IVAS : Identification of genetic Variants affecting Alternative Splicing

Seonggyun Han and Sangsoo Kim

April 15, 2025

Contents

1 Introduction

Alternative splicing controls relative expression ratios of mature mRNA isoforms from a single gene. Mapping studies of Splicing Quantitative Trait Loci (SQTL), a genetic variant affecting the alternative splicing, are important steps to understand gene regulations and protein activity [1]. We present an effective and user-friendly computational tool to detect SQTLs using transcript expression data from RNA-seq and genotype data, both measured on the same sample. As RNA sequencing (RNA-seq) provides insight into relatively precise measurments of expression level of transcript isoforms from a gene, it is a useful tool to analyze complicated biological phenomenon of RNA transcripts including the alternative splicing [2]. The mapping analysis uses two statistical models : Linear regression model [3] and/or Generalized linear mixed model [5].

2 The input data set

The next subsection introduces the input data. To run this tool, two experimental data sets (an expression data frame from RNA-seq and a genotype data frame) are required. Moreover, we also need a data frame for positions of SNP markers and GTF file for transcript models. As any other genome-wide analyses, it is recommended to use as many samples as possible, usually of population scale, in order to guarantee a statistically significant result.

2.1 The genotype data

The genotype data should be prepared as a simple matrix data. Each column represents an individual and its name should match that of the expression matrix described below (2.2)

	ind1	ind2	ind3	ind4
SNP1	AA	AA	AT	TT
SNP2	CG	CC	GG	CG
SNP3	TT	TT	AT	TT

2.2 The expression data

The expression matrix must comprise expression values of transcripts from RNA-seq. We may obtain them by using alignment tools such as cufflinks. Each column represents an individual and its name should match that of the genotype matrix described above (2.1)

	ind1	ind2	ind3	ind4
transcript1	10.5	15.4	6.7	12.4
transcript2	6.4	7.2	4.5	9.2
transcript3	15.4	14.5	13.2	17.8

2.3 The SNP marker position data

To search SNPs affecting alternative splicing, a data frame comprising genomic location of each SNP is required. It consists of following columns: SNP (SNP marker name), CHR(chromosome number), and locus(SNP position).

SNP	CHR	locus
SNP1	1	4964005
SNP2	1	23513047

2.4 The transcripts model data

We need a reference GTF (General Feature Format) file including information about gene structures such as the positions of exons, introns, and transcripts of genes. The GTF file must be `TxDb` object from the *GenomicFeatures* package [4].

3 The example dataset : data from Geuvadis RNA sequencing project of 1000 Genome samples

This example uses filtered data from an origin data generated by Geuvadis RNA sequencing project, available at `http://www.geuvadis.org/web/geuvadis/RNAseq-project` [6]. The example expression data includes transcripts of 11 randomly selected genes. The genotype data comprises SNPs in those genes.

4 Loading data

For this analysis, you need to load the *IVAS* package, SNP data, expression data, SNP position data, and `TxDb` object from GTF.
Loading *IVAS* package :

```
> library(IVAS)
```

Loading expression data :

```
> data(sampleexp)
```

Loading SNP data :

```
> data(samplesnp)
```

Loading SNP position data :

```
> data(samplesnplocus)
```

Loading `TxDb` object :

```
> sampleDB <- system.file("extdata","sampleDB", package="IVAS")
> sample.Txdb <- loadDb(sampleDB)
```

If you want to create the `TxDb` object from a GTF file, you need to use the `makeTxDbFromGFF` function in the *GenomicFeatures* package.

5 The `ASdb` obect

The `ASdb` object is a s4 type class object, and the object is used by the *IVAS* package to store the results from functions in this *IVAS* package. The functions of *IVAS* will save their results by adding a slot. Each slot contains a list object consisting of three elements named as "ES", "ASS", and "IR" for each alternatively splicing pattern type (i.e. ES, ASS, and IR means exon skipping, alternative splice site, and intron retention, respectively).

