Identification of alternative splicing from transcript sequences without a reference genome – the AStrap package

2018-03-06

1 Overview

The package *AStrap* implements a *de novo* approach to detect alternative splicing (AS) from transcript sequences without a reference genome, including identification of AS events by extensive pair-wise alignments of transcript sequences from SMRT sequencing data and prediction of AS types by a machine-learning model integrating more than 500 assembled features. AS events of four types including intron retention (IR), exon skipping (ES), alternative donor sites (AltD), and alternative acceptor sites (AltA) (Wang, et al., 2008) were considered. *AStrap* consists of four main stages: data preprocessing, feature construction, classification model building, identification of AS events and prediction of AS types. This vignette explains the use of the package. *AStrap* could be a valuable addition to the community for the study of AS in non-model organisms with limited genetic resources.

2 Preparations

AStrap needs at least three types of files: the transcriptome sequence file (FASTA); isoform cluster file (TEXT) generated by *CD-HIT* (Fu, et al., 2012); aligned sequence file (GFF3) generated by *GMAP* or other sequence alignment tools. Optionally, if users want to train a specific classification model on their own data sets, they should provide the alternative splicing database in TEXT format and the corresponding sequence information in FASTA, or using the R package *BSgenome* (Herve, 2014) to load the genome sequence.

2.1 File of transcript sequences

Transcript sequence file (FASTA format) stores all the full-length or non-full-length transcript sequences for AS detection, which is used to construct the sequence features. To demonstrate the use of *AStrap*, we adopted the SMRT sequencing data from Amborella (Liu, et al., 2017). They deposited the raw sequencing reads with NCBI BioProject database under the

accession number PRJNA374048. For getting high-quality full-length transcripts, users can adopt the *PacBio's SMRT Analysis* pipeline to process the raw ISO-seq data (Gordon, et al., 2015).
Here is an example data file:

```
>inDir <- system.file("extdata",package = "AStrap")
>ASfiles <- list.files(inDir,pattern = "*.fasta$",full.names = TRU
E)
>basename(ASfiles)
[1] "example_TRsequence.fasta"
```

2.2 File of isoform clusters

It is necessary to provide a text file of list of clusters for *AStrap*. To determine isoform clusters, users can use *CD-HIT-EST* that gathers the transcripts into clusters at a user-defined similarity threshold (Fu, et al., 2012) (Example comand line: `cd-hit-est -i input -o output -r 0 -c 0.80 -n 5 -M 1600 -T 16 -d 0`). The output is a text file of list of clusters as *AStrap*'s input. Based on the above transcripts, the output .cluster file by *CD-HIT-EST* looks like

```
>Cluster 0
0   1054nt, >AMTR202... at +/99.34%
1   983nt, >AMTR1121... at +/100.00%
2   1027nt, >AMTR1812... at +/84.42%
3   1706nt, >AMTR4451... at +/99.53%
4   1939nt, >AMTR4661... at +/99.54%
7   4102nt, >AMTR9147... at +/98.17%
8   4451nt, >AMTR10153... *
>Cluster 1
0   946nt, >AMTR1226... at +/99.79%
1   1076nt, >AMTR1487... at +/99.91%
2   1844nt, >AMTR2164... at +/93.82%
```

If you are unfamiliar with how to use *CD-HIT-EST*, following the explanations given on the CD-HIT web page: http://weizhongli-lab.org/cd-hit/.
Here is an example data file:

```
inDir <- system.file("extdata",package = "AStrap")
ASfiles <- list.files(inDir,pattern = "*.clstr$",full.names = TRU
E)
basename(ASfiles)
[1] "example_cdhitest.clstr"
```

2.3 File of alignments

Aligned sequence file (GFF3 format) is used to identify the similarity of isoforms of the same cluster in *AStrap*, which is generated by *GMAP* (Wu and

Watanabe, 2005) (Example comand line: `gmap -D dir -d index -t 10 --min-intronlength 9 -z sense_force --min-trimmed-coverage 0.7 --min-identity 0.7 -f 2 input > output.gff3`) or other sequence alignment tools. Note that it is a transcript-to-transcript alignment rather than transcript-to-genome. Please refer to https://genome.ucsc.edu/FAQ/FAQformat#format3 to see the GFF3 format. Here is an example data file:

```
>inDir <- system.file("extdata",package = "AStrap")
>ASfiles <- list.files(inDir,pattern = "*.gff3$",
                       full.names = TRUE)
>basename(ASfiles)
[1] "example_gmap.gff3"
```

2.4 Alternative splicing database

Two classification models trained on collected AS data from rice and human were integrated in *AStrap*, which could be directly applied for distinguishing among AS types for other species. AS data in rice were retrieved from the previous study (Chamala, et al., 2015), and in human were identified form the GRCh38 genome annotation using *ASTALAVISTA* (Foissac and Sammeth, 2007). Meanwhile, users can also train a specific classification model on their own data sets, but it is necessary to provide the AS database in TEXT format and the corresponding sequence information in FASTA or BSgenome format. The file containing AS events needs to contain the locus of spliced sites, including the coordinates (the corresponding column names are *coord1, coord2, coord3*, *coor4*), the strand of AS event (the column name is *strand*), the chromosome of AS event (the column name is *chr*), and the label of AS types (the column name is *type*). The coordinate definition is the same as the previous study (Chamala, et al., 2015). **Figure 1** illustrates the meaning of the coordinates. The corresponding content is shown in **Table 1**.

```
>inDir <- system.file("extdata",package = "AStrap")
>ASfiles <- list.files(inDir,pattern = "*AS.txt$",full.names = TRUE)
>basename(ASfiles)
##[1] "sample_humanAS.txt" "sample_riceAS.txt"
```

