**Contents**

[Supplementary Figure S1 2](#_Toc533157166)

[Supplementary Figure S2 3](#_Toc533157167)

[Supplementary Figure S3 5](#_Toc533157168)

[Supplementary Table S1 6](#_Toc533157169)

[Supplementary Table S2 7](#_Toc533157170)

[Supplementary Note 8](#_Toc533157168)



**Supplementary Figure S1: Spliceogen Workflow**.

Spliceogen accepts VCF/BED inputs. The user-provided GTF annotation is used to extract the strand orientation of each variant. The GTF is also used to generate a genome-wide index of donor and acceptor motif intervals, allowing identification of variants that overlap splice sites. We developed scanning functionality which allows extraction of reference and alternative sequences for comparison. Four prediction algorithms are utilised (Branchpointer, GeneSplicer, MaxEntScan and ESRseq) that cover all major *cis* motifs that guide splicing. The scores from all these methods are merged for each variant, and variants that overlap splice sites are identified. Predictions are generated to identify potential donor/acceptor creating or disrupting variants, with the most likely candidates ranked and outputted in separate files. A full output file containing all scores and all predictions for all variants is also provided in an ANNOVAR format.



**Supplementary Figure S2: Spliceogen scanning functionality.**

A variant can occur within any position of a motif. Spliceogen scans reference and alternative sequences flanking a variant to identify potential motifs. This allows scoring by MaxEntScan that requires requiring a 9bp or 23bp input string aligned with the respective motifs a priori. All possible hexamers for reference and alternative sequences are also scanned and given a score based on ESRseq. Adjustments are necessary for indels, to ensure that equivalent positions are scanned between reference and alternative. An example shows the scanning adjustment made for a short deletion in extracting hexamers for ESRseq scoring. Similar adjustments are made for donor and acceptor motif scanning.

**Supplementary Figure S3: Decision tree for the identification of splice-altering variants.**

A recommended approach is outlined for the identification of disruptive variants based on their location, along with the direction and magnitude of the change in scores between reference and alternative alleles using all prediction algorithms.

**Supplementary Table S1: Speed improvement due to GeneSplicer multi-line adaptation**

For the original command line implementation of GeneSplicer, to assess a batch of variants it is necessary to create at least 1 file per variant, since only a single FASTA is read per file, and GeneSplicer only reads from file (not stdin). By adapting GeneSplicer to read multi-line FASTA inputs, we achieved a dramatic increase in speed, mainly through decreased input/output. Adapting GeneSplicer to read variant information from the FASTA header was another change we made to substantially improve its scalability in a variant analysis pipeline. However, only the GeneSplicer runtime was measured here, so the observed 50-fold improvement in speed was due only to the multi-line adaptation. The script we used for benchmarking can be found at: https://github.com/VCCRI/Spliceogen/blob/master/sources/gsComparison.sh

|  |  |  |  |
| --- | --- | --- | --- |
| **Input variants (thousands)** | **Time(sec) GS original** | **Time(sec) GS adapted** | **Time(original) / Time(adapted)** |
| 50 | 592 | 12 | 49 |
| 100 | 1150 | 24 | 48 |
| 150 | 1767 | 34 | 52 |
| 200 | 2607 | 47 | 55 |
| 250 | 2872 | 57 | 50 |

**Supplementary Table S2: Spliceogen benchmarking.**

Runtime and peak memory usage for several input file sizes up to 25 million variants.

|  |  |  |  |
| --- | --- | --- | --- |
| **Input variants****(millions)** | **Time (sec)** | **Millions/hr** | **Memory (MB)** |
| 4 | 6406 | 2.24 | 383 |
| 12 | 18912 | 2.28 | 382 |
| 25 | 38149 | 2.36 | 424 |

**Supplementary Note: Logistic Regression training data and model development**

**Data**

The positive dataset published in Shiraishi et al., 2018, was identified using the SAVnet method of pairing RNAseq-identified splicing disruptions with somatic variants located strictly within the disrupted or de novo splice site junctions, applied to matched whole-exome and transcriptome samples from The Cancer Genome Atlas. We used all variants that were annotated as donor/acceptor creating, excluding those that occurred within annotated splice sites. In total there were 1,156 donor-creating and 342 acceptor-creating variants in the positive set. For the negative set, we randomly sampled biallelic variants called across 2,500 samples from chr20 of 1000 Genomes Project (hg38 version) (Auton,A., 2015). We used approximately a 1:1 ratio of positive:negative scores. Because (as expected) there were more missing scores in the negative dataset, and more missing scores for donors compared to acceptors, the actual number of positive:negative variants was varied for each model to achieve this approximate 1:1 ratio.

**Feature Selection**

Spliceogen provided the reference and alternative scores for both GeneSplicer and MaxEntScan, for donor and acceptors. Score magnitude and score change (alternative minus reference) were used as independent variables for predicting donor/acceptor creating variants. This meant four features were included in the donor/acceptor models: MaxEntScan alternative score, MaxEntScan score change, GeneSplicer alternative score and GeneSplicer score change.

**Missing score imputation**

Missing donor/acceptor scores for MaxEntScan occur when there are no GT/AG dinucleotides within the Spliceogen scanning window. All such missing scores were assigned minus 20. GeneSplicer has many more missing scores since it only outputs scores above its threshold for identifying likely splice sites. When both reference and alternative scores were missing for GeneSplicer, 0 was assigned. When only one of either the reference or alternative score was missing, a stronger penalty of minus 3 was assigned. The values for missing scores for both MaxEntScan and GeneSplicer were chosen to follow the scale of the scores. The model was tested for different imputation values and the accuracy was stable.

**Model development**

We used the R package "caret" (Kuhn,M., 2008) with its "glm" method. The "repeatedcv" method was used to implement 10-fold cross validation with 5 repeats. We used Area Under the Curve values to evaluate model performance. To account for potential bias arising from the training:test data split, the reported AUC values (0.952 for donor, 0.914 for acceptor) were obtained by averaging over 20 random 80:20 splits of training and test data.

**References**

Auton,A. et al. (2015) A global reference for human genetic variation. Nature 526, 68-74.

Kuhn,M. (2008) Building Predictive Models in R Using the caret Package. Journal of Statistical Software, 28(5), 1-26.