TECHNICAL REPORT

Genome-wide comparison of predicted splicing effects between OpenSpliceAI and SpliceAI

By Kuan-Hao Chao

Sep 9, 2026 · 4 min read · Updated Sep 11, 2026

Abstract

OpenSpliceAI reproduces the SpliceAI architecture in PyTorch and supports retraining with different annotations and species. Here, we compare their predicted variant effects across the published collection of masked genic single-nucleotide variants. The frozen analysis contains 3,334,708,099 matched variant–gene annotations. Agreement differs by event: acceptor and donor loss scores have Pearson correlations of 0.900 and 0.906, compared with 0.681 and 0.597 for gains. At a cutoff of 0.5, OpenSpliceAI makes more loss calls and fewer gain calls, with the largest reduction in donor gain. Among jointly called gains, the models usually select the same position. On identical observations shared by two OpenSpliceAI training seeds and SpliceAI, gain-score differences exceed the observed seed difference more strongly than loss-score differences. These results describe agreement between a published SpliceAI ensemble and OpenSpliceAI individual retrained checkpoints; they do not establish relative biological accuracy.

Center for Computational Biology, Johns Hopkins University, Baltimore, MD, USA. Correspondence: kh.chao@jhu.edu. Scientific snapshot: 2026-09-09. Progress snapshot completed: 2026-09-10 20:30:22+00:00. Architecture and benchmarks.

Introduction

OpenSpliceAI implements the SpliceAI architecture in PyTorch and supports retraining with different annotations and species [2]. Similar splice-site prediction performance does not establish that two trained models assign the same effect to a sequence variant. Here, we compare OpenSpliceAI and SpliceAI across the published collection of masked genic single-nucleotide variants (SNVs), asking where their predicted effects agree and which types of effects differ most.

We distinguish four events: **acceptor gain (AG), acceptor loss (AL), donor gain (DG), and donor loss (DL)**. Gains summarize increases in predicted splice-site scores after substitution; losses summarize decreases. Each event score records the largest change within the search window and has an associated selected position.

Our comparisons address score magnitude, overlap of calls at a common cutoff, boundary proximity, and selected positions. We also compare two OpenSpliceAI training seeds on identical observations. SpliceAI is the comparator throughout. The collection contains no experimental splicing outcomes, so concordance identifies similarities and differences between predictions without determining which model is correct when they disagree.