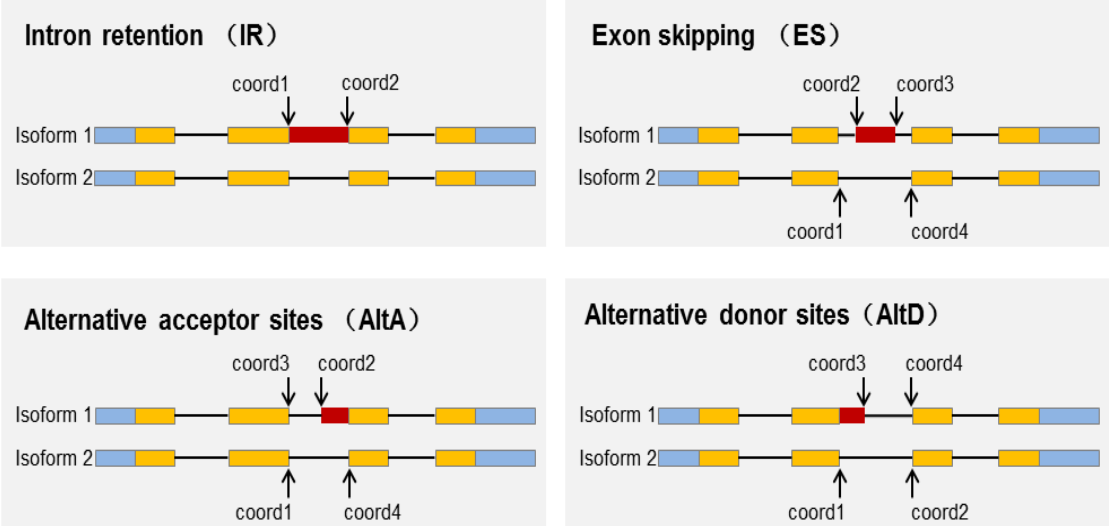

Figure 1. **Schematic diagram of four AS types with genomic coordinates**. The block in red denotes the AS region.

Table 1. **List of AS events.**

chr	coord1	coord2	coord3	coord4	strand	type
chr14	75004900	75005866	75004900	75005897	+	AltA
chr3	47097955	47088249	47097955	47088247	-	AltA
chr2	199948337	199943834	199948318	199943834	-	AltD
chr5	119634252	119635028	119634734	119635028	+	AltD
chr11	46617890	46618260	46618375	46621042	+	ES
chr17	17893661	17884768	17884634	17882863	-	ES
chr1	112698789	112699685	NA	NA	+	IR
chr17	44321434	44321231	NA	NA	-	IR

3 Standard analysis work-flow

3.1 Data loading

To illustrate the *AStrap* work flow, the SMRT sequencing data from *Amborella* is taken as an example (Liu, et al., 2017). Here for demonstration, we only use part of the data. First, in order to extract sequences around splice sites for the feature construction, users need to provide transcript sequences. We can use *readDNAStringSet* or *readRNAStringSet* to read FASTA file in an XStraingSet.

```
##Loading transcript sequences
>trSequence.path <- system.file("extdata",
                                "example_TRsequence.fasta",
                                 package = "AStrap")
> trSequence <-  readDNAStringSet(trSequence.path,
                                    format = "fasta")
```

```
> head(trSequence)
  A DNAStringSet instance of length 6
    width           seq                                      names
[1]   963 AACTCAACAGTTAAACCTAAAT...GTTATATAGGTCCATATTTTG AMTR2
[2]   983 TCTCTATCTGCGTGCGCCCTCT...AAATGTGGTTGATTAAAGCTC AMTR3
[3]  1028 GTGAAAAAAATGTGATCCACAT...TCTGTCACTGACTCACTTTTC AMTR4
[4]  1054 GAAGGGGCCAGTATCGGAGGAC...GTGGGAATATATTTTGCTACT AMTR5
[5]   944 TTCAACTTAGGGTTTTCATTTT...AGGTGGAGCTCTCTATTGTGT AMTR6
[6]  1014 ATGCATCCGGTGCCGAGTCTGA...TAAGGAAATAAAGGCCTTAAC AMTR8
> class(trSequence)
[1] "DNAStringSet"
attr(,"package")
[1] "Biostrings"
```

Next, use the *readCDHIT* to get a cluster table arranged by the column *ClusterID*. This function returns a data frame with four columns: *ClusterID, seqID, seqLen, seqNum*. *ClusterID* is the name of the cluster; *seqID* is the name of the transcript; *seqLen* is the length of the transcript; *seqNum* is the number of isoforms contained in the cluster.

```
##Loading the file of a list of clusters generated by CD-HIT-EST
> cdhit.path <- system.file("extdata","example_cdhitest.clstr",
                            package = "AStrap")
> raw.cluster <- readCDHIT(cdhit.path)
> head(raw.cluster)
  ClusterID   seqID seqLen seqNum
1         1  AMTR876   1386      1
2         2 AMTR1338    964      3
3         2 AMTR1349   1290      3
4         2 AMTR1399   1049      3
5         3 AMTR1956   1106      3
6         3 AMTR8587   3017      3
```

