S-IRFindeR: an R package for detecting and measuring Intron Retention using RNA-seq data

Lucile Broséus

Introduction

Abstract

Accurate quantification of intron retention (IR) levels is currently the crux for detecting and interpreting the function of retained introns. S-IRFindeR implements our new approach to measuring intron retention levels using second generation RNA-seq data, the Stable Intron Retention ratio (SIRratio). In addition to this, the package also enables to detect IR events and compute observed IR rates using third generation RNA-seq data.

Summary

Package Installation

The R package *SIRFindeR* can be installed from GitHub by copy-pasting the following code line:

```
devtools::install_github("lbroseus/SIRFindeR")
```

Then, load *SIRFindeR*:

```
suppressPackageStartupMessages( library(SIRFindeR) )
```

Estimating IR-levels using second generation RNA-seq data

The package SIRFindeR implements our proposed method to estimate Intron Retention levels from short RNA-seq data (Please see reference *1* for more details).
On a bam file from one ultra-deep RNA-seq sample, computations can usually be performed on a PC, within 10-15 minutes. For large multi-sample experiments, it might be better to dispatch computations a server.

Prerequisite: short read alignment using STAR

In order to compute SIRratio values, we will need two files from *STAR* alignments: - the bam file with read alignments; - the *SJ.out.tab* file.

Here is a typical command line to perform the required genomic alignment step using *STAR*:

```
STAR  --genomeDir $STARindex  \                    # Path to the STAR index
      --readFilesIn  Reads_1.fq,Reads_2.fq\        # Read files
      --outFileNamePrefix $mySample  \             # A prefix for output files
      --runThreadN $nthreads  \                    # Number of threads
      --outStd BAM_Unsorted --outSAMtype BAM   Unsorted \
      --outSAMstrandField intronMotif --outSAMunmapped None --outFilterMultimapNmax 1
```

Note: SIRFindeR was tested using reference data from *ENSEMBL*: https://www.ensembl.org/index.html.

Computing SIRratios

Once the alignment step is completed, SIRratios can be obtained from a wrapper function as follows:

```
#Input data
bamFile <- "Unsorted.bam"
junctionFile <- "SJ.out.tab"

#Read length:
readLength <- 100
#Indicate whether your read are single-end ("SE") or paired-end ("PE")
libraryType <- "SE"

computeSIRratio(gtf,
                bamFile = bamFile,
                readLength = readLength,
                libraryType = libraryType,
                junctionFile = junctionFile,
                saveDir = saveDir)
```

This will create several sample-specific files in the directory *saveDir*.
Among whose:
- *SIRratio.txt*: which contains final results with SIRratio values for each sample-curated independent intron;
- *ResultsByIntron.txt*: with the SIRratio values per independent intron. Independent introns are reference genomic intervals common to all samples from the same organism. This is the file you will need if you want to aggregrate and compare several samples.