Study design and inference progress

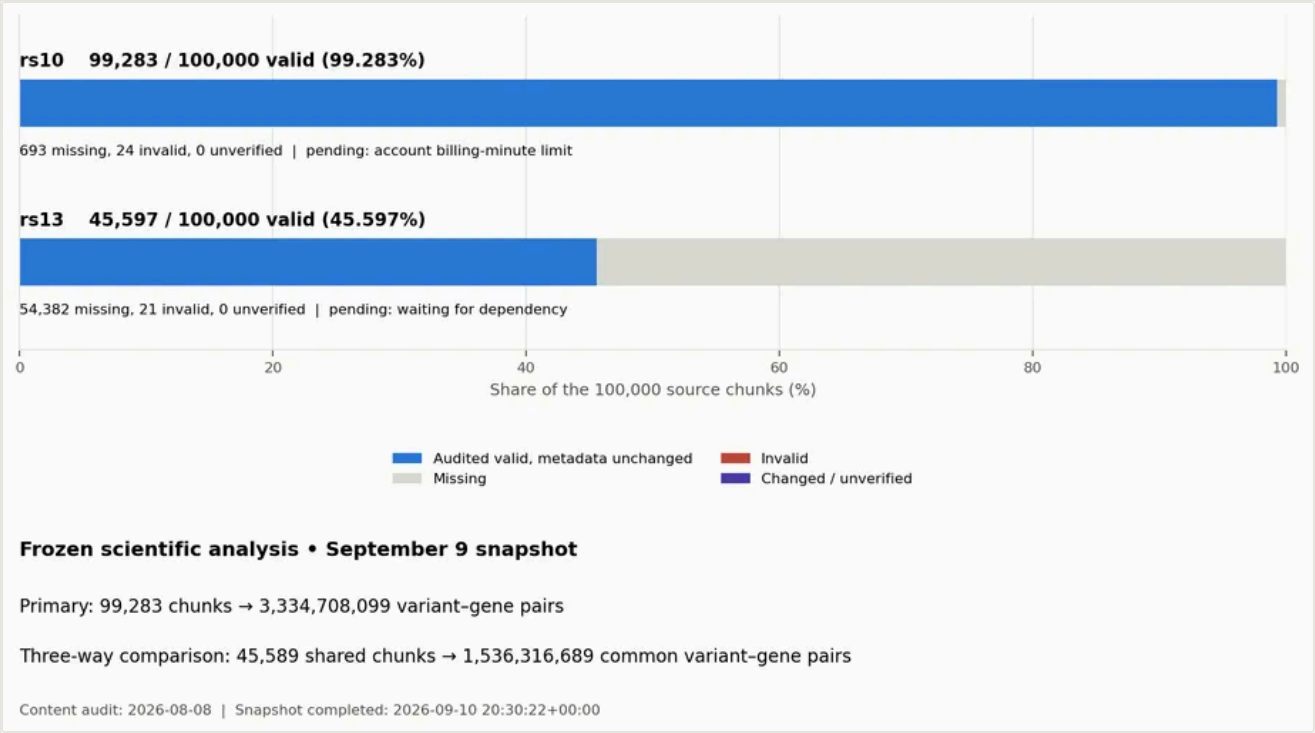

Figure 1. Inference progress and analyzed coverage. Current input/output size and modification-time checks are compared with the dated audit manifests. Unchanged metadata carries forward an audited state; it is not a new content audit. Missing, invalid and changed/unverified chunks remain separate. Job reasons are a dated scheduler observation, not a completion forecast. The lower panel shows the distinct primary and exact three-way analysis domains.