5.1 Searching alternatively spliced exons based on a reference transcript model.

The `Splicingfinder` function tabulates patterns of alternatively spliced exons. This needs the `TxDb` object from `makeTxDbFromGFF` by reading a reference GTF file for reference transcript models. The `Splicingfinder` function categorizes alternatively spliced exons into four types of AS patterns (i.e. exon skipping, alternative 3-prime splice site, alternative 5-prime splice site, and intron retention). The result will be saved in the "SplicingModel" slot of `ASdb`.
To use this function :

```
> ASdb <- Splicingfinder(GTFdb=sample.Txdb,calGene=NULL,Ncor=1,out.dir=NULL)

[1] "-------------------Processing : chr 2 -------------------"
[1] "-------------------Processing : chr 3 -------------------"
[1] "-------------------Processing : chr 6 -------------------"
[1] "-------------------Processing : chr 8 -------------------"
[1] "-------------------Processing : chr 9 -------------------"
[1] "-------------------Processing : chr 11 -------------------"
[1] "-------------------Processing : chr 17 -------------------"
[1] "-------------------Processing : chr 19 -------------------"

> ASdb
```

```
Splicing Models : ES = 182 Rows & ASS = 11 Rows & IR = 2 Rows
#ASdb object with SplicingModel

> head(slot(ASdb,"SplicingModel")$"ASS")

  Index  EnsID             Nchr Strand ShortEX
1 "ASS1" "ENSG00000186001" "3"  "+"    "197562545-197562609"
2 "ASS2" "ENSG00000183826" "6"  "-"    "38565686-38565833"
3 "ASS3" "ENSG00000183826" "6"  "-"    "38565686-38565833"
4 "ASS4" "ENSG00000172728" "8"  "-"    "33319006-33319245"
5 "ASS5" "ENSG00000172728" "8"  "-"    "33318930-33319243"
6 "ASS6" "ENSG00000166263" "17" "+"    "53076993-53077203"
  LongEX                ShortNeighborEX       LongNeighborEX
1 "197562545-197562693" "197566192-197566268" "197566192-197566268"
2 "38565686-38565897"   "38607576-38607924"   "38607576-38607700"
3 "38565686-38565897"   "38607576-38607924"   "38580610-38580809"
4 "33318890-33319243"   "33310734-33311028"   "33310734-33311028"
5 "33318890-33319243"   "33310734-33311028"   "33310734-33311028"
6 "53076987-53077203"   "53076706-53076812"   "53076706-53076812"
  Short_des             Long_des              ShortNeighbor_des
1 "197562545-197562609" "197562545-197562693" "197566192-197566268"
2 "38565686-38565833"   "38565686-38565897"   "38607576-38607924"
3 "38565686-38565833"   "38565686-38565897"   "38607576-38607924"
4 "33319006-33319245"   "33318890-33319243"   "33310734-33311028"
5 "33318930-33319243"   "33318890-33319243"   "33310734-33311028"
6 "53076993-53077203"   "53076987-53077203"   "53076706-53076812"
  LongNeighbor_des                      Types
1 "197566192-197566268"                 "A5SS"
2 "38580610-38580809,38607576-38607700" "A5SS"
3 "38580610-38580809,38607576-38607700" "A5SS"
4 "33310734-33311028"                   "A3SS"
5 "33310734-33311028"                   "A3SS"
6 "53076706-53076812"                   "A3SS"
```

You are able to define only a single gene if the single gene is inputted. The first column, named by "Index", is a generally used as an identifier and commonly used in other functions of *IVAS*.