Example: IR events on chr10 in the GM12878 human cell line

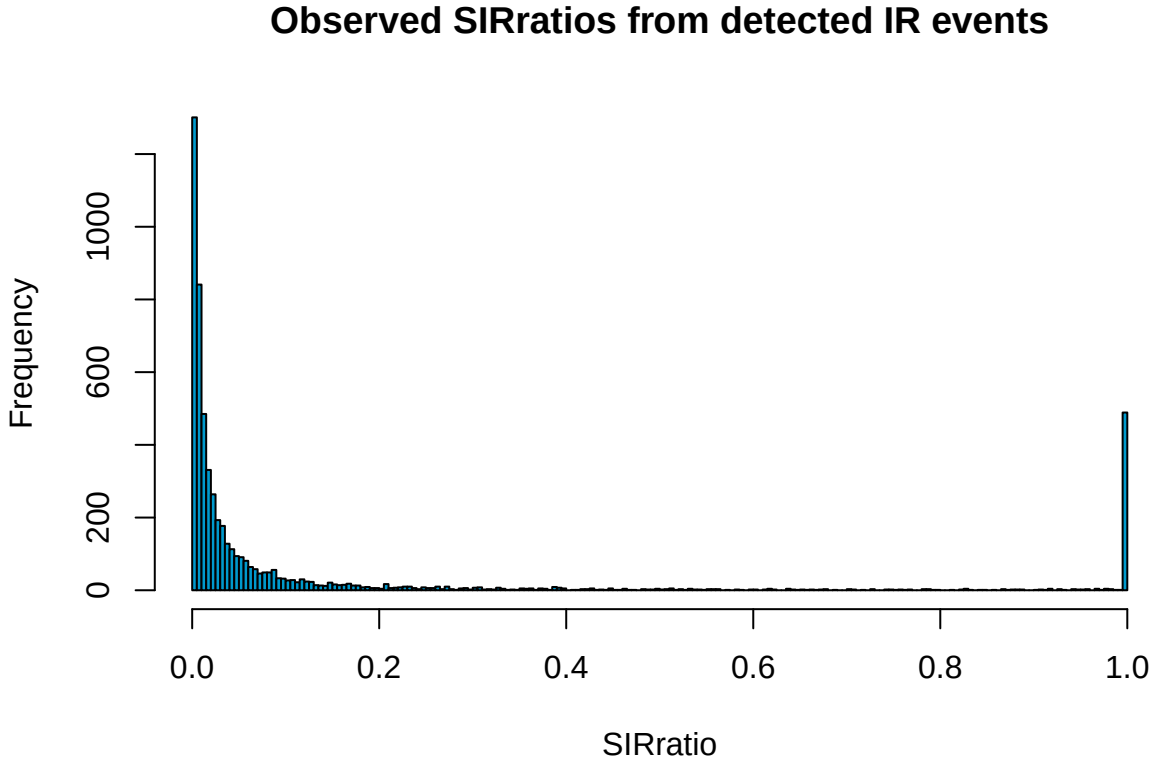
An excerpt of a typical final output on short read data is provided with the package. These are SIRratios for introns on the chromosome 10, computed on an Illumina RNA-seq experiment from the human cell lines GM12878 (downloaded from https://www.ncbi.nlm.nih.gov/sra?term=SRX159821).

```
## SIRratios:

SIRFile <- system.file("extdata", "ResultsByIntron.txt", package = "SIRFindeR")
SIRratios <- read.table(file = SIRFile, header = T)
```

```
# Quick Overview:

head( SIRratios[, -1] )
#>   seqnames  start    end width strand  gene_name         gene_id SpliceLeft
#> 1       10  44902  44951    50      + AL713922.1 ENSG00000237297          0
#> 2       10  45205  45308   104      + AL713922.1 ENSG00000237297          0
#> 3       10  45406  45833   428      + AL713922.1 ENSG00000237297          0
#> 4       10  45883  46244   362      + AL713922.1 ENSG00000237297          0
#> 5       10  46360  46841   482      + AL713922.1 ENSG00000237297          0
#> 6       10 135560 179993 44434      +    ZMYND11 ENSG00000015171         35
#>   SpliceRight SpliceMax intronicCount   SIRratio
#> 1           0         0             0 0.00000000
#> 2           0         0             0 0.00000000
#> 3           0         0             0 0.00000000
#> 4           0         0             0 0.00000000
#> 5           0         0             0 0.00000000
#> 6          33        35           332 0.02910349
```

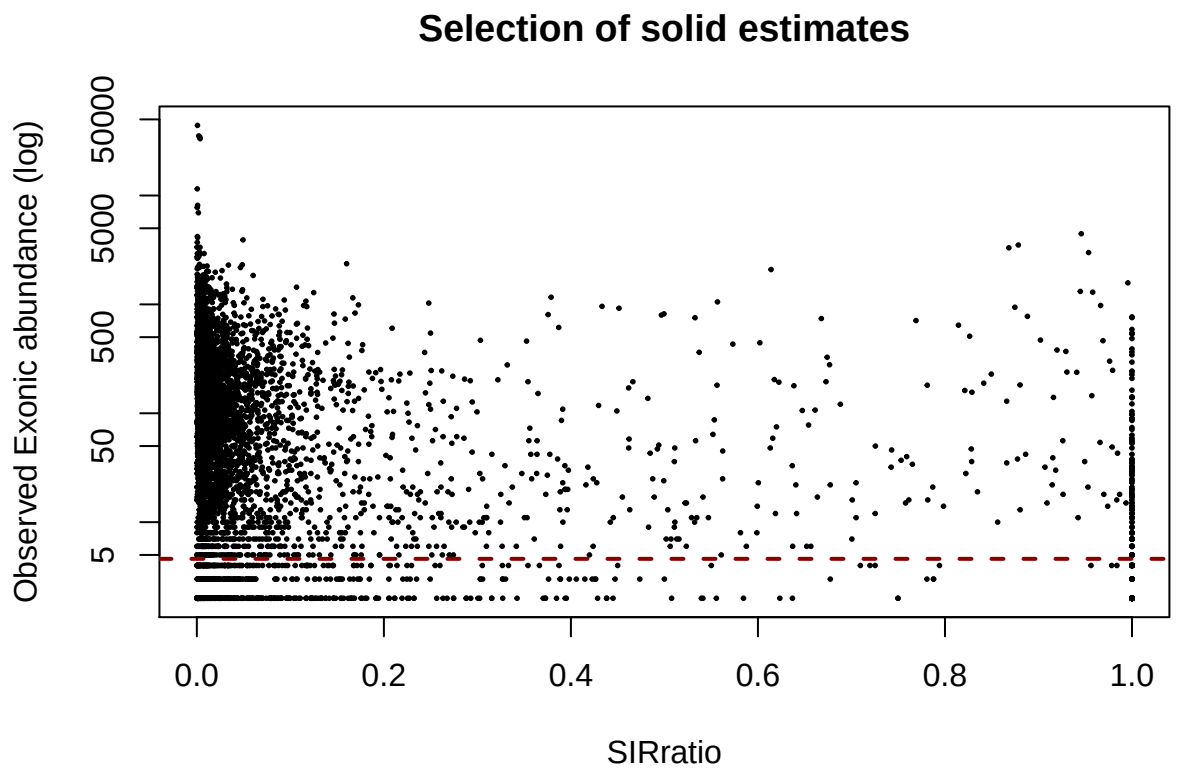
Let's have a look at the overall observed distribution of SIRratios for non-zero introns:

```
hist(main = "Observed SIRratios from detected IR events",
     xlab = "SIRratio",x = SIRratios$SIRratio[SIRratios$SIRrati>0], breaks = seq(0,1,0.005),
     col = "deepskyblue3")
```