Loss scores agree more closely than gain scores

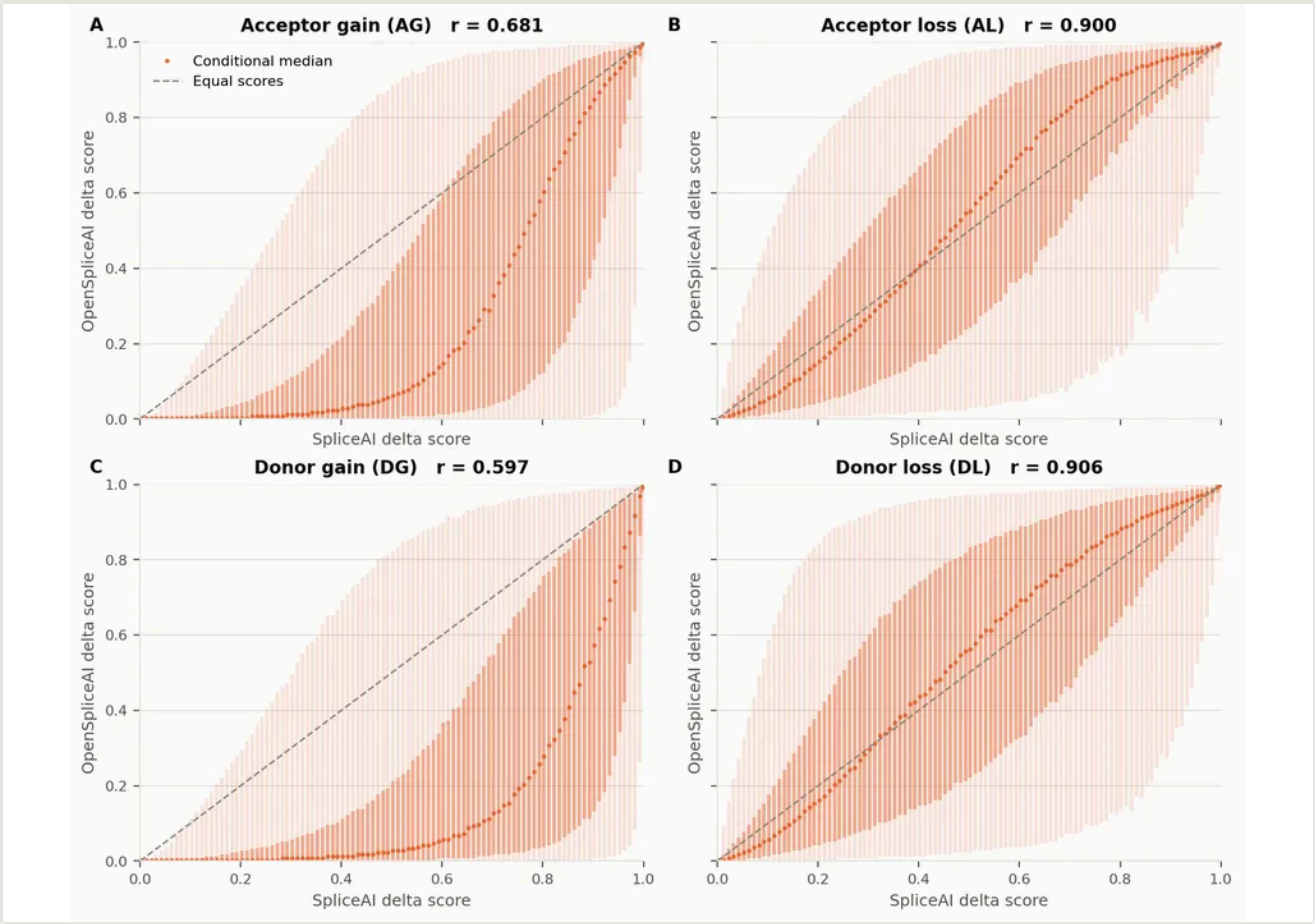

Figure 2. Loss scores agree more closely than gain scores. OpenSpliceAI scores conditional on SpliceAI scores for the primary paired variant–gene annotations. Points are medians; dark and light intervals contain the central 50% and 90%. The dashed diagonal denotes equal scores. Quantiles use stored 200-bin histograms (width 0.005); empty comparator bins remain gaps. Pearson correlations use the original scores and include shared zeros. The panels describe score agreement, not calibration against experimental outcomes.