Third, use the *readGMAP* to load pairwise sequence alignments. This function is used to remove redundant or invalid sequence alignments and returns non-redundant pairwise alignments of isoforms of the same cluster. In addition, users can choose whether to adjust the clustering result by the parameter *recluster* of this function. When *recluster* is *TRUE* (the default), isoforms from single-isoform clusters will be reassigned to the corresponding clusters if the single isoform is similar to a transcript of other clusters. By default, both of the arguments *recluster.coverage* and *recluster.identity* are set to 0.7.

```
##Loading the alignment file in GFF3 format generated by GMAP
>gmap.path <- system.file("extdata","example_gmap.gff3",
                          package = "AStrap")
>cluster.align <- readGMAP(gmap.path,raw.cluster,
                          recluster = TRUE,
                          recluster.identity = 0.7,
```

```
                                recluster.coverage = 0.7)
#Pairwise alignment of isoforms in the same cluster
> alignment <- cluster.align$alignment
#Adujust  clusters
rew.cluster <- cluster.align$cluster
> head(alignment[,1:4])
       Qid      Sid Coverage identity
1 AMTR10014 AMTR8587     98.1     99.8
2  AMTR1009 AMTR1388     95.1    100.0
3  AMTR1012 AMTR1365     94.1     99.9
4  AMTR1018 AMTR1307     96.9     99.6
5  AMTR1020   AMTR47    100.0    100.0
6  AMTR1020  AMTR125     99.5     99.8
```

In *AStrap*, pairwise alignments of isoforms of the same cluster can be visualized by the function *plotCluster* and *plotAlign*, where the identity of a pairwise alignment is denoted as a line with different thickness and colors. The higher the identity is, the thicker the line will be. Red line denotes the identity of 1 (**Figure 2**).

```
##Plotting a network graph
>gg1 <- plotCluster(raw.cluster,cluster.id=c("7"))
>plot(gg1)
>gg2 <- plotAlign(alignment,cluster.id=c("7"))
>plot(gg2)
```

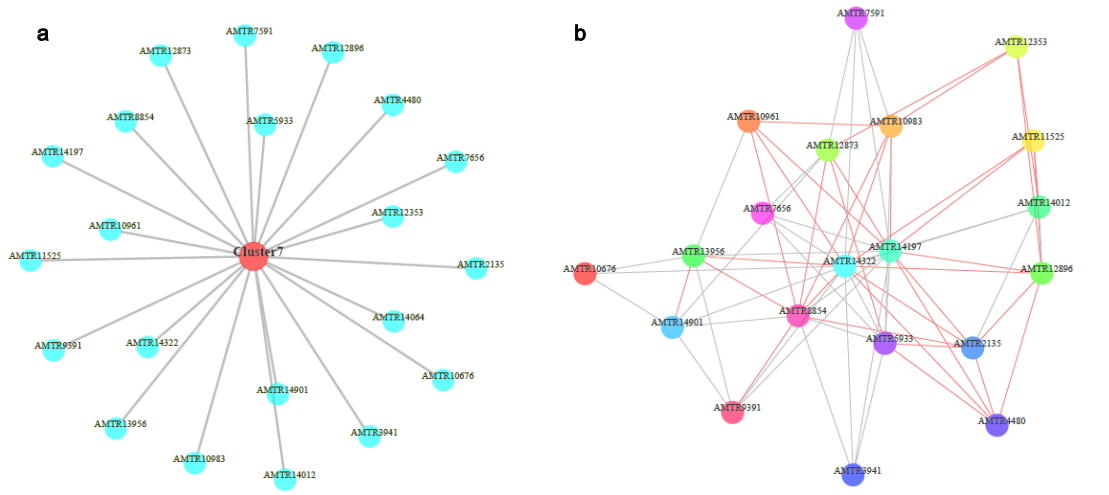

Figure 2. **Visualization of a cluster and isoforms in the cluster.** (a) Graph showing all isoforms in a cluster. (b) Network graph showing identities of pairwise alignments in a cluster.

3.2 Feature construction

Feature construction means to extract features from sequences around splice sites. We have compiled a compendium of 511 unique features that covers major factors known to shape introns and/or exons in *AStrap*, including (see

our paper for more details)

1) Position-specific weight matrix (PWM) for donor and acceptor sites.
2) Pattern of dinucleotide motifs at intron/exon junctions.
3) AS length.
4) Divisibility of AS length by three.
5) Number of occurrences of stop codons in the AS region.
6) GC-content of the AS region.
7) Trinucleotide frequencies in the downstream 10 bp and 20 bp regions of donor sites.
8) Composition, transition, and distribution (CTD) in different regions.

