Yang et al. *Algorithms for Molecular Biology(2026)21:1*
https://doi.org/10.1186/s13015-025-00293-7

Algorithms for
Molecular Biology

RESEARCH Open Access

Improving spliced alignment by modeling splice sites with deep learning

Siying Yang[1,2], Neng Huang[1,2] and Heng Li[1,2,3*]

*Correspondence:
hli@ds.dfci.harvard.edu
[1]Department of Biomedical Informatics, Harvard Medical School, 10 Shattuck St, Boston, MA 02215, USA
[2]Department of Data Science, Dana-Farber Cancer Institute, 450 Brookline Ave, Boston, MA 02215, USA
[3]Broad Institute of MIT and Harvard, 415 Main St, Cambridge, MA 02142, USA

Abstract

Motivation Spliced alignment refers to the alignment of messenger RNA (mRNA) or protein sequences to eukaryotic genomes. It plays a critical role in gene annotation and the study of gene functions. Accurate spliced alignment demands sophisticated modeling of splice sites, but current aligners use simple models, which may affect their accuracy given dissimilar sequences.

Results We implemented minisplice to learn splice signals with a one-dimensional convolutional neural network (1D-CNN) and trained a model with 7026 parameters for vertebrate and insect genomes. It captures conserved splice signals across phyla and reveals GC-rich introns specific to mammals and birds. We used this model to estimate the empirical splicing probability for every `GT` and `AG` in genomes, and modified minimap2 and miniprot to leverage pre-computed splicing probability during alignment. Evaluation on human long-read RNA-seq data and cross-species protein datasets showed our method greatly improves the junction accuracy especially for noisy long RNA-seq reads and proteins of distant homology.

Availability and implementation https://github.com/lh3/minisplice

Introduction

RNA splicing is the process of removing introns from precursor mRNAs (pre-mRNAs). It is widespread in eukaryotic genomes [28]. In human, for example, each protein-coding gene contains 9.4 introns on average; >98% of introns start with `GT` on the genome (or more precisely `GU` on the pre-mRNA) and >99% end with `AG` [35]. On the other hand, there are hundreds of millions of di-nucleotide `GT` or `AG` in the human genome. Only ∼0.1% of them are real splice sites. Identifying real splice sites, which is required for gene annotation, is challenging due to the low signal-to-noise ratio.

To annotate splice sites and genes in a new genome, we can perform RNA sequencing (RNA-seq) and align mRNA sequences to the target genome. This approach does not work well for genes lowly expressed in sequenced tissues. A complementary strategy is to align mRNA or protein sequences from other species to the target genome. Spliced alignment through introns is essential in both cases.

It is important to look for splice signals during spliced alignment as the residue alignment around a splice site can be ambiguous. For example, the three alignments in Fig. 1 are

Fig. 1 Ambiguity in spliced alignment. The same genome-mRNA sequence pair can be aligned differently without mismatches or gaps

equally good if we ignore splice signals. However, as the putative intron in alignment (1) does not match the `GT..AG` signal, it is unlikely to be real. While both (2) and (3) match the signal, alignment (3) is more probable because it fits the splice consensus `GTR..YAG` better [22,23], where "`R`" stands for an `A` or a `G` base and "`Y`" for `C` or `T`. In this toy example, the query sequence matches the reference perfectly in all three cases. On real data, spliced aligners may introduce extra mismatches and gaps to reach splice sites. The splice model has a major influence on the final alignment especially for diverged seqeences when aligners need to choose between multiple similarly scored alignments around splice junctions.

Position weight matrix (PWM) is a classical method for modeling splice signals [37]. It however does not perform well because it cannot capture dependencies between positions [9] or model regulatory motifs such as branch points that do not have fixed positions. Many models have been developed to overcome the limitation of PWM [10]. In recent years, deep learning is gaining attraction and has been shown to outperform traditional methods [1,15,51]. Early deep learning models are small with only a few 1D-CNN layers. Later models are larger, composed of residual blocks [11,25,45,49] or transformer blocks [12,47]. It is also possible to fine tune genomic large-language models for splice site prediction [8,14,33]. Developed for general purposes, large-language models are computationally demanding and may be overkill if we just use them to predict splice sites.

At the same time, Helixer [21] and Tiberius [16] combined Hidden Markov Models (HMM) and deep-learning models for *ab initio* gene prediction and achieved high accuracy. They however do not report alternative isoforms and are not suitable for genomes with frequent alternative splicing. Mainstream gene annotation pipelines such as Ensembl and NCBI EGAP/EGAPx still heavily rely on alignment.

While qualified spliced aligners all look for the `GT..AG` splice signal, they model additional flanking sequences differently. Intra-species mRNA-to-genome aligners such as BLAT [27], GMAP [42] and Splign [26] often do not model extra sequences beyond `GT..AG` because alignment itself provides strong evidence and ambiguity shown in Fig 1 is rare. Minimap2 [29] prefers the `GTR..YAG` consensus [22]. This helps to improve the alignment of noisy long RNA-seq reads. GSNAP [41] integrated MaxEnt [46] for scoring novel splice sites, and the feature was also added to GMAP. Protein-to-genome aligners tend to employ better models due to more ambiguous alignment given distant

homologs. Miniprot [31] considers rarer `GC..AG` and `AT..AC` splice sites and optionally prioritizes on the `G|GTR..YNYAG` consensus common in vertebrate and insect [23], where "|" indicates splice boundaries. Exonerate [36], Spaln [17,18,24] and the original GeneWise [5] use PWM. GeneSeqer [39] and GenomeThreader [19] apply more advanced models [6,7]. Deep learning models have been applied to refining splice sites as a post-processing step [11,43] but have not been integrated into spliced aligners until recently [3].

In this article, we introduce minisplice, a command-line tool implemented in C, that learns splice signals and scores candidate splice sites with a small 1D-CNN model. We have modified minimap2 and miniprot to take the splice scores as input for improved spliced alignment. Importantly, we aim to develop a simple model that is more capable than PWM and is still easy to deploy; we do not intend to compete with the best splice models which are orders of magnitude larger.

Methods

Our overall workflow consists of three steps: training, prediction and alignment (Fig. 2a). First, we train a deep learning model and transform scores outputted by the model to empirical probabilities using known gene annotation. Second, given a target genome to which mRNA or protein sequences will be aligned, we predict the empirical probability of splice sites at each `GT` or `AG` in the genome and output the logarithm-scaled splice scores to a file. Third, when aligning mRNA sequences with minimap2 [29] or aligning protein sequences with miniprot [31], we feed the precomputed splice scores to the aligners which use the scores during dynamic-programming-based residue alignment. This procedure will improve the accuracy around splice sites. As we will show later, we can merge the training data from several vertebrates and insect to obtain a model working well across phyla. We do not need to train often.