Call overlap differs between the four events

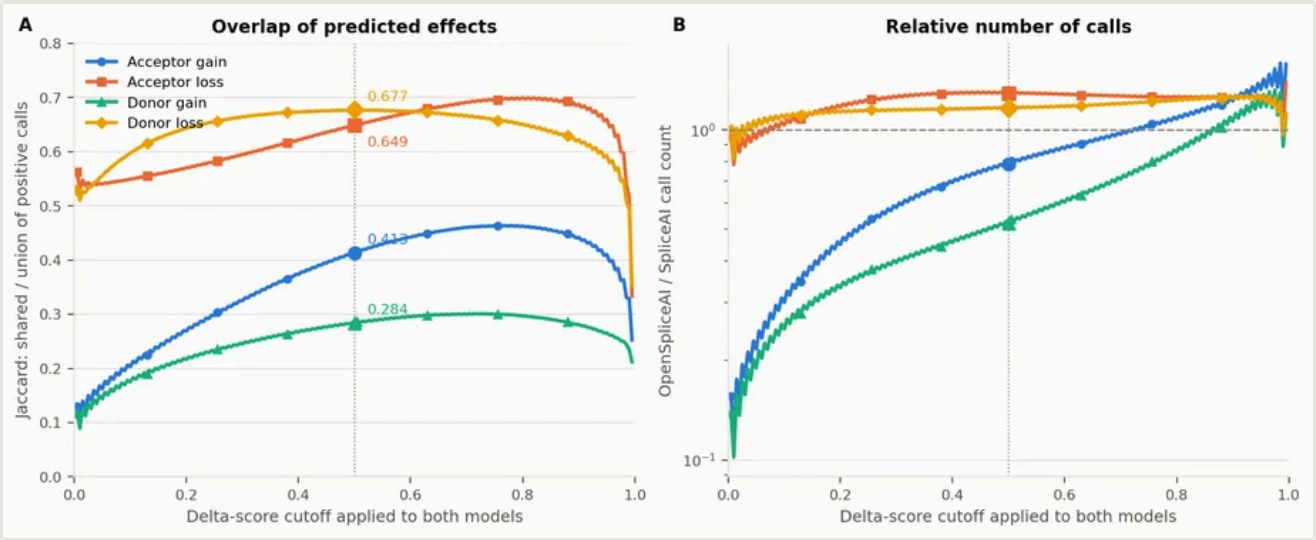

Figure 3. Gain calls overlap less, especially donor gain. A, shared positive calls divided by their union. B, OpenSpliceAI calls divided by SpliceAI calls; the dashed line denotes equal counts and the vertical scale is logarithmic. Colors and marker shapes distinguish the four events. Curves use approximate histogram cutoffs; larger symbols at 0.5 show exact counters. Shared negatives do not enter Jaccard. Fewer calls do not establish higher specificity.