Users can use the function *getFeature* to construct all the above features. In fact, feature construction has been embedded in the function *AStrap* (see below) , users therefore don't need to carry out this step. In spite of that, here is a simple example to illustrate how to extract features from sequences around splice site based on sequence alignment data. The data should contain at least the start of subject (column *Start*), the end of subject (column *Send*), the name of subject (column *Sname*) in pairwise alignment. In this study, the longer isoform is called subject, and the shorter one is called object in pairwise alignment.

```
##Loading example data
>load(system.file("data","sample_Alignddata.Rdata",
                  package = "AStrap"))
>head(Alignddata)
     Qname    Sname Qstart Qend Sstart Send Coverage identity
2 AMTR10014 AMTR8587    691  692    689 1081     98.1     99.8
3  AMTR1069  AMTR712    584  585    599  707    100.0     99.5
4 AMTR10781 AMTR7740    279  280    319  425    100.0    100.0
5 AMTR10781 AMTR8876     88   89    128  257    100.0    100.0
6 AMTR10781 AMTR7821     88   89    128  257    100.0    100.0
7 AMTR10781 AMTR7821    279  280    447  553    100.0    100.0

##Extracting sequence around splice sites based on the
##transcript sequences
Alignddata <- extract_IsoSeq_tr(Alignddata,trSequence )
> colnames(Alignddata)
 [1] "Qname"      "Sname"      "Qstart"     "Qend"       "Sstart"
   "Send"       "Coverage"   "identity"
 [9] "num"        "length"     "seq"        "Ddown10"    "Ddown20"
   "Aup10"      "Aup20"      "Aup30"
[17] "donorSeq"   "acceptorSeq"
> head(Alignddata$Ddown10)
[1] "GTGAGTTTCT" "CTGCAATGAA" "GTATAGAAAC" "TTATCTGTAG" "TTATCTGTA
G" "GTATAGAAAC"
> head(Alignddata$Aup10)
```

```
[1] "GGGTTTATAG" "TGTGGAAGAT" "TCTCTTACAG" "TACTACACAG" "TACTACACA
G" "TCTCTTACAG"
##Loading the consensus matrix of sequences of the [-2,+3] region of
acceptor sites
>load(system.file("data","example_PWM_acceptor.Rdata",
                  package = "AStrap"))
##Loading the consensus matrix of the sequences of the [-2,+3] region
of donor sites
>load(system.file("data","example_PWM_donor.Rdata",
                  package = "AStrap"))
##Constructing the feature space
> feature <- getFeature(Alignedata)
> ncol(feature)
[1] 511
```

3.3 Model building and performance evaluation

Two classification models trained on collected AS data from rice and human were integrated in *AStrap*, which could be directly applied for distinguishing among AS types for other species. AS data in rice were retrieved from the previous study (Chamala, et al., 2015), and those in human were identified form the GRCh38 genome annotation using *ASTALAVISTA* (Foissac and Sammeth, 2007). For classification of AS types, we applied and compared three widely used machine-learning techniques, including support vector machine (SVM) implemented in the R package *libsvm* (Chang and Lin, 2011), random forests (RF) implemented in the R package *randomForest* (Liaw and Wiener, 2002), and *adaptive boosting* (AdaBoost) implemented in the R package adabag (Alfaro-Cortés, et al., 2013). According to our analysis (see our paper), the RF-based model performed the best, followed by the AdaBoost-based model, and the SVM-based model performed the worst. Therefore, it is recommended that users adopt RF-based model for prediction of AS types.

```
##Loading AS classification model of rice
> rice_model<- load(system.file("data","rice_model.Rdata",
                  package = "AStrap"))
> rice_model
[1] "rice_SVMmodel" "rice_RFmodel"  "rice_ABmodel"  "trainset"
[5] "testset"       "PWM_Acceptor"  "PWM_Donor"
##Rice SVM-based model
> class(rice_SVMmodel)
[1] "svm.formula" "svm"
##Rice RF-based model
> class(rice_RFmodel)
[1] "randomForest.formula" "randomForest"
```

```
##Rice AdaBoost-based model
> class(rice_ABmodel)
[1] "boosting"

##Loading AS classification model of human
> human_model<- load(system.file("data","human_model.Rdata",
                  package = "AStrap"))
> human_model
[1] "human_SVMmodel" "human_RFmodel"  "human_ABmodel"
[4] "trainset"       "testset"       "PWM_Acceptor"
[7] "PWM_Donor"
##Human SVM-based model
> class(human_SVMmodel)
[1] "svm.formula" "svm"
##Human RF-based model
> class(human_RFmodel)
[1] "randomForest.formula" "randomForest"
##human AdaBoost-based model
> class(human_ABmodel)
[1] "boosting"
```