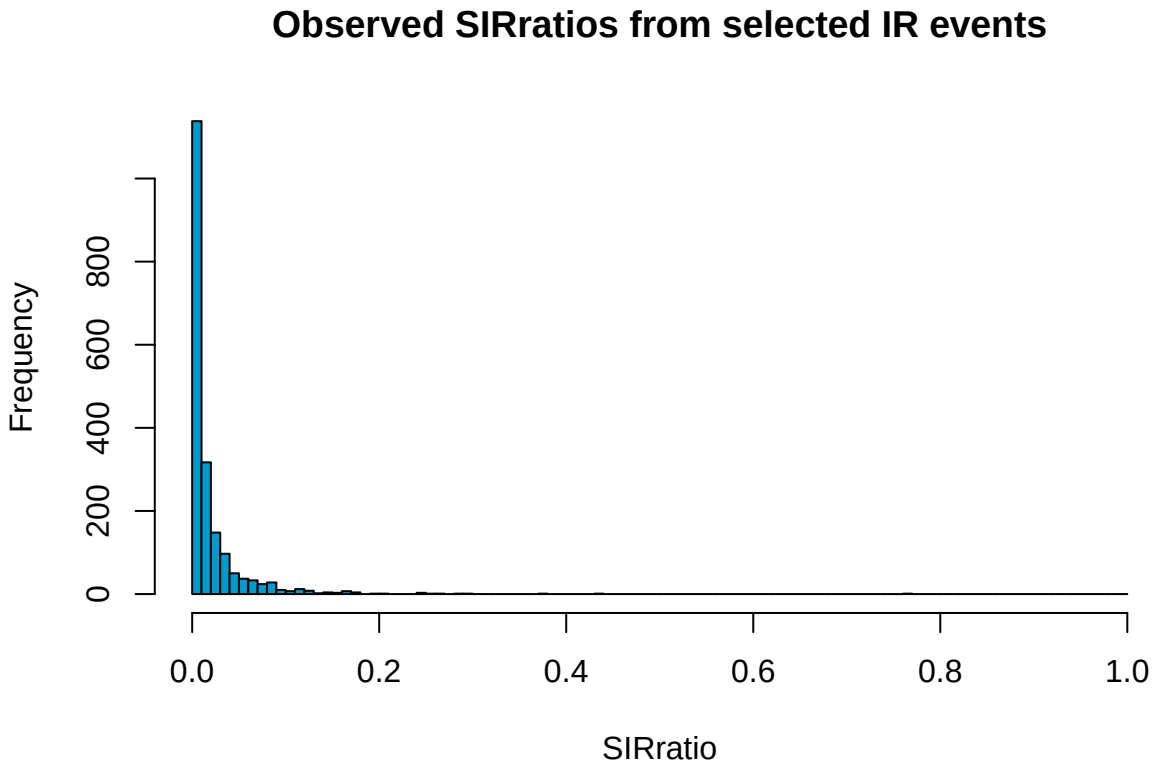


When coverage information is sparse, estimates are less reliable. Thus, if exonic and intronic abundances are below say, 100, counts (ie: $SpliceMax+intronicCout<100$ -below the red dashed line-), SIRratio should be utilized with caution. As you can see on the graph below, very high levels (over 0.9) are likely spurious values.

```
reliabilityThr <- 100

nonZero <- which(SIRratios$SIRratio>0)

plot(main = "Selection of solid estimates",
     xlab = "SIRratio",
     x = SIRratios$SIRratio[nonZero],
     ylab = "Observed Exonic abundance (log)",
     y = SIRratios$SpliceMax[nonZero]+SIRratios$intronicCount[nonZero]+1,
     log = 'y', pch = 19, cex = 0.25)
abline(h = log(reliabilityThr), lty = "dashed", lwd = 2, col = "darkred")
```




```
reliableFeatures <- which(SIRratios$SIRratio>0 & SIRratios$SpliceMax>reliabilityThr)

hist(main = "Observed SIRratios from selected IR events",
     xlab = "SIRratio",x = SIRratios$SIRratio[reliableFeatures], breaks = seq(0,1,0.01),
     col = "deepskyblue3")
```



A summary of reliable IR levels obtained using short read data:

```
summary(SIRratios$SIRratio[reliableFeatures])
#>      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
#> 0.0000703 0.0029567 0.0073916 0.0194494 0.0200723 0.7690533
```

Estimating IR-levels using third generation RNA-seq data

Several functions for estimating (observed) IR rates are available in SIRFindeR.
Our proposed method is reference-based (evaluated intron are extracted from a reference transcriptome annotation, ie: a gtf file), and makes use of (spliced) genome alignments.
We intend to make it more flexible in the near future.

Prerequisite: long read alignment

Thus, first, you will need to align long reads onto a reference genome. There exist several long read splice-aware aligners. We mainly tested two of them (*Minimap2* and *GMAP*). For our applications, *Minimap2* achieved better results on raw data; while being definitely faster. They both showed comparable performances on corrected data.
If, for some reason, you want to use *GMAP*, we strongly advise to perform long read correction before aligning the reads; this may improve significantly alignment rates and accuracy (cf: reference *4*).
In case you also have matched short read data, you can perform hybrid correction using *TALC*: https://github.com/lbroseus/TALC.

Here are typical command lines for aligning *Oxford Nanopore* long read using *Minimap2* (https://github.com/lh3/minimap2, reference *3*):

```
minimap2 -ax splice -uf -k14 \
                    yourReferenceGenome.fa \ #reference genome (eg: fasta from ENSEMBL)
                    yourSample.fa \          #fasta/fastq file with long read sequences
                    -t $nthreads \           #Number of threads
                    >yourSample.mmap2.sam    #A name for the output file
```

Computing observed IR rates

SIRFindeR also implements functionalities to compute observed IR rates using third generation RNA-seq data.
Note: these are "naive" estimates of the *true* IR-levels. In many cases, the intron coverage is so low (ie: only a few counts) that the observed value is not a reliable measure. We suggest not to interprete observed IR rates when intron abundances (column *IntronAbundance*) are too low (eg: below 30 counts).

Again, all you will need is a bam file with genomic alignments from a whole sample, and a reference transcriptome in a gtf file, from which to define intronic regions.

```
# Input data and paths

gtf <- "MyBelovedTranscriptAnnot.gtf"
bamFile <- "yourSample.bam"
saveDir <- "Where/To/Output/Results"
```

```
# Calculate IR rates

computeIRrates(gtf = gtf, bamFile = bamFile, saveDir = saveDir, keepSecondaryAlignments = FALSE)
```

This will:
1. extract annotated intron intervals from the reference transcriptome;
2. detect intron retention events (output in *IRevents.txt*);
3. calculate observed IR rates (output in *IRrates.txt*).
Both *.txt* files will be written in the directory *saveDir*.

Example: IR events on chr10 in the GM12878 human cell line

For illustration purposes, we provide an excerpt of a typical output from SIRFindeR. These are IR events and IR levels on chromosome 10, computed on the publicly available NA12878 direct-RNA dataset (https://github.com/nanopore-wgs-consortium/NA12878/blob/master/RNA.md).

File *IRevents* lists all IR calls with their cognate long read name and the genomic coordinates of the intron. A read can appear several times in the data.frame (eg: if it retains multiple introns):

```
## IR events:

IReventsFile <- system.file("extdata", "IRevents.txt", package = "SIRFindeR")
IRevents <- read.table(file = IReventsFile, header = T)
```

```
# Quick Overview:

head( IRevents)
```

In file *IRrates.txt* you will find, for each reference intron (identified by its genomic coordinates), overall and intronic counts (column *readCount* and *IntronicAbundance* resp.).
Observed IR levels are indicated in column *ratio*.
Note: when *readCount* is zero, the *ratio* is set to zero, by default.