Boundary proximity distinguishes local patterns

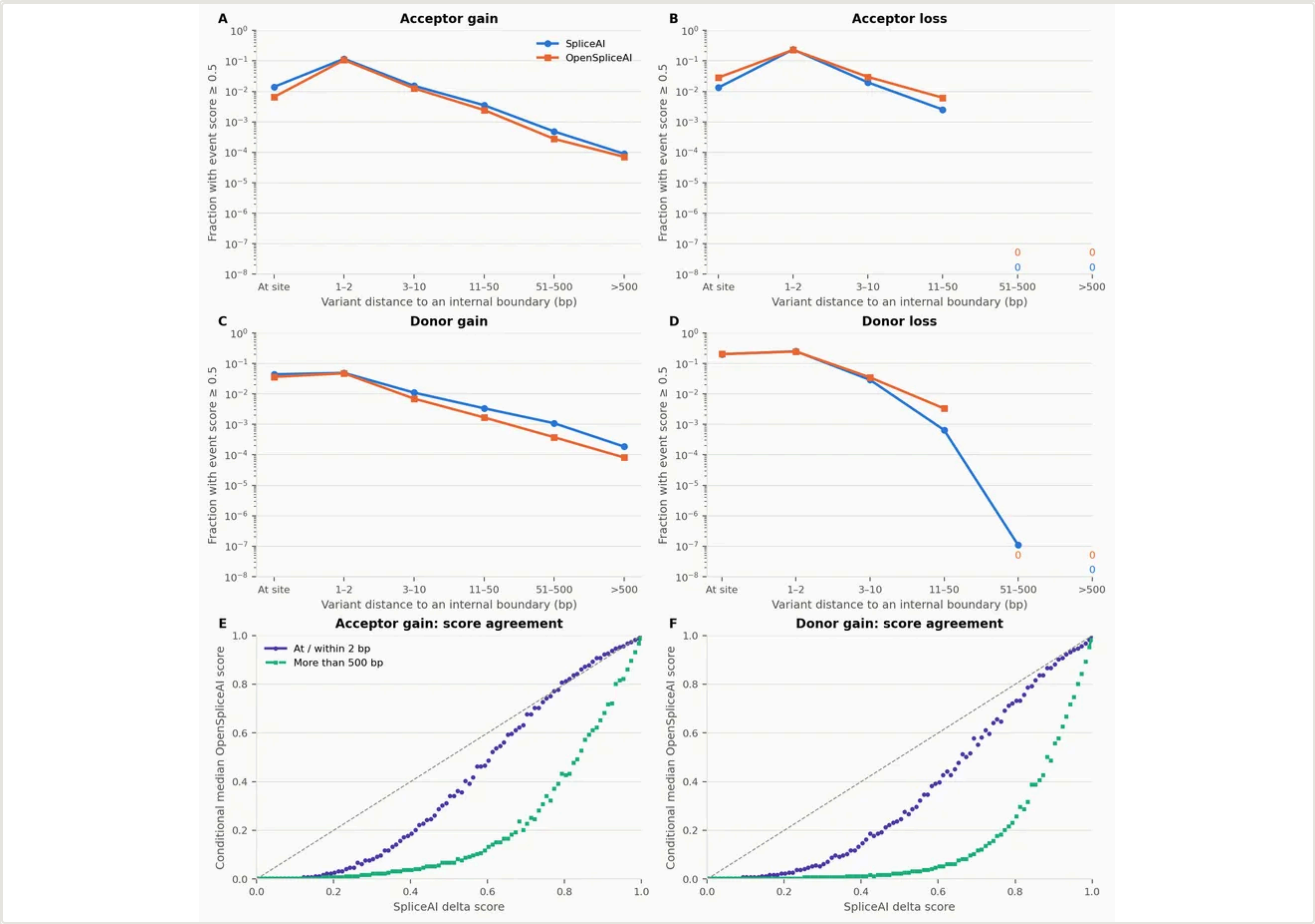

Figure 4. Annotation context distinguishes gain and loss differences. A–D, event score ≥0.5 divided by all paired variant–gene annotations in each distance group. Counts are pooled across acceptor, donor and ambiguous boundary types before division. Both models use the same group denominator. The rate axis is logarithmic; printed zeros indicate measured zero calls. E–F, histogram-derived conditional gain-score medians at/within 2 bp versus more than 500 bp from a pooled internal boundary. The strand-aware index excludes transcript ends. Variant distance differs from selected-event distance and does not classify variants as intronic or exonic.