Meanwhile, users can also train a specific classification model on their own data sets using function *buildTrainModel*. Specified number of AS events of each type from the input data are randomly selected for model training and test based on the parameter *chooseNum* (default: 1000). Two thirds samples from these randomly selected AS events were used for training and the remaining AS events were used for test by default (parameters: *proTrain* and *proTest*) . Users can also employ all input data by the parameter *use.all* (default: FALSE) for model building. The classification method can be chosen using parameter *classifier*, including SVM, RF (default), and AdaBoost. In addition, users can filter training and test data according to the AS length by parameter *ASlength* (default: 0), and adopt other classifiers, see below **Additional notes.**

```
##Loading example alternative splicing data
> path <- system.file("extdata","sample_riceAS.txt",
                      package = "AStrap")
> rice_ASdata <-read.table(path,sep="\t",head = TRUE,
                           stringsAsFactors = FALSE)
> head(rice_ASdata)
    chr   coord1   coord2   coord3   coord4 strand type
1 Chr12 10771270 10772247 10771366 10772247      + AltD
2  Chr2  8979499  8979340  8979494  8979340      - AltD
3  Chr2  4168154  4167750  4167948  4167750      - AltD
4  Chr8 22093271 22092763 22093024 22092763      - AltD
5  Chr5 12995844 12996155 12995866 12996155      + AltD
6  Chr1 33792362 33792568 33792417 33792568      + AltD
```

```
##Loading genome using the package BSgenome
> library("BSgenome.Osativa.MSU.MSU7")
> rice_ASdata<- extract_IsoSeq_ge(rice_ASdata,Osativa)
> names(rice_ASdata)
 [1] "chr"        "coord1"     "coord2"     "coord3"     "coord4"
    "strand"     "type"       "length"
 [9] "seq"        "up"         "down"       "Ddown10"    "Ddown20"
    "Aup10"      "Aup20"      "Aup30"
[17] "donorSeq"    "acceptorSeq"
##Classification model building based on the random forest method
>library(randomForest)
>library(ROCR)
>library(ggplot2)
> model <- buildTrainModel(rice_ASdata, chooseNum = 100,
                      proTrain = 2/3, proTest = 1/3, ASlength =0,
                      classifier = "rf", use.all = FALSE)
##Or classification model building based on SVM method
>library(e1071)
>library(ROCR)
>library(ggplot2)
> model <- buildTrainModel(rice_ASdata, chooseNum = 100,
                      proTrain = 2/3, proTest = 1/3, ASlength =0,
                      classifier = "svm", use.all = FALSE)

##Or classification model building based on AdaBoost method
>library(adabag)
>library(ROCR)
>library(ggplot2)
> model <- buildTrainModel(rice_ASdata, chooseNum = 100,
                      proTrain = 2/3, proTest = 1/3, ASlength =0,
                      classifier = "adaboost", use.all = FALSE)
```

This function returns a list, including training set, test set, fitted model, predicted classification results, evaluation matrix of the fitted model and an ROC curve (**Figure 3**). To know better about the performance evaluation of a classification model, please refer to https://en.wikipedia.org/wiki/Precision_and_recall.

```
##Performance evaluation
> names(model)
[1] "trainSet"         "testSet"          "randomforestModel"
[4] "randomforestPre"   "accuracy"          "confusion"
[7] "evaluate"         "ROC"
> model$evaluate
    precision       sp    recall       f1
```

```
AltA 0.9375000 0.9797980 0.7142857 0.8108108
AltD 0.7352941 0.9181818 0.8064516 0.7692308
ES   0.7407407 0.9391304 0.7692308 0.7547170
IR   0.8541667 0.9292929 0.9761905 0.9111111
mean 0.8169254 0.9416008 0.8165396 0.8114674
> model$confusion
     true
pred   AltA AltD ES IR
  AltA   30    0  2  0
  AltD    5   25  3  1
  ES      6    1 20  0
  IR      1    5  1 41
> model$accuracy
[1] 0.822695
```

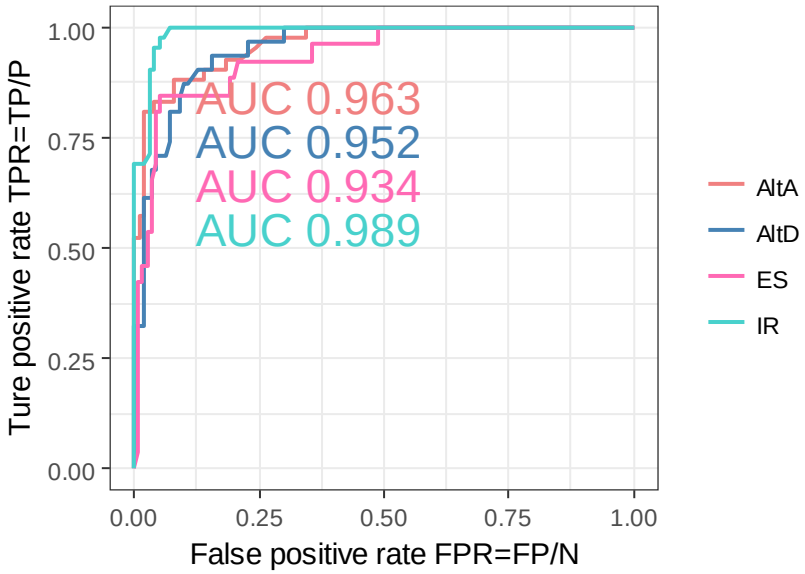

Figure 3. **ROC curves of AS event detection using example rice data in AStrap.**

3.4 Identification of AS events and prediction of AS types

This section describes the identification of AS events based on pairwise alignment of isoforms of the same cluster and prediction of AS types based on the fitted model. Using different fitted models may product different prediction results. We recommend using the RF-based model according to the results in our study. In order to detect AS events and distinguish among AS types at the transcriptome level, we have constructed the function *AStrap* with seven parameters (*cluster.alignment, transcriptSeq, trainModel, identity, coverage, bias, ASlength*), which facilitates selecting AS events meeting different criteria.

1) *cluster.alignment*: a data frame holds pairwise alignment of isoforms in

a cluster (see **3.1**).