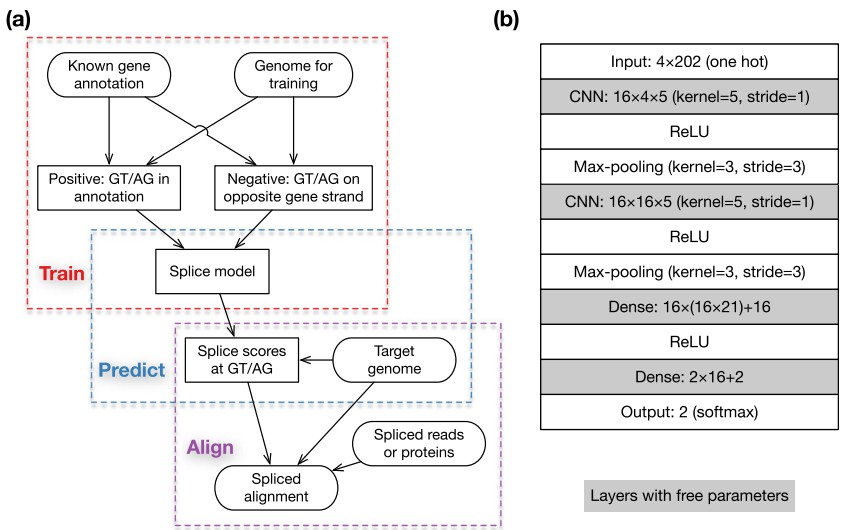

Fig. 2 Method overview. **a** Overall workflow. Training and prediction are done by minisplice. Alignment is done by minimap2 or miniprot. **b** Default model architecture and parameterization, including tensor shapes. For example, "4 × 202" indicates the input is a 4 × 202 matrix; "16 × 4 × 5" suggests the 1D-CNN layer has a kernel size of 5 and outputs 16 features. Shaded boxes contain free parameters

Generating training data

Consistent with the literature, we call the 5'-end of an intron as the *donor* site and 3'-end as the *acceptor* site. Minisplice takes the genome sequence in the FASTA format and gene annotation in the 12-column BED format (BED12) as input. It provides a script to convert annotation in GTF or GFF3 format to BED12. To generate training data, minisplice inspects each annotated donor site and labels it as a positive site if its sequence is `GT`; similarly, minisplice labels a positive acceptor if it is an annotated `AG`. We ignore donor sites without `GT` or acceptor sites without `AG` because other types of splice sites are rare [35]. Miniprot and minimap2 still consider rarer `GC..AG` or `AT..AC` at the alignment step.

Due to potentially incomplete gene annotation in non-model organisms, a `GT` dinucleotide could be a real unannotated donor site. To alleviate this issue, we label an unannotated `GT` as a negative donor site only if it comes from the opposite strand of an annotated gene [11]. In the rare case when two genes on opposite strands overlap with each other, we ignore the overlapping region. To balance positive and negative sites, we randomly select a subset of negative sites such that the positive-to-negative ratio is 1:3. For each positive or negative donor site, we extract 100bp immediately before and after `GT`. The total length of sequences used for training is thus 202bp. Negative acceptor sequences are generated similarly. We also experimented 102bp and 302bp window sizes.

Model architecture

Minisplice uses a model with two 1D-CNN layers (Fig. 2b). The architecture is common among small models for splice site prediction [48]. The default model uses 16 features at both CNN layers and has 7026 free parameters in total (sum of numbers in shaded boxes). During development, we experimented alternative models with different kernel sizes, more CNN or dense layers, optional dropout layers or more parameters. We chose a relatively small model in the end as it is more efficient to deploy. Because minimap2 and miniprot score each donor or acceptor site independently, we also model splice sites independently.

Training and testing

We use 80% of genes on the odd chromosomes or contigs for training and reserve the rest 20% for validation. We stop training if the validation cost increases over several epochs. Recall that we intend to predict splice sites across the whole genome but when generating training data, we downsample negative `GT`/`AG` to a small fraction. To test the model accuracy in a setting closer to the prediction task, we apply the trained model to every `GT`/`AG` on the even chromosomes or contigs and compare the prediction to the known gene annotation to measure accuracy. In comparison to training data, testing data may contain errors in known gene annotation: missing junctions in the annotation would appear to be false positives (FPs), while falsely annotated junctions would look false negatives (FNs). It is not straightforward to compare accuracy across species.

Transforming raw model scores to probabilities

With the 'softmax' operator at the end, the model scores each candidate splice site with a number between 0 and 1, the higher the better. This score is not a probability in particular

when the property of the training data is distinct from our intended application. We need to transform this score to probability to work with the probability-based scoring system of miniprot.

We evenly divide raw model scores into b bins (50 by default) such that raw score $t \in [0, 1)$ is assigned to bin $i = \lfloor tb \rfloor$, $i = 0, 1, \ldots, b-1$. Let P_i be the number of annotated splice sites scored to bin i and N_i be the number of unannotated `GT`/`AG` sites scored to bin i. $P_i/(P_i + N_i)$ is the empirical probability of a candidate site in bin i being real. Let $P = \sum_i P_i$ and $N = \sum_i N_i$. Given raw score t, the transformed score is

$$s(t) \triangleq 2\log_2\left(\frac{P_{\lfloor tb \rfloor}}{P_{\lfloor tb \rfloor} + N_{\lfloor tb \rfloor}} \cdot \frac{P+N}{P}\right) \tag{1}$$

It computes the log odds of the probability estimated with the deep learning model versus with the null model that assumes every `GT`/`AG` having equal probability of being real. The $2\log_2$ scaling is imposed by BLOSUM scoring matrices [20] which miniprot uses.

Aligner integration

Minimap2 [29] uses the following equation for spliced alignment:

$$\begin{cases} H_{ij} = \max\{H_{i-1,j-1} + s(i,j), E_{ij}, F_{ij}, \tilde{E}_{ij} - a(i)\} \\ E_{i+1,j} = \max\{H_{ij} - q, E_{ij}\} - e \\ F_{i,j+1} = \max\{H_{ij} - q, F_{ij}\} - e \\ \tilde{E}_{i+1,j} = \max\{H_{ij} - d(i) - \tilde{q}, \tilde{E}_{ij}\} \end{cases} \tag{2}$$

where q is the gap open penalty, e the gap extension penalty and $s(i, j)$ gives the substitution score between the i-th position on the reference and the j-th position on the query sequence. $d(i)$ and $a(i)$ are the donor and acceptor scores, respectively, calculated with Eq. (1). Miniprot [31] uses a more complex equation which has the same donor and acceptor score functions.

