRANCHR predicts "complete loss" for a variant of unknown significance (VUS); the downstream exon (not shown) has variants labeled Pathogenic for "Ullrich congenital muscular dystrophy" (see ClinVar Variation ID 17181). Further examples omitted due to space constraints.

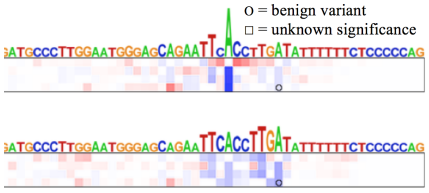

Figure 7: Mutation map example 1. "No change" is predicted on benign variant (an alternative branch point is found).

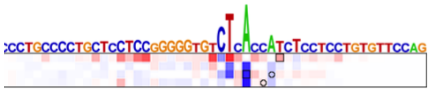

Figure 8: Mutation map example 2. Complete loss predicted on variant of unknown significance.

5.3 Application to Splice Site Recognition

Splice site detection is an important part of genome annotation and variant analysis pipelines. We extended MaxEntScan [15], a splice site detector commonly used for *in silico* analysis in research and clinical settings. The MaxEntScan 3' acceptor model analyses up 20bp into the intron, so is sensitive to the polypyrimidine tract but not to branch points. Complex splicing models [14] also do not model the branch point.

Data set for splice site recognition. Our positive set (real acceptor sites) is derived from GENCODE v25 comprehensive lifted to hg19. First, acceptors on exons annotated as protein coding and having transcript support levels 1 or 2 were collected. Only constitutive exons were retained, *i.e.* included in every transcript satisfying the same criteria. Next, a set of negatives was extracted by searching for all `AG` dinucleotides within a 501nt window centered on the `G` of each positive acceptor, then filtering out any positions that appear in any annotated transcript (of any support level). Filtering out unannotated acceptors, *e.g.* Intropolis [10], would make the negative set even cleaner.

Modeling and results. We trained a small neural network (64 sigmoid units), using chromosomes 1 and 2 as held-out test set. We also stratified the test set into "hard" acceptors with MaxEntScan score in range $[-5, 5]$; this contained 4,793 positive and 413,512 negative acceptors (approx. 20% of full test set). Each run was repeated 10 times. The results in Table 1 show a substantial improvement in sensitivity for these otherwise difficult acceptor sites. Note that the genome has highly imbalanced (more negatives), so the low-FPR regime is again most relevant to clinical and research utility.

Table 1: Comparison of accuracy at recognizing splice sites. When all candidate splice sites are considered, dominated by easy positives (strong core splicing signal) and easy negatives (no core splicing signal beyond `AG` dinucleotide), BRANCHR adds a consistent but modest boost. When evaluation focuses on the hardest 22% of cases (see text), BRANCHR provides 1.2–2.5x sensitivity.

Model (test set)	TPR @ 0.1% FPR	TPR @ 1.0% FPR	TPR @ 5.0% FPR
MaxEntScan only (all)	15.3%	54.5%	88.2%
MaxEnt+BRANCHR (all)	18.6% (±0.00)	57.6% (±0.00)	89.6% (±0.00)
Sensitivity improvement	1.2x	1.06x	1.02x
MaxEntScan only **(hard)**	1.1%	9.7%	36.0%
MaxEnt+BRANCHR **(hard)**	2.8% (±0.00)	16.8% (±0.02)	42.6% (±0.03)
Sensitivity improvement	**2.5x**	**1.73x**	**1.18x**

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*, 33:831–838, 2015.

[3] K. Gao, A. Masuda, T. Matsuura, and K. Ohno. Human branch point consensus sequence is yunay. *Nucleic acids research*, 36(7):2257–2267, 2008.

[4] D. R. Kelley, J. Snoek, and J. Rinn. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 2016.

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[6] M. J. Landrum, J. M. Lee, M. Benson, G. Brown, C. Chao, S. Chitipiralla, B. Gu, J. Hart, D. Hoffman, J. Hoover, W. Jang, K. Katz, M. Ovetsky, G. Riley, A. Sethi, R. Tully, R. Villamarin-Salomon, W. Rubinstein, and D. R. Maglott. Clinvar: public archive of interpretations of clinically relevant variants. *Nucleic Acids Research*, 2015.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[8] K. H. Lim, L. Ferraris, M. E. Filloux, B. J. Raphael, and W. G. Fairbrother. Using positional distribution to identify splicing elements and predict pre-mrna processing defects in human genes. *Proceedings of the National Academy of Sciences*, 108(27):11093–11098, 2011.

[9] T. R. Mercer, M. B. Clark, S. B. Andersen, M. E. Brunck, W. Haerty, J. Crawford, R. J. Taft, L. K. Nielsen, M. E. Dinger, and J. S. Mattick. Genome-wide discovery of human splicing branchpoints. *Genome research*, 25(2):290–303, 2015.

[10] A. Nellore, A. E. Jaffe, J.-P. Fortin, J. Alquicira-Hernández, L. Collado-Torres, S. Wang, R. A. Phillips, N. Karbhari, K. D. Hansen, B. Langmead, and J. T. Leek. Human splicing diversity across the sequence read archive. *bioRxiv*, 2016.

[11] M. M. Scotti and M. S. Swanson. Rna mis-splicing in disease. *Nature Reviews Genetics*, 17(1):19–32, 2016.

[12] N. Sheth, X. Roca, M. L. Hastings, T. Roeder, A. R. Krainer, and R. Sachidanandam. Comprehensive splice-site analysis using comparative genomics. *Nucleic acids research*, 34(14):3955–3967, 2006.

[13] G. D. Stormo. Dna binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, 2000.

[14] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes, Q. Morris, Y. Barash, A. R. Krainer, N. Jojic, S. W. Scherer, B. J. Blencowe, and B. J. Frey. The human splicing code reveals new insights into the genetic determinants of disease. 2015.

[15] G. Yeo and C. B. Burge. Maximum entropy modeling of short sequence motifs with applications to rna splicing signals. *Journal of Computational Biology*, 11:377–394, 2004.

[16] M. Q. Zhang. Statistical features of human exons and their flanking regions. *Human Molecular Genetics*, 7(5):919–932, 1998.