2) *transcriptSeq*: a XStringSet object holds the transcript sequence (see **3.1**).
3) *trainModel*: a model for which prediction is desired.
4) *identity*: AS detection is performed if the mapping identity above a given threshold (default: 0.7).
5) *coverage*: AS detection is performed if the mapping coverage above a given threshold (default: 0.7).
6) *bias*: maximum number of mismatches in a pairwise sequence alignment is allowed (default: 0).
7) *ASlength*: AS detection is performed if AS length above a given threshold (default: 0).

```
##Loading rice model
> rice_model<- load(system.file("data","rice_model.Rdata",
                          package = "AStrap"))
> rice_model
[1]  "rice_SVMmodel"  "rice_RFmodel"   "rice_ABmodel"   "trainset"
"testset"      "PWM_Acceptor"  "PWM_Donor"
##Identification and prediction based on RF-based model of rice
> result <- AStrap(alignment,trSequence,rice_RFmodel)
> names(result)
[1] "ASevent" "feature" "predict"
> head(result$ASevent)
       Qid Qlength     Sid Slength Clusterid CluterSeqNum Qstart
1 AMTR10014    2675 AMTR8587    3017        3           3    691
2  AMTR1069    1063  AMTR712    1205     1087           2    584
3 AMTR10781    2730 AMTR7740    2966        5           8    279
4 AMTR10781    2730 AMTR8876    2914        5           8     88
5 AMTR10781    2730 AMTR7821    3003        5           8     88
6 AMTR10781    2730 AMTR7821    3003        5           8    279
  Qend Sstart Send identity coverage ASlength        Qalign
1  692    689 1081     99.8     98.1      391      :3-691:692-2625
2  585    599  707     99.5    100.0      107      :1-584:585-1063
3  280    319  425    100.0    100.0      105      :1-279:280-2730
4   89    128  257    100.0    100.0      128        :1-88:89-2730
5   89    128  257    100.0    100.0      128 :1-88:89-279:280-2730
6  280    447  553    100.0    100.0      105 :1-88:89-279:280-2730
            Salign         prediction spliceSeq
1       :1-689:1081-3017         IR    GT-AG
2       :16-599:707-1185         ES    CT-AT
3       :41-319:425-2875         IR    GT-AG
4       :41-128:257-2898       AltA    TT-AG
5 :41-128:257-447:553-3003      AltA    TT-AG
6 :41-128:257-447:553-3003        IR    GT-AG
```

```
 > length(result$feature)
[1] 511
> head(names(result$feature))
[1] "length"     "Muthree"    "donorGT"    "donorGC"    "donorAT"
"acceptorAG"
> head(result$predict)
   1    2    3    4    5    6
  IR   ES   IR AltA AltA   IR
Levels: AltA AltD ES IR
```

The splice isoforms, sequence alignment, AS type and the splice sites of each AS event can be obtained from the result of *AStrap*. In addition, we can call the function *plotAS* to visualize intuitively the result (**Figure 4**).

```
##Visualization of AS events of different AS types
> library(Gviz)
> plotAS(result$ASevent, id = 1)
> plotAS(result$ASevent, id = 7)
> plotAS(result$ASevent, id = 13)
> plotAS(result$ASevent, id = 21)
```