Jointly called gains usually select the same position

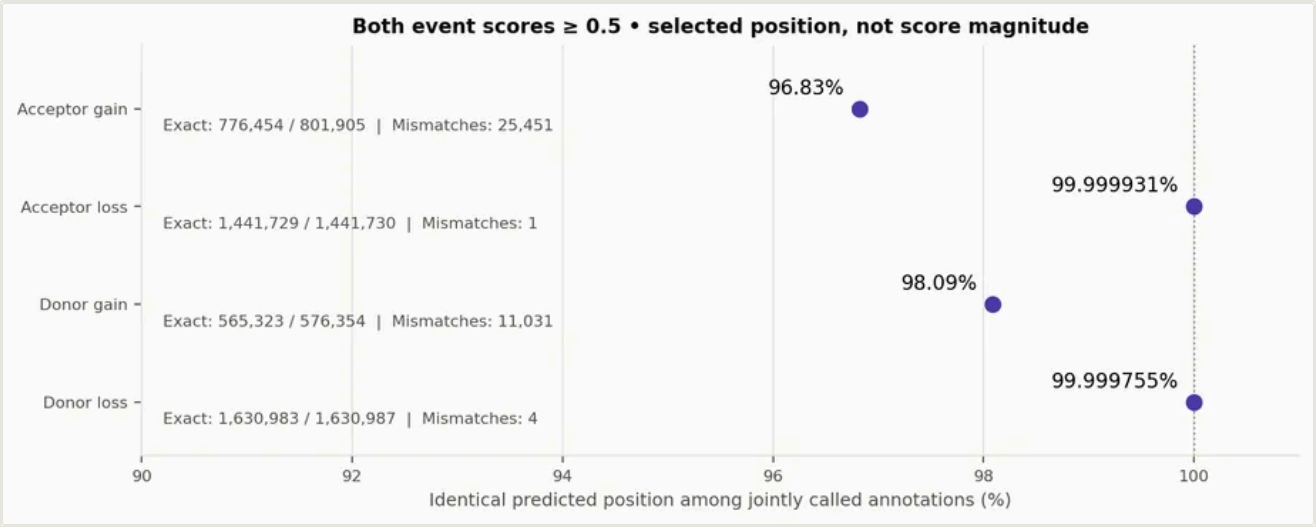

Figure 5. Jointly called gains usually select the same position. Both event scores must be ≥0.5. Labels give exact-position matches, eligible annotation counts and mismatches. AL has one mismatch and DL four; extra decimal places prevent rounding these results to perfect agreement. The axis spans 90–101%. Recorded masking restricts retained loss positions, but does not establish the cause of the exceptions. Calls unique to one model are excluded; shared positions do not validate splice-site use.

Gain disagreement exceeds the observed seed difference

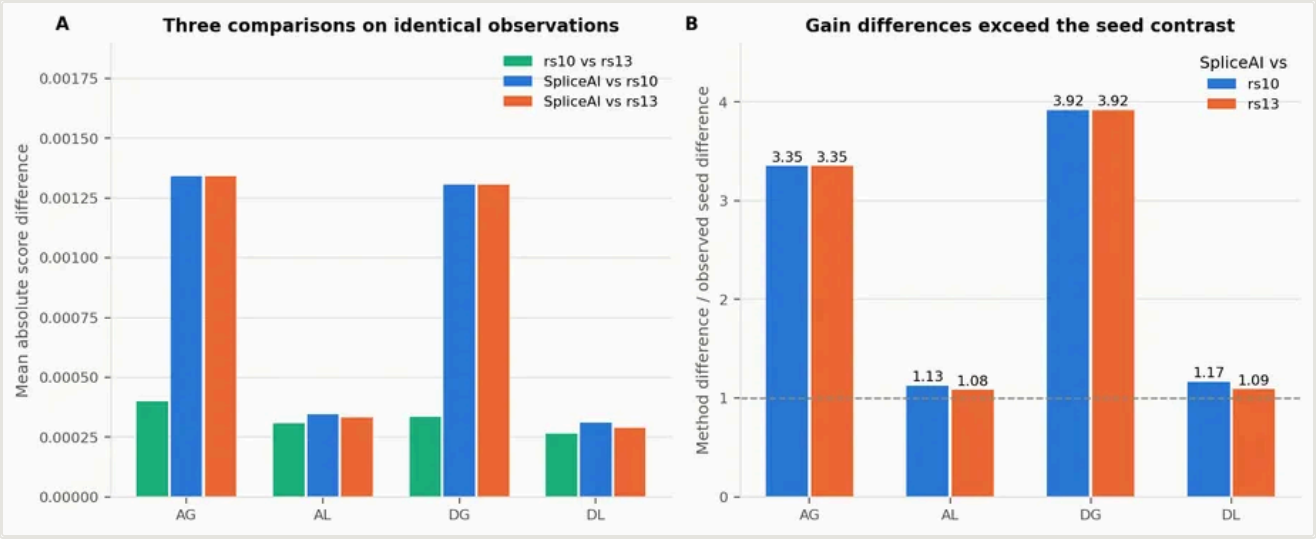

Figure 6. Gain differences exceed the observed seed contrast. A, mean absolute differences for rs10–rs13 and for each checkpoint versus SpliceAI. B, each method difference divided by the rs10–rs13 difference; one denotes equal mean differences. All contrasts use exactly the same variant–gene annotations. Two seeds provide one observed contrast, not a population estimate of training variability. Single OpenSpliceAI checkpoints and the published SpliceAI ensemble remain different comparison units.

References

1. Jaganathan K, Kyriazopoulou Panagiotopoulou S, McRae JF, et al. Predicting Splicing from Primary Sequence with Deep Learning. Cell, 2019. doi:10.1016/j.cell.2018.12.015
2. Chao K-H, Mao A, Liu A, Salzberg SL, Pertea M. OpenSpliceAI provides an efficient modular implementation of SpliceAI enabling easy retraining across nonhuman species. eLife 14:RP107454, 2025. doi:10.7554/eLife.107454.3