Implementation

Minisplice is implemented in the C programming language with the only dependency being zlib for reading compressed files. It uses a deep-learning library we developed earlier for identifying human centromeric repeats [30]. Minisplice outputs splice scores in a TAB-delimited format like:

```
chr2    4184146    +    A    9
chr2    4184167    +    A    -5
chr2    4184191    -    D    5
chr2    4184199    +    A    -5
chr2    4184213    +    D    3
```

where the second column corresponds to the offset of the splice boundary and the last column gives the splice score. We modified minimap2 and miniprot to optionally take such a file as input and use the splice scores during residue alignment. Notably, minimap2 and miniprot do not directly depend on minisplice. Users can provide splice scores estimated by other means in principle.

Table 1 Datasets

Label	Species	Accession
human†	*Homo sapiens*	GCA_000001405.29
mouse†*	*Mus musculus*	GCA_000001635.9
chicken†*	*Gallus gallus*	GCA_016699485.1
zebrafish†*	*Danio rerio*	GCA_000002035.4
fruitfly†*	*Drosophila melanogaster*	GCA_000001215.4
mosquito†	*Anopheles gambiae*	GCA_943734735.2
mCanLup*	*Canis lupus baileyi*	GCF_048164855.1
mLagAlb*	*Lagenorhynchus albirostris*	GCF_949774975.1
bAccGen*	*Astur gentilis*	GCF_929443795.1
bAnaAcu	*Anas acuta*	GCF_963932015.1
bTaeGut	*Taeniopygia guttata*	GCF_048771995.1
rEmyOrb	*Emys orbicularis*	GCF_028017835.1
aDenEbr	*Dendropsophus ebraccatus*	GCF_027789765.1
fCarCar*	*Carassius carassius*	GCF_963082965.1
fPunPun	*Pungitius pungitius*	GCF_949316345.1
sMobHyp	*Mobula hypostoma*	GCF_963921235.1
icTenMoli*	*Tenebrio molitor*	GCF_963966145.1
idCalVici*	*Calliphora vicina*	GCF_958450345.1
idStoCalc	*Stomoxys calcitrans*	GCF_963082655.1
ihPlaCitr*	*Planococcus citri*	GCF_950023065.1
ilCydFagi*	*Cydia fagiglandana*	GCF_963556715.1
ilOstNubi	*Ostrinia nubilalis*	GCF_963855985.1
iyBomTerr*	*Bombus terrestris*	GCF_910591885.1
iyVesCrab	*Vespa crabro*	GCF_910589235.1

Ensembl or Gencode annotations were used for model organisms (marked by "†"); RefSeq annotations used for non-model organisms whose labels follow the naming standard developed by the Darwin Tree of Life Project: prefix "m" stands for mammals, "b" for birds, "r" for reptiles, "a" for amphibians, "f" for fish, "s" for sharks, "ic" for order Coleoptera (beetles), "id" for Diptera (flies), "ih" for Hemiptera (true bugs), "il" for Lepidoptera (butterflies and moths), "iy" for Hymenoptera (bees and ants). Species marked by "*" are used for training cross-species models

Results

We evaluated the accuracy of trained models using Receiver Operating Characteristic (ROC) curves where we computed the true positive rate and false positive rate at different thresholds on raw model scores. In the ROC plot, we focused on the region with sensitivity above 50% and false positive rate below 10% because we intend to improve spliced alignment in this region. For each curve, we calculated rAUC, which is the area under the ROC curve restricted to and scaled by this region. Due to scaling, rAUC $\in [0, 1]$.

To find a small model that is generalized to multiple species and is fast to deploy, we experimented models under several settings. Our final model is trained on six insect genomes from five orders and seven vertebrate genomes. We did not train a plant model because we are less familiar with the plant phylogeny.

Datasets

We acquired the genome sequences and gene annotations for six model organisms and 16 non-model organisms (Table 1). For model organisms, only chicken is annotated with the Ensembl pipeline; others are annotated by third parties. For non-model organisms, we intentionally chose species that have PacBio HiFi assemblies and are annotated by

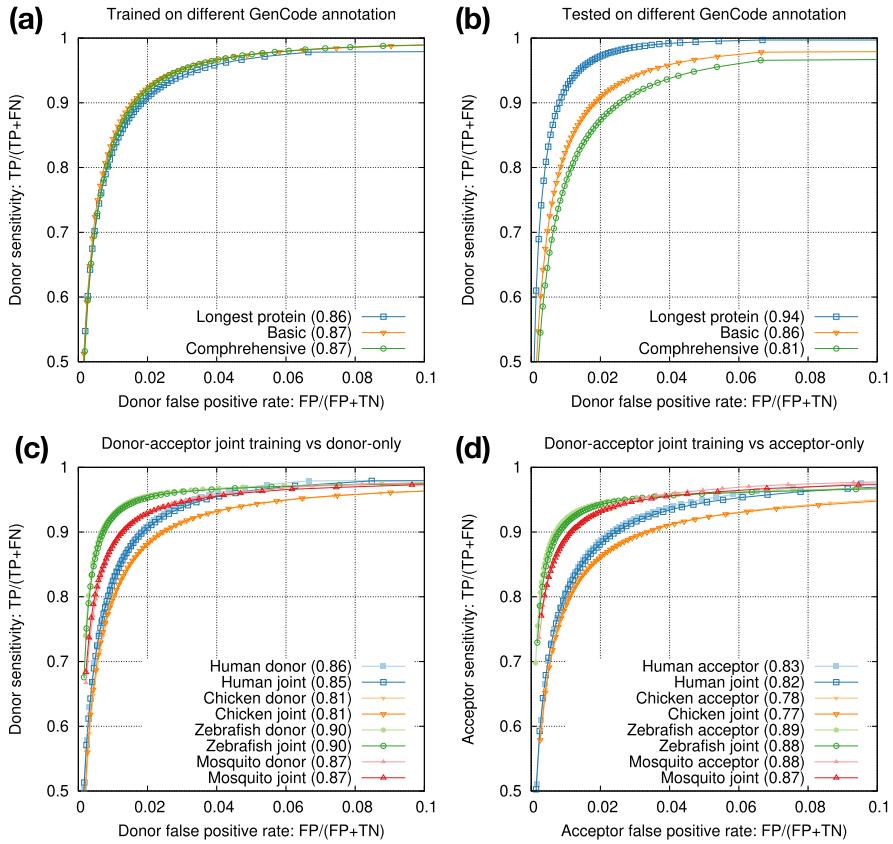

Fig. 3 Intra-species training. **a** Models trained on different human annotations and tested on the Gencode basic annotation. Numbers in parentheses denote rAUC, which is the Area Under the Curve restricted to the plotted region. **b** Model trained from the longest human protein-coding transcripts and tested on different human annotations. **c** Donor-only training versus joint donor-acceptor training. **d** Acceptor-only training versus joint donor-acceptor training

both RefSeq and Ensembl if possible. Only aDenEbr and sMobHyp do not have Ensembl annotations.