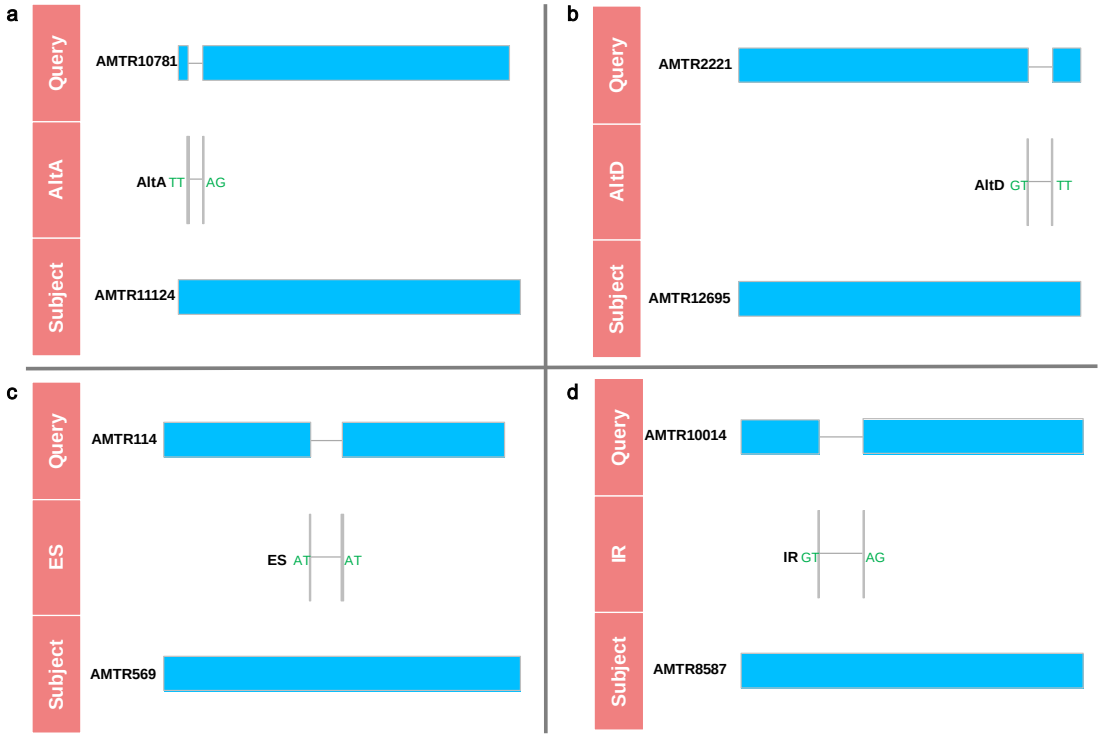

Figure 4. **Visualization of AS events of different AS types.** (a) AltA; (b) AltD; (c) ES; (D) IR. The upper panel shows the query isoform and the lower panel shows the subject isoform. The middle panel shows the splice junction and the splice sites.

4 Additional notes

4.1 Feature selection

Feature selection is the process of selecting a subset of raw features, which can remove irrelevant features and improve model accuracy. However, we did not integrate this step in *AStrap* since the feature selection did not improve the performance according to our analysis. Users who are interested in using smaller feature space can employ the R package *Rweka (Hornik, et al., 2009)* or the data mining tool *WEKA* (Witten and Frank, 2005) for feature selection. In WEKA, the function *AttributeSelectedClassifier* can implement attribute selection with multiple evaluators, such as *PrincipalComponents, SymmetricalUncertAttributeEval, CorrelationAttributeEval, CfsSubsetEval* and *WrapperSubsetEval*. In *AStrap*, calling function *plotGain* can display information gain of top features, which calls function *Rweka::InfoGainAttributeEval* to evaluate the contribution of an attribute by measuring the information gain with respect to the class (**Figure 5**).

```
>##Loading example data
>rice_model<- load(system.file("data","rice_model.Rdata",
                  package = "AStrap"))
>library(RWeka)
>library(ggplot2)
>plotGain(trainset,20)
```

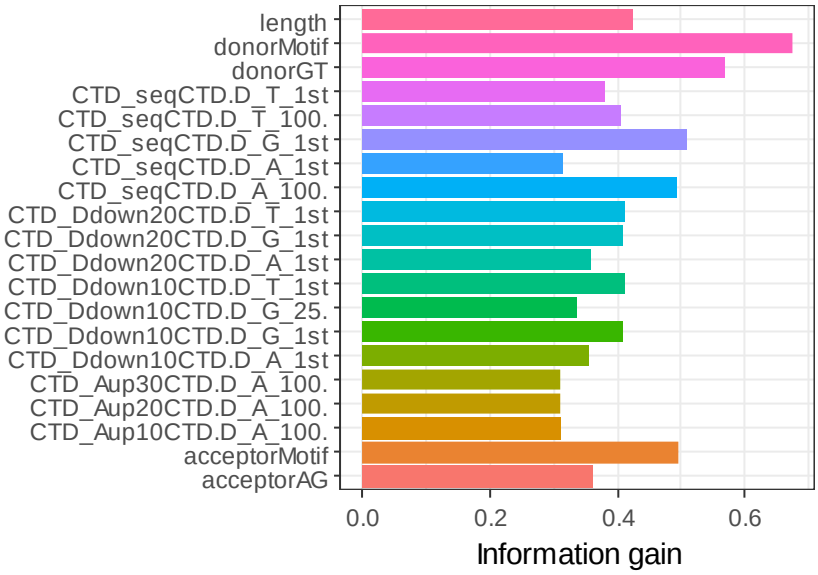

Figure 5. **Top 20 features based on the entropy value.** The Y-axis represents the name of the feature, and the X-axis represents the entropy value.

4.2 Parameter tuning

According to our analysis using different combintations of parameters, the performance of AStrap is quite robust and is not affected greatly by different values of parameters. However, it is easy for users to adjust parameters in *AStrap*. Parameter tuning can be performed based on the grid search by function *tune* of the R package *e1071* (David, et at., 2017). For support vector machine, random forest and adaptive boosting model, users can use function *tune.svm*, *tune.randomForest* and *tune.rpart* to adjust parameters of classifiers, respectively.

4.3 Using additional classifiers

We applied three widely used machine-learning techniques, including SVM, RF, Adaboost in *AStrap*. Besides, users can use other classifiers they are interested in for the prediction of AS types. Function *classify* provides 10 classification algorithms in the R package *BioSeqClass* (Li Hong, 2016), such as bagging, k-nearest neighbor (K-NN), naive bayes, recursive partitioning trees. It also supports feature selection by WEKA (Witten and Frank, 2005).