5.2 Estimating expression ratio of AS exons with a data set including FPKM values of transcripts

The `RatioFromFPKM` function calculates expression ratio between transcripts with and without alternatively spliced exons. First, `RatioFromFPKM` divides the isoforms from a single gene into two groups: transcripts with and without an alternatively spliced exon. Then, the ratio of the group totals of transcript FPKM values is calculated. The `RatioFromFPKM` requires expression data set of transcript FPKM values and `ASdb` with the "SplicingModel" slot. The result will be saved in the "Ratio" slot of `ASdb`

```
> ASdb <- RatioFromFPKM(GTFdb=sample.Txdb,ASdb=ASdb,Total.expdata=sampleexp,
+     CalIndex="ASS7",Ncor=1,out.dir=NULL)
> ASdb

Splicing Models : ES = 182 Rows & ASS = 11 Rows & IR = 2 Rows
Ratio : ES = 0 Rows by 0 samples & ASS = 1 Rows by 78 samples & IR = 0 Rows by 0 samples
#ASdb object with SplicingModel & Ratio

> head(slot(ASdb,"Ratio")$"ASS")

  Index  EnsID             Nchr Strand ShortEX             LongEX
1 "ASS7" "ENSG00000170889" "19" "+"    "54704610-54704756" "54704740-54704829"
  ShortNeighborEX     LongNeighborEX      Short_TX
1 "54705028-54705149" "54705028-54705149" "ENST00000302907|ENST00000391751"
  Long_TX                           Types  NA06984
1 "ENST00000391752|ENST00000402367" "A5SS" "0.0370610175563808"
  NA06986              NA06989             NA06994
1 "0.0754673755080699" "0.431995041306961" "0.352248098956179"
  NA07037             NA07048           NA07051             NA07056
1 "0.615508066951179" "0.2535297934717" "0.396359920018477" "0.229019337579839"
  NA07347             NA07357             NA10847
1 "0.147021679772774" "0.294091318766693" "0.0835188716083212"
  NA10851             NA11829              NA11830
1 "0.030954840680335" "0.0174246902189581" "0.030532429762246"
  NA11831            NA11843             NA11892             NA11893
1 "0.15728432880497" "0.215269984759597" "0.136324142619792" "0.38436403201718"
  NA11894             NA11920              NA11930
1 "0.212453507751045" "0.0333217262293684" "0.0681235360867984"
  NA11931              NA11992             NA11993
1 "0.0248643687301508" "0.245183066925427" "0.276368360584032"
  NA11994             NA11995              NA12004
1 "0.114340505887804" "0.0688073456257966" "0.0218694795539099"
  NA12006             NA12043              NA12044
1 "0.719903020532971" "0.0253480818915533" "0.0691011133998759"
  NA12045             NA12058            NA12144
1 "0.269579049697507" "0.35466877311412" "0.495392792194793"
  NA12154              NA12155              NA12249
1 "0.0353058516847381" "0.0356549500182518" "0.332527122764556"
  NA12272             NA12273              NA12275
1 "0.547392066663861" "0.0461822327489977" "0.134086715517285"
  NA12282             NA12283              NA12286
1 "0.584161781407799" "0.0376982756006002" "0.242375101101388"
  NA12287             NA12340             NA12341
```

```
1 "0.298063167714008" "0.344188773640231" "0.0398630023057284"
  NA12342             NA12347             NA12348
1 "0.189423547621436" "0.304069583466665" "0.0175489426208136"
  NA12383              NA12399              NA12400
1 "0.0285158376928488" "0.0208852172856115" "0.162129766403219"
  NA12413             NA12489             NA12546
1 "0.272617060489197" "0.102893668902972" "0.0480414433339426"
  NA12716            NA12717             NA12718
1 "0.38349497995995" "0.180658216895047" "0.269490808164129"
  NA12749              NA12750             NA12751
1 "0.0967290881103072" "0.226219189907337" "0.024698616842959"
  NA12761             NA12763              NA12775
1 "0.167469444660537" "0.0288808382631836" "0.475715991162855"
  NA12777             NA12778               NA12812
1 "0.281683152810493" "0.00965356629010742" "0.0351200909069428"
  NA12814             NA12815             NA12827
1 "0.583521251701744" "0.500254757772165" "0.313010002548281"
  NA12829             NA12830              NA12842
1 "0.035631624411982" "0.0102342527143945" "0.684792301681123"
  NA12843             NA12872             NA12873
1 "0.251585485678078" "0.220474614626964" "0.353762604584194"
  NA12874              NA12889             NA12890
1 "0.0362672467194004" "0.222764959132477" "0.27775528737905"
```

In this example, we will estimates ratio in the "ASS7" index among splicing models in `ASdb`.

5.3 Finding SQTLs

Using "SplicingModel" and "Ratio" slots in `ASdb` from `Splicingfinder` and `RatioFromFPKM`, respectively, the `sQTLsFinder` function can identifies significant SNPs associated with alternative splicing rate (ratio). The result will be saved in the "sQTLs" slot of `ASdb`

```
> ASdb <- sQTLsFinder(ASdb=ASdb,Total.snpdata=samplesnp,
+     Total.snplocus=samplesnplocus,method="lm",Ncor=1)
> ASdb

Splicing Models : ES = 182 Rows & ASS = 11 Rows & IR = 2 Rows
Ratio : ES = 0 Rows by 0 samples & ASS = 1 Rows by 78 samples & IR = 0 Rows by 0 samples
sQTLs : ES = 0 Rows & ASS = 1 Rows & IR = 0 Rows
#ASdb object with SplicingModel & Ratio & sQTLs

> head(slot(ASdb,"sQTLs")$"ASS")

     SNP         Index  EnsID             Strand Nchr Types
[1,] "rs3810232" "ASS7" "ENSG00000170889" "+"    "19" "A5SS"
```

```
     ShortEX               LongEX                ShortNeighborEX
[1,] "54704610-54704756" "54704740-54704829" "54705028-54705149"
     LongNeighborEX      pByGeno                FdrByGeno              diff
[1,] "54705028-54705149" "3.98508717225349e-13" "3.98508717225349e-13" "diff"
     met
[1,] "lm"
```

In this example, we will run the function with the linear regression model. `sQTLsFinder` shows chromosome numbers during mapping analysis.

6 Identification of SQTLs using multiple cores

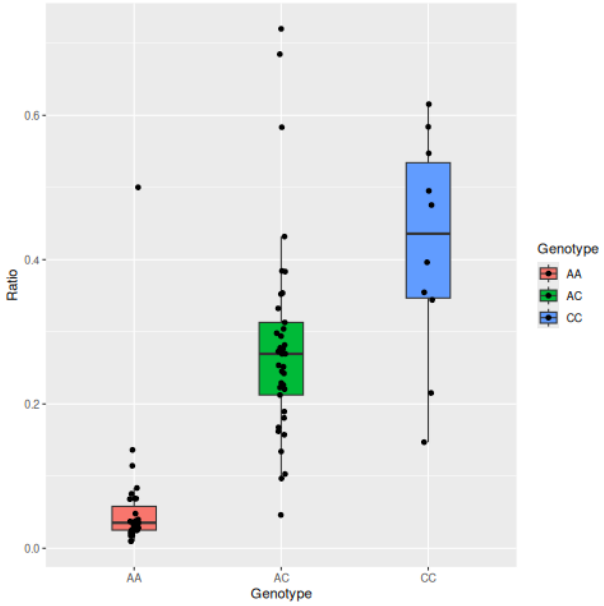
`Splicingfinder`, `RatioFromFPKM`, and `sQTLsFinder` functions provide to use multi-thread through `foreach` function. The last argument "Ncor" of the functions denotes the number of threads.

```
> ASdb <- Splicingfinder(GTFdb=sample.Txdb,calGene=NULL,Ncor=4)
> ASdb <- RatioFromFPKM(GTFdb=sample.Txdb,ASdb=ASdb,Total.expdata=sampleexp,Ncor=4)
> ASdb <- sQTLsFinder(ASdb=ASdb,Total.snpdata=samplesnp,
+     Total.snplocus=samplesnplocus,method="lm",Ncor = 4)
> ASdb
```

7 Visualizing the result

To visualize the results into boxplot, the *IVAS* package provides the `saveBplot` function. Using the data frame from the output of `sQTLsFinder` function, `saveBplot` can make the boxplot.

```
> saveBplot(ASdb=ASdb,Total.snpdata=samplesnp,Total.snplocus=samplesnplocus,
+     CalIndex="ASS7",out.dir="./result")
```



The output png files are saved in "result" folder.

8 Session Information

```
R version 4.5.0 RC (2025-04-04 r88126)
Platform: x86_64-pc-linux-gnu
Running under: Ubuntu 24.04.2 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.21-bioc/R/lib/libRblas.so
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0  LAPACK version 3.12.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_GB              LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: America/New_York
tzcode source: system (glibc)

attached base packages:
[1] stats4    stats     graphics  grDevices utils     datasets  methods
[8] base
```

```
other attached packages:
 [1] IVAS_2.28.0            ggplot2_3.5.2          GenomicFeatures_1.60.0
 [4] AnnotationDbi_1.70.0   Biobase_2.68.0         GenomicRanges_1.60.0
 [7] GenomeInfoDb_1.44.0    IRanges_2.42.0         S4Vectors_0.46.0
[10] BiocGenerics_0.54.0    generics_0.1.3

loaded via a namespace (and not attached):
 [1] tidyselect_1.2.1            ggfortify_0.4.17
 [3] farver_2.1.2                dplyr_1.1.4
 [5] blob_1.2.4                  Biostrings_2.76.0
 [7] bitops_1.0-9                fastmap_1.2.0
 [9] RCurl_1.98-1.17             GenomicAlignments_1.44.0
[11] XML_3.99-0.18               digest_0.6.37
[13] lifecycle_1.0.4             KEGGREST_1.48.0
[15] RSQLite_2.3.9               magrittr_2.0.3
[17] compiler_4.5.0              rlang_1.1.6
[19] tools_4.5.0                 yaml_2.3.10
[21] rtracklayer_1.68.0          knitr_1.50
[23] labeling_0.4.3              S4Arrays_1.8.0
[25] bit_4.6.0                   curl_6.2.2
[27] DelayedArray_0.34.0         abind_1.4-8
[29] BiocParallel_1.42.0         withr_3.0.2
[31] purrr_1.0.4                 grid_4.5.0
[33] colorspace_2.1-1            MASS_7.3-65
[35] scales_1.3.0                iterators_1.0.14
[37] SummarizedExperiment_1.38.0 cli_3.6.4
[39] rmarkdown_2.29              crayon_1.5.3
[41] reformulas_0.4.0            httr_1.4.7
[43] rjson_0.2.23                minqa_1.2.8
[45] DBI_1.2.3                   cachem_1.1.0
[47] stringr_1.5.1               splines_4.5.0
[49] parallel_4.5.0              BiocManager_1.30.25
[51] XVector_0.48.0              restfulr_0.0.15
[53] matrixStats_1.5.0           vctrs_0.6.5
[55] boot_1.3-31                 Matrix_1.7-3
[57] jsonlite_2.0.0              bit64_4.6.0-1
[59] foreach_1.5.2               tidyr_1.3.1
[61] glue_1.8.0                  nloptr_2.2.1
[63] codetools_0.2-20            stringi_1.8.7
[65] gtable_0.3.6                BiocIO_1.18.0
[67] UCSC.utils_1.4.0            lme4_1.1-37
[69] munsell_0.5.1               tibble_3.2.1
[71] pillar_1.10.2               htmltools_0.5.8.1
[73] GenomeInfoDbData_1.2.14     R6_2.6.1
```

```
[75] Rdpack_2.6.4                doParallel_1.0.17
[77] evaluate_1.0.3              lattice_0.22-7
[79] rbibutils_2.3               png_0.1-8
[81] Rsamtools_2.24.0            memoise_2.0.1
[83] BiocStyle_2.36.0            Rcpp_1.0.14
[85] nlme_3.1-168                gridExtra_2.3
[87] SparseArray_1.8.0           xfun_0.52
[89] MatrixGenerics_1.20.0       pkgconfig_2.0.3
```

References

[1] Keyan Zhao, Zhi-xiang Lu, Juw Won Park, Qing Zhou, Yi Xing. 2013. GLiMMPS: robust statistical model for regulatory variation of alternative splicing using RNA-seq data. Genome Biol 14, R74.

[2] Joseph K. Pickrell, John C. Marioni, Athma A. Pai, Jacob F. Degner, Barbara E. Engelhardt, Everlyne Nkadori, Jean-Baptiste Veyrieras, Matthew Stephens, Yoav Gilad, Jonathan K. Pritchard. 2010. Understanding mechanisms underlying human gene expression variation with RNA sequencing. Nature 464, 768-722.

[3] N.E. Breslow and D.G. Clayton. 1993. Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association 88 421: 9-25.

[4] Michael Lawrence, et al. 2013. Software for Computing and Annotating Genomic Ranges. PLoS Comput Biol. 9(8): e1003118.

[5] Chambers, J. M. 1992. Linear models. Chapter 4 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth, and Brooks Cole.

[6] Tuuli Lappalainen, et al. 2013. Transcriptome and genome sequencing uncovers functional variation in humans. Nature 501, 506-511.