Intra-species training

Gencode provides a smaller set of "basic" gene annotations and a larger set of "comprehensive" annotations. We also have an option to select the longest transcript of each protein-coding gene for high-fidelity splice sites. We found training data had minor effect on testing accuracy (Fig. 3a) but annotations used for testing had larger effect (Fig. 3b). We decided to train on splice sites from the longest protein-coding transcripts for training as they are most accurate, and to test on basic annotations because non-model organisms probably do not have annotations comparable to comprehensive Gencode annotations.

The two central bases in donor training data are always `GT` and the two bases in acceptor training data are always `AG`. We speculated 1D-CNN models could easily learn the difference, so we mixed donor and acceptor training data and trained one joint model for each species. The joint models achieved nearly the same accuracy as separate donor or acceptor models (Fig. 3c, d). In later experiments, we thus always trained one joint model to simplify the training process.

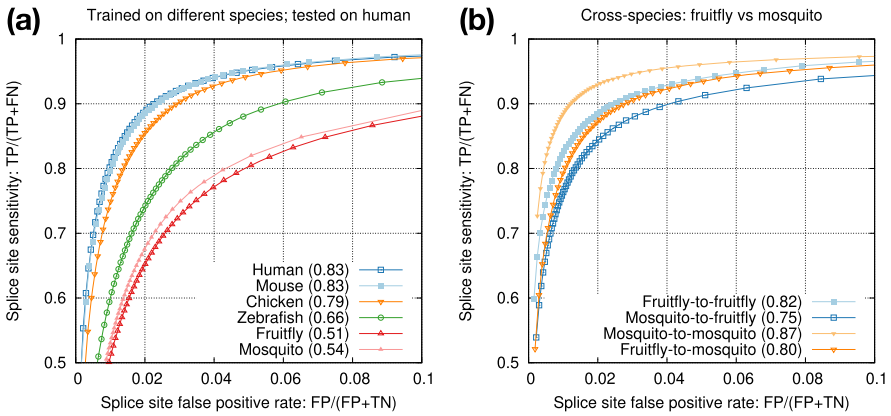

Fig. 4 Cross-species training. **a** Models trained from different species and tested on human. **b** Intra- versus cross-species training with mosquito and fruitfly

Cross-species training

Our end goal is to improve spliced alignment accuracy for species without known annotations. Training and prediction are often applied to different species. To test how well a model trained from one species can predict splice sites in a different species, we applied models trained from model organisms to human (Fig. 4a). We can see the test accuracy drops quickly with increased evolution distance. The mouse model is almost good as the human model because mammals are closely related. We also applied the mosquito model to the fruitfly genome (Fig. 4b). The test accuracy is lower than the accuracy we obtain with the fruitfly model, but the drop is much smaller in comparison to applying the mosquito model to human. These experiments suggested we can achieve reasonable accuracy with a model trained from a closely related species.

We went a step further by combining the training data across multiple vertebrate and insect species and derived model vi2, which was trained from six vertebrate and seven insect genomes marked in Table 1. Although this model is not as accurate as the species-specific model trained from individual species itself (Fig. 5a, b), it is better than applying an insect model to human (Fig. 4a). Note that vi2 is not trained on amphibian or shark genomes but it still accurately predicts splice sites in aDenEbr and sMobHyp. It is capturing common signals across large evolutionary distance while reducing overfitting to individual species. We also trained a vertebrate-only model (v2) and an insect-only model (i2). They outperformed vi2 for vertebrate and insect genomes, respectively (Fig. 5c, d), but we deemed the improvement is small and outweighed by the convenience of having one model across vertebrate and insect genomes.

Model vi2 considers 202bp sequences around splice sites. We tried to reduce the window size to 102bp but the accuracy dropped (Fig. 6a). Increasing the window size to 302bp, on the other hand, only had a small effect (Fig. 6b). To understand what signals the model is learning, we visualized the activation rates at the last max-pooling layer which has 16 features (Fig. 6c,d). While some features such as 4 and 7 focus on signals at splice sites, other features such as 2 and 13 likely capture compositions of introns and exons. Intronic sequences around acceptors appear to provide more signals than around donors. This might be related to branch points which are located within tens of basepairs upstream to acceptors. The activation difference fades away beyond 100bp around splice sites especially

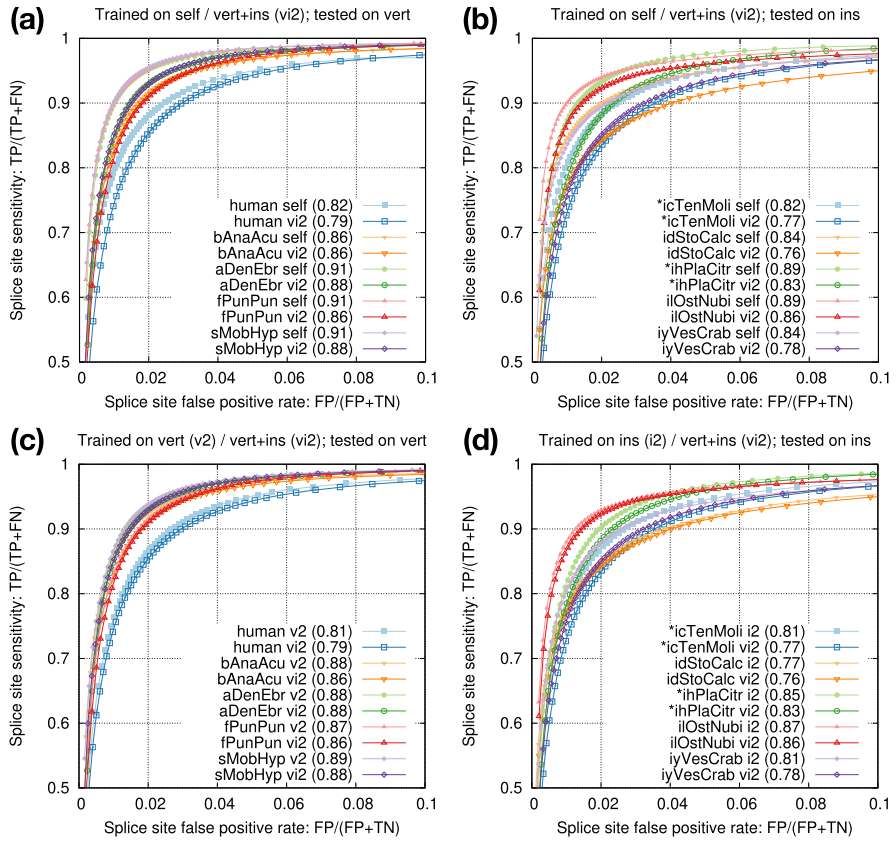

Fig. 5 Training from multiple species. **a** Accuracy of model vi2 on vertebrate genomes. vi2 is trained from six vertebrate and seven insect genomes. **b** Accuracy of vi2 on insect genomes. Odd chromosomes in starred species are used for training. Testing is applied to even chromosomes only. **c** Comparison between vi2 and a model trained from vertebrate genomes (v2). **d** Comparison between vi2 and a model trained from insect genomes (i2)

within introns. It is possibly why increasing window size from 202bp to 302bp has minor effect (Fig. 6b).

We further used the values at the last max-pooling layer to generate the UMAP of a random subset of training samples (Fig. 7). The UMAP separated the training samples into three large clusters which correspond to positive acceptor sites, negative sites and positive donor sites, respectively. Notably, some areas in the UMAP (the two red circles) only contain splice sites from mammals and birds but rarely from fish and insect. We extracted human splice sites in these areas and found the GC content of the 100bp intronic sequences following the donor sites reaches 64%, much higher than the genome average of ~40% in human. The presence of these areas might explain why fish and insect models do not work well for mammals (Fig. 4a) but mammalian models tend to work better for non-mammals [32].

To more directly investigate the lack of GC-rich introns in fish and insects, we extracted homologous intron pairs and compared their GC content (Fig. 8). We saw introns reaching 60% GC in human but their homologs in zebrafish are <40% in GC (Fig. 8a). Consistent with our earlier observation in UMAP, few zebrafish introns are GC-rich. Although the GC content is overall higher in amphibians, not many introns can reach 60% GC (Fig. 8b). Birds have high-GC introns like human (Fig. 8c). This is unrelated to warm- versus cold-

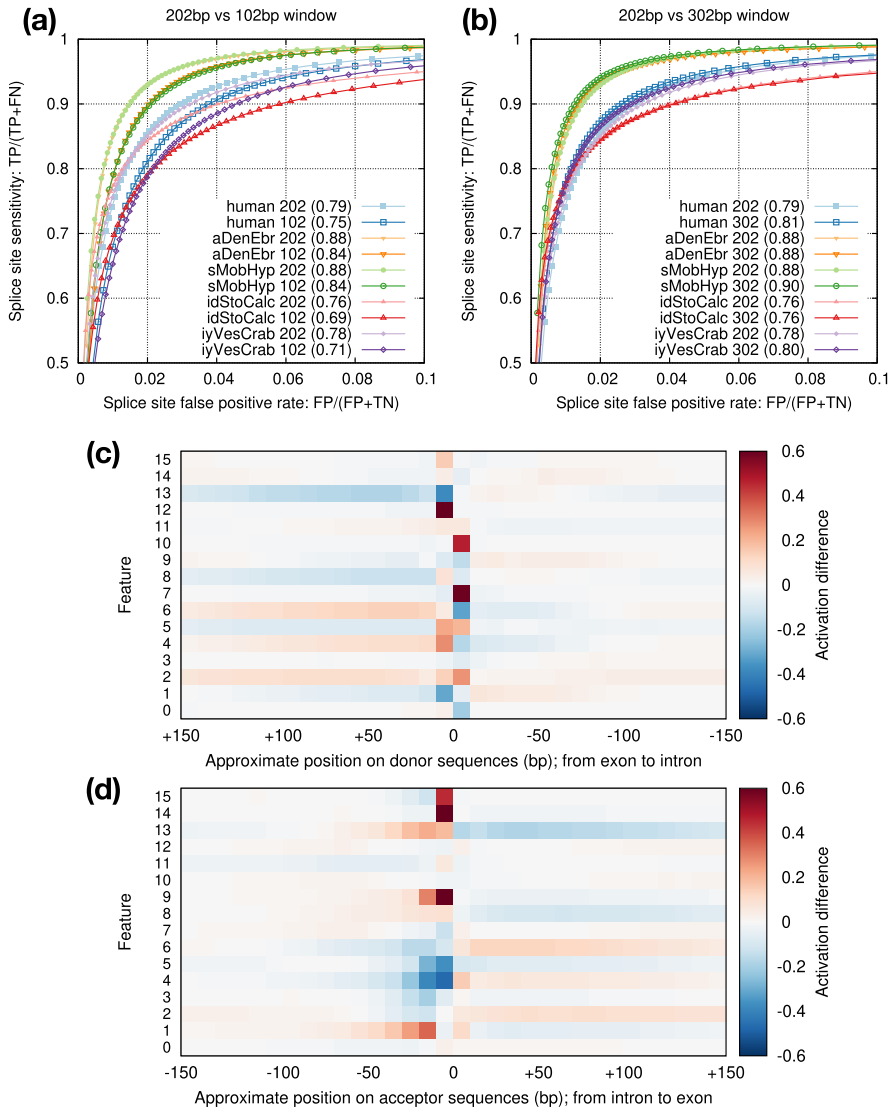

Fig. 6 Effect of window size. **a** Comparison between the default 202bp window size (±100bp around GT/AG) and 102bp. **b** Comparison between 202bp and 302bp window sizes. **c** Activation differences between positive donor sites and negative GT sites at the last max-pooling layer. Positive coordinates correspond to exonic bases. With 302bp window, this layer consists of 16 × 32 cells which are non-negative due to the ReLU function. Given a set of sequences, the activation rate of a cell is the frequency of the cell being positive (i.e. non-zero). The heat shows the rate difference between positive and negative sets. Dark red indicates more frequent activation among positive donor sites; dark blue is the opposite. **d** Activation difference for acceptor sites

blooded organisms per isochore theory [4] because reptiles also have GC-rich introns (Fig. 8d).

GC-rich introns are known to exist in human [40]. They may help stabilize RNA secondary structure [50] and may influence splice site recognition [2]. They are more likely to be retained than GC-poor introns [35], and they tend to be located in the nuclear center [13] which may be partly correlated with the higher CpG dentity in the nuclear center [38,44].

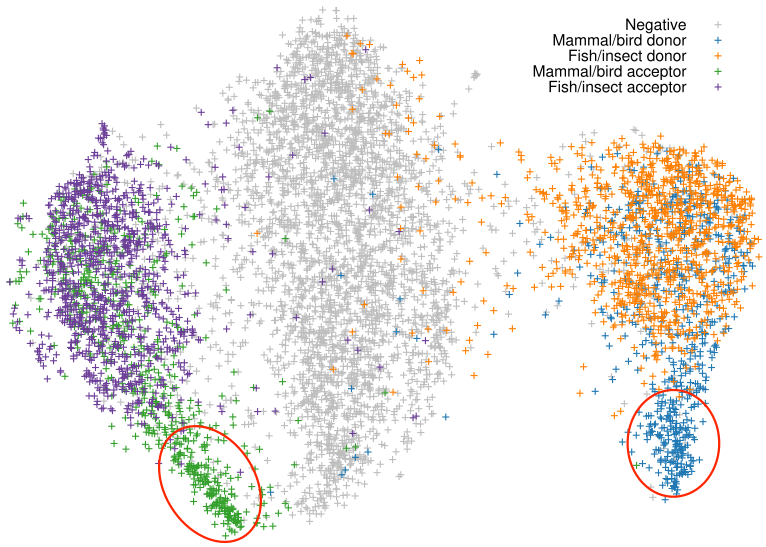

Fig. 7 UMAP of the last max-pooling layer. Training data was downsampled to 20k splice sites with 10k positive and 10k negative sites

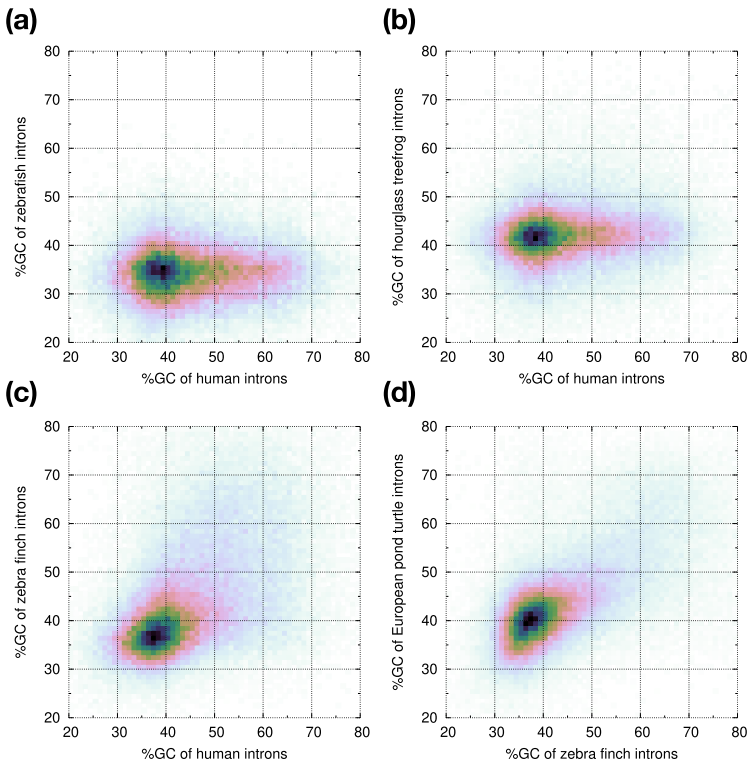

Fig. 8 GC content of homologous introns. Vertebrate protein sequences from Swiss-Prot are aligned to each genome with miniprot. Given two genomes, if a protein sequence is aligned with introns at the same position on the protein, the two aligned introns are considered homologous to each other. **a** Comparing percent GC content between homologous human and zebrafish introns. **b** Human versus hourglass tree frog (aDenEbr). **c** Human versus zebra finch (bTaeGut). **d** Zebra finch versus European pond turtle (rEmyOrb)

Improving spliced alignment

To evaluate the effect of splice models, we aligned zebrafish proteins to the human genome with miniprot (Table 2) and checked if an aligned junction is annotated in human. Unannotated junctions are likely errors in the protein-to-genome alignment. If we ask miniprot to only look for the minimal `GT..AG` signal, 14.01% of aligned junctions are not annotated in human. This percentage drops with improved splice models and reaches 4.37% when minisplice scores are considered during alignment. The similar trend is also observed when aligning mosquito proteins to the fruitfly genome. This confirms the critical role of splice models in protein-to-genome alignment.

We stratified the zebrafish proteins by their alignment identity (Fig. 9a) and inspected the alignment accuracy in each identity bin. The fraction of aligned junctions and the junction accuracy both drop with alignment identity (Fig. 9b, c). The minisplice model achieves higher accuracy across all bins, often halving the error rate in comparison to the default miniprot model.

We were evaluating junction accuracy. The exon accuracy was 9–11% lower for the zebrafish-to-human alignment. We manually inspected some exon-specific errors and classified them into three cases. First, an alignment did not reach the start of the protein sequence. The first junction in the alignment could still be correct but the beginning of the first exon was often an error. Second, 10.7% of longest zebrafish proteins did not start with methionine "M", the start amino acid. Third, zebrafish proteins starting with "M" may occasionally be mutated in human. This is an interesting but rare case. Overall, we intend to improve junction accuracy in this work but not the three cases above. Junction accuracy serves our goal better.

The minisplice model also greatly reduces the junction error rate for direct RNA-seq reads (Fig. 9d), though the reduction is less pronounced when the alignment identity reaches >98%. For this RNA-seq run with the latest the Nanopore RNA004 kit [52], 74.6% of reads are mapped with identity >98%. The overall junction error rate is marginally reduced from 1.4% to 1.0%. The advantage of advanced splice models will be more noticeable for old data of lower quality, regions of high diversity or cross-species cDNA-to-genome alignment.

As to performance, minisplice precomputed splice scores for the human genome in 1.4 h over 16 CPU threads. This step only needs to be done once per genome. Minimap2 and miniprot took ∼13 s upfront to load precomputed splice scores in the gzip format. The following alignment speed was barely affected as looking up splice scores is much faster than alignment.

Discussions

Minisplice is a command-line tool to learn splice signals with 1D-CNN and to predict the probability of splicing in the whole genome. While neural networks have long been applied to the modeling of splice sites [34], minisplice represents the first effort to directly integrate deep learning into alignment algorithms. The integration substantially improves the splice junction accuracy of minimap2 and miniprot, and is likely to help downstream analyses such as transcriptome reconstruction and gene annotation.

A marked limitation of minisplice is that it only models `GT..AG` splice sites. Although miniprot considers `GC..AG` and `AT..AC` signals and minimap2 has recently adopted

Table 2 Effect of splice site models on protein-to-genome alignment

	Zebrafish proteins to human genome					Mosquito proteins to fruitfly genome				
Aligner	**Miniprot**	**Miniprot**	**Miniprot**	**Miniprot**	**Spaln3**	**Miniprot**	**Miniprot**	**Miniprot**	**Miniprot**	**Spaln3**
Splice model	**GT-AG**	**Default**	**Extended**	**minisplice**	**Tetrapod**	**GT-AG**	**Default**	**Extended**	**Minisplice**	**InsectDm**
# predicted junc	168,030	164,094	160,935	164,860	144,544	30,279	28,780	27,307	28,722	24,538
# annotated junc	144,495	146,898	147,880	157,654	132,394	24,022	24,465	24,203	27,107	21,161
% unannotated	14.01	10.48	8.11	4.37	8.41	20.68	14.99	11.37	5.62	13.76
% Base Sn	60.09	60.12	60.04	60.85	51.71	57.00	56.92	56.76	57.28	44.53
% Base Sp	94.33	95.49	96.09	97.30	94.37	98.39	98.65	98.64	99.28	97.37

For the splice models, "GT-AG" only considers `GT..AG`, "default" considers `GTR..YAG`, "extended" additionally considers `G|GTR..YYYAG`, and "minisplice" uses the "vi2-7k" model on top of "extended". Miniprot was invoked with "`-I`" which sets the maximum intron length to $\min\{\max\{3.6\sqrt{G}, 10^4\}, 3 \times 10^5\}$, where G is the genome size. Spaln3 was invoked with "`-LS -yS -yB -yZ -yX2`" and with its own splice models. $\%\text{unannotated} = 1 - \#\text{annotated}/\#\text{predicted}$

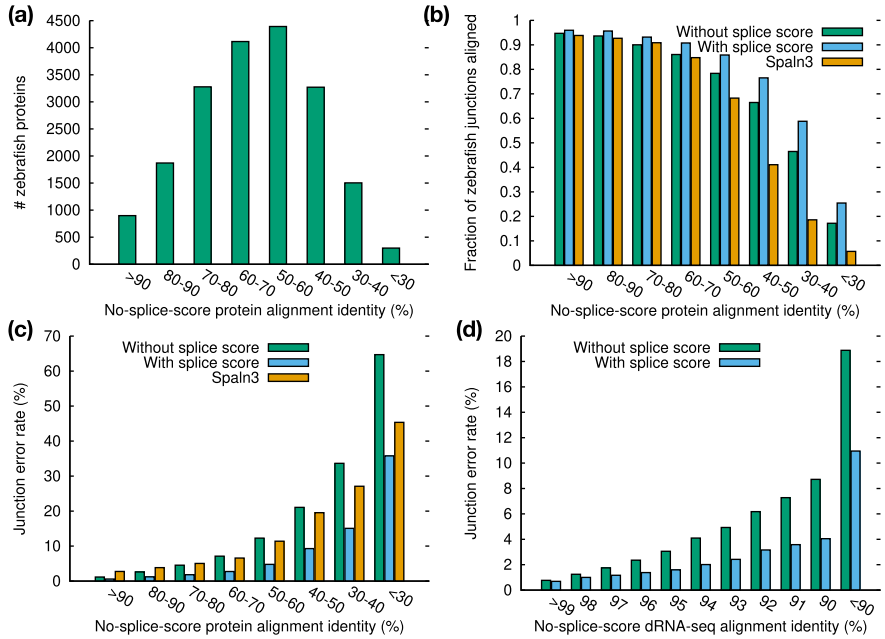

Fig. 9 Junction accuracy stratified by alignment identity. **a** Number of zebrafish proteins in each identity bin. Identity is estimated from alignment of zebrafish proteins against the human genome under the default miniprot splice model. **b** Number of aligned junctions divided by the number of coding junctions in the zebrafish annotation. **c** Junction error rate of protein-to-genome alignment. **d** Junction error rate of RNA-seq alignment. HG002 direct-RNA reads are aligned to the human genome with minimap2

the same model, they may still prefer `GT..AG` well scored by minisplice and misalign introns with non-`GT..AG` splicing. In addition, minisplice predicts splice scores from the reference genome. It is not aware of mutations in other samples that may alter splice signals. This bias towards the reference genome may lead to misalignment in rare cases but it is minor in comparison to annotation-guided alignment implemented in minimap2, which is biased towards both the reference genome and the reference annotation.

If we could predict splice sites to 100% accuracy, we would be close to solving the *ab initio* gene finding problem. However, the conserved splice signals only come from several bases around splice sites. Given hundreds of millions of `GT`/`AG` in a mammalian genome, we would make many wrong predictions just based on the several bases. The activation pattern at an internal layer suggests our model draws power from long-range composition of exonic sequences in addition to short-range signals around splice sites. The model may tend to classify a site as donor if the sequence on the left is similar to exonic sequences in composition. Nonetheless, even at 1% false positive rate, our model would predict 3.75 million ($= 3 \times 10^9 \times 2/16 \times 1\%$) false donor sites on average, an order of magnitude more than real donor sites. We must consider additional information for more accurate prediction. With $\geq$10kb windows and orders of magnitude more parameters, recent deep learning models such as SpliceAI, Pangolin and DeltaSplice will learn composition better. They may additionally see the promoter regions of many genes and species- or even tissue-specific regulatory elements. It is not clear how much their signals come from splice sites. At this point, methods combining HMM and deep learning [16,21] may be

more advantageous for coding sequences as they explicitly model gene structures and keep protein sequences in phase.

Acknowledgements
We thank Kuan-Hao Chao for pointing us to the Splam and OpenSpliceAI models prior to their publication.

Author contribution H.L. conceived the project. S.Y., N.H. and H.L. implemented the algorithms and analyzed the data. S.Y. and H.L. drafted the manuscript.

Funding
This work is supported by National Institute of Health grant R01HG010040 and R01HG014175 (to H.L.).

Data availability
No datasets were generated or analysed during the current study.

Declarations

Competing interests
The authors declare no competing interests.

Received: 16 June 2025 Accepted: 15 December 2025
Published online: 02 January 2026

References

1. Albaradei S, Magana-Mora A, Thafar M, Uludag M, Bajic VB, et al. Splice2Deep: an ensemble of deep convolutional neural networks for improved splice site prediction in genomic DNA. Gene. 2020;763S:100035.
2. Amit M, Donyo M, Hollander D, Goren A, Kim E, et al. Differential gc content between exons and introns establishes distinct strategies of splice-site recognition. Cell Rep. 2012;1:543–56.
3. Berakdar F, Wu TD, Zhu T, Samadi M, Vats P. DeepSAP: improved RNA-Seq alignment by integrating transcriptome guidance with transformer-based splice junction scoring. bioRxiv, 2025. p. 2025.04.23.650072.
4. Bernardi G, Olofsson B, Filipski J, Zerial M, Salinas J, Cuny G, et al. The mosaic genome of warm-blooded vertebrates. Science. 1985;228:953–8.
5. Birney E, Clamp M, Durbin R. GeneWise and Genomewise. Genome Res. 2004;14:988–95.
6. Brendel V, Kleffe J. Prediction of locally optimal splice sites in plant pre-mRNA with applications to gene identification in arabidopsis thaliana genomic DNA. Nucleic Acids Res. 1998;26:4748–57.
7. Brendel V, Xing L, Zhu W. Gene structure prediction from consensus spliced alignment of multiple ESTs matching the same genomic locus. Bioinformatics. 2004;20:1157–69.
8. Brixi G, Durrant MG, Ku J, Poli M, Brockman G, et al. Genome modeling and design across all domains of life with evo 2. bioRxiv, 2025. p. 2025.02.18.638918.
9. Burge C, Karlin S. Prediction of complete gene structures in human genomic dna. J Mol Biol. 1997;268:78–94.
10. Capitanchik C, Wilkins OG, Wagner N, Gagneur J, Ule J. From computational models of the splicing code to regulatory mechanisms and therapeutic implications. Nat Rev Genet. 2025;26:171–90.
11. Chao K-H, Mao A, Salzberg SL, Pertea M. Splam: a deep-learning-based splice site predictor that improves spliced alignments. Genome Biol. 2024;25:243.
12. Chen K, Zhou Y, Ding M, Wang Y, Ren Z, Yang Y. Self-supervised learning on millions of primary RNA sequences from 72 vertebrates improves sequence-based RNA splicing prediction. Brief Bioinform. 2024;25:bbae163.
13. Choquet K, Patop IL, Churchman LS. The regulation and function of post-transcriptional RNA splicing. Nat Rev Genet. 2025;26:378–94.
14. Dalla-Torre H, Gonzalez L, Mendoza-Revilla J, Lopez Carranza N, Grzywaczewski AH, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. Nat Methods. 2025;22:287–97.
15. Du X, Yao Y, Diao Y, Zhu H, Zhang Y, Li S. DeepSS: exploring splice site motif through convolutional neural network directly from DNA sequence. IEEE Access. 2018;6:32958–78.
16. Gabriel L, Becker F, Hoff KJ, Stanke M. Tiberius: end-to-end deep learning with an hmm for gene prediction. Bioinformatics. 2024;40:btae685.
17. Gotoh O. Direct mapping and alignment of protein sequences onto genomic sequence. Bioinformatics. 2008;24:2438–44.
18. Gotoh O. Spaln3: improvement in speed and accuracy of genome mapping and spliced alignment of protein query sequences. Bioinformatics. 2024;40:btae517.
19. Gremme G, Brendel V, Sparks ME, Kurtz S. Engineering a software tool for gene structure prediction in higher organisms. Inf Softw Technol. 2005;47(15):965–78.
20. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA. 1992;89:10915–9.
21. Holst F, Bolger A, Günther C, Maß J, Triesch S, et al. Helixer–de novo prediction of primary eukaryotic gene models combining deep learning and a hidden markov model. bioRxiv, 2023: p. 2023.02.06.527280.
22. Irimia M, Roy SW. Evolutionary convergence on highly-conserved 3′ intron structures in intron-poor eukaryotes and insights into the ancestral eukaryotic genome. PLoS Genet. 2008;4:e1000148.
23. Iwata H, Gotoh O. Comparative analysis of information contents relevant to recognition of introns in many species. BMC Genomics. 2011;12:45.
24. Iwata H, Gotoh O. Benchmarking spliced alignment programs including spaln2, an extended version of Spaln that incorporates additional species-specific features. Nucleic Acids Res. 2012;40:e161.

25. Jaganathan K, Kyriazopoulou Panagiotopoulou S, McRae JF, Darbandi SF, Knowles D, et al. Predicting splicing from primary sequence with deep learning. Cell. 2019;176:535-548.e24.
26. Kapustin Y, Souvorov A, Tatusova T, Lipman D. Splign: algorithms for computing spliced alignments with identification of paralogs. Biol Direct. 2008;3:20.
27. Kent WJ. BLAT-the BLAST-like alignment tool. Genome Res. 2002;12:656–64.
28. Keren H, Lev-Maor G, Ast G. Alternative splicing and evolution: diversification, exon definition and function. Nat Rev Genet. 2010;11:345–55.
29. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34:3094–100.
30. Li H. Identifying centromeric satellites with dna-brnn. Bioinformatics. 2019;35:4408–10.
31. Li H. Protein-to-genome alignment with miniprot. Bioinformatics. 2023;39:014.
32. McCue K, Burge CB. An interpretable model of pre-mRNA splicing for animal and plant genes. Sci Adv. 2024;10:eadn1547.
33. Nguyen E, Poli M, Faizi M, Thomas A, Birch-Sykes C, et al. HyenaDNA: long-range genomic sequence modeling at single nucleotide resolution. arXiv:2306.15794. 2023
34. Reese MG, Eeckman FH, Kulp D, Haussler D. Improved splice site detection in genie. J Comput Biol. 1997;4:311–23.
35. Sibley CR, Blazquez L, Ule J. Lessons from non-canonical splicing. Nat Rev Genet. 2016;17:407–21.
36. Slater GSC, Birney E. Automated generation of heuristics for biological sequence comparison. BMC Bioinform. 2005;6:31.
37. Staden R. Computer methods to locate signals in nucleic acid sequences. Nucleic Acids Res. 1984;12:505–19.
38. Tan L, Xing D, Chang C-H, Li H, Xie XS. Three-dimensional genome structures of single diploid human cells. Science. 2018;361:924–8.
39. Usuka J, Brendel V. Gene structure prediction by spliced alignment of genomic DNA with protein sequences: increased accuracy by differential splice site scoring. J Mol Biol. 2000;297:1075–85.
40. Wang D, Yu J. Both size and GC-content of minimal introns are selected in human populations. PLoS ONE. 2011;6:e17945.
41. Wu TD, Nacu S. Fast and SNP-tolerant detection of complex variants and splicing in short reads. Bioinformatics. 2010;26:873–81.
42. Wu TD, Watanabe CK. GMAP: a genomic mapping and alignment program for mRNA and EST sequences. Bioinformatics. 2005;21:1859–75.
43. Xia Y, Jin Z, Zhang C, Ouyang L, Dong Y, et al. TAGET: a toolkit for analyzing full-length transcripts from long-read sequencing. Nat Commun. 2023;14:5935.
44. Xie WJ, Meng L, Liu S, Zhang L, Cai X, Gao YQ. Structural modeling of chromatin integrates genome features and reveals chromosome folding principle. Sci Rep. 2017;7:2818.
45. Xu C, Bao S, Wang Y, Li W, Chen H, Shen Y, et al. Reference-informed prediction of alternative splicing and splicing-altering mutations from sequences. Genome Res. 2024;34:1052–65.
46. Yeo G, Burge CB. Maximum entropy modeling of short sequence motifs with applications to rna splicing signals. J Comput Biol. 2004;11:377–94.
47. You N, Liu C, Gu Y, Wang R, Jia H, et al. Splicetransformer predicts tissue-specific splicing linked to human diseases. Nat Commun. 2024;15:9129.
48. Zabardast A, Tamer EG, Son YA, Yılmaz A. An automated framework for evaluation of deep learning models for splice site predictions. Sci Rep. 2023;13:10221.
49. Zeng T, Li YI. Predicting RNA splicing from DNA sequence using Pangolin. Genome Biol. 2022;23:103.
50. Zhang J, Kuo CCJ, Chen L. GC content around splice sites affects splicing through pre-mRNA secondary structures. BMC Genomics. 2011;12:90.
51. Zhang Y, Liu X, MacLeod J, Liu J. Discerning novel splice junctions derived from rna-seq alignment: a deep learning approach. BMC Genomics. 2018;19:971.
52. Zheng Z, Yu X, Chen L, Lee Y-L, Xin C, et al. Clair3-RNA: a deep learning-based small variant caller for long-read RNA sequencing data. bioRxiv, 2025. p. 2024.11.17.624050.

Publisher's Note