5 Session information

```
> sessionInfo()
R version 3.3.3 (2017-03-06)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 7 x64 (build 7601) Service Pack 1

locale:
[1] LC_COLLATE=Chinese (Simplified)_People's Republic of China.936

[2] LC_CTYPE=Chinese (Simplified)_People's Republic of China.936
[3] LC_MONETARY=Chinese (Simplified)_People's Republic of China.93
6
[4] LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_People's Republic of China.936

attached base packages:
[1] parallel  stats4    stats     graphics  grDevices utils     datas
ets
[8] methods   base
```

```
other attached packages:
 [1] AStrap_0.1.0        adabag_4.1          caret_6.0-78
 [4] lattice_0.20-35     mlbench_2.1-1       rpart_4.1-11
 [7] randomForest_4.6-12  ROCR_1.0-7         gplots_3.0.1
[10] pROC_1.10.0         ggplot2_2.2.1       BioSeqClass_1.32.0
[13] scatterplot3d_0.3-40 BSgenome_1.42.0     Biostrings_2.42.1
[16] XVector_0.14.1      RWeka_0.4-34        igraph_1.1.2
[19] rtracklayer_1.34.2   GenomicRanges_1.26.4 GenomeInfoDb_1.10.3

[22] IRanges_2.8.2       S4Vectors_0.12.2    BiocGenerics_0.20.0
[25] stringr_1.2.0

loaded via a namespace (and not attached):
 [1] nlme_3.1-131             bitops_1.0-6
 [3] lubridate_1.7.1          dimRed_0.1.0
 [5] tools_3.3.3              R6_2.2.2
 [7] KernSmooth_2.23-15       lazyeval_0.2.0
 [9] colorspace_1.3-2         nnet_7.3-12
[11] withr_2.1.0              tidyselect_0.2.2
[13] mnormt_1.5-5             klaR_0.6-12
[15] Biobase_2.34.0           sandwich_2.4-0
[17] sfsmisc_1.1-1            caTools_1.17.1
[19] scales_0.5.0             DEoptimR_1.0-8
[21] mvtnorm_1.0-6            psych_1.7.8
[23] robustbase_0.92-7        Rsamtools_1.26.2
[25] foreign_0.8-69           pkgconfig_2.0.1
[27] RWekajars_3.9.1-3        rlang_0.1.2
[29] ddalpha_1.3.1            bindr_0.1
[31] zoo_1.8-0                combinat_0.0-8
[33] BiocParallel_1.8.2       gtools_3.5.0
[35] dplyr_0.7.4              ModelMetrics_1.1.0
[37] RCurl_1.95-4.8           magrittr_1.5
[39] modeltools_0.2-21        Matrix_1.2-11
[41] Rcpp_0.12.13             munsell_0.4.3
[43] yaml_2.1.14              stringi_1.1.5
[45] multcomp_1.4-7           MASS_7.3-47
[47] SummarizedExperiment_1.4.0 zlibbioc_1.20.0
[49] plyr_1.8.4               recipes_0.1.2
[51] grid_3.3.3               strucchange_1.5-1
[53] gdata_2.18.0             splines_3.3.3
[55] party_1.2-3              reshape2_1.4.2
[57] codetools_0.2-15         CVST_0.2-1
[59] XML_3.98-1.9             glue_1.1.1
[61] foreach_1.4.3            gtable_0.2.0
```

```
[63] purrr_0.2.3              tidyr_0.7.1
[65] kernlab_0.9-25           assertthat_0.2.0
[67] DRR_0.0.3                gower_0.1.2
[69] coin_1.2-1               prodlim_1.6.1
[71] broom_0.4.3              tree_1.0-37
[73] e1071_1.6-8              class_7.3-14
[75] survival_2.41-3          timeDate_3042.101
[77] RcppRoll_0.2.2           tibble_1.3.4
[79] rJava_0.9-8              iterators_1.0.8
[81] GenomicAlignments_1.10.1   bindrcpp_0.2
[83] lava_1.5.1               TH.data_1.0-8
[85] ipred_0.9-6
```

References

Alfaro-Cortés, E., Gámez, M. and Garc'ia, N. adabag An R Package for Classification with Boosting and Bagging. *Journal of Statistical Software* 2013;54:1-35.

Chamala, S., *et al.* Genome-wide identification of evolutionarily conserved alternative splicing events in flowering plants. *Front Bioeng Biotechnol* 2015;3:33.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2011;2(3):1-27.

Foissac, S. and Sammeth, M. ASTALAVISTA: dynamic and flexible analysis of alternative splicing events in custom gene datasets. *Nucleic Acids Res.* 2007;35(Web Server):W297-W299.

Fu, L., *et al.* CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 2012;28(23):3150-3152.

Gordon, S.P., *et al.* Widespread polycistronic transcripts in fungi revealed by single-molecule mRNA sequencing. *PloS one* 2015;10(7):e0132628.

Herve, P. BSgenome: Infrastructure for Biostrings-based genome data packages. In.; 2014.

Hornik, K., Buchta, C. and Zeileis, A. Open-source machine learning: R meets Weka. *Computational Statistics* 2009;24(2):225-232.

Liaw, A. and Wiener, M. Classification and Regression by RandomForest. *R News* 2002;2:18-22.

Liu, X., *et al.* Detecting alternatively spliced transcript isoforms from single-molecule long-read sequences without a reference genome. *Mol Ecol Resour* 2017.

Wang, E.T., *et al.* Alternative isoform regulation in human tissue transcriptomes. *Nature* 2008;456(7221):470 - 476.

Witten, I. and Frank, E. Data Mining: Practical Machine Learning Tools and Techniques. 2005.

Wu, T.D. and Watanabe, C.K. GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* 2005;21(9):1859-1875.
Li Hong. BioSeqClass: Classification for Biological Sequences. R package version 1.32.0.; 2014.
David, M., *et at*. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-8.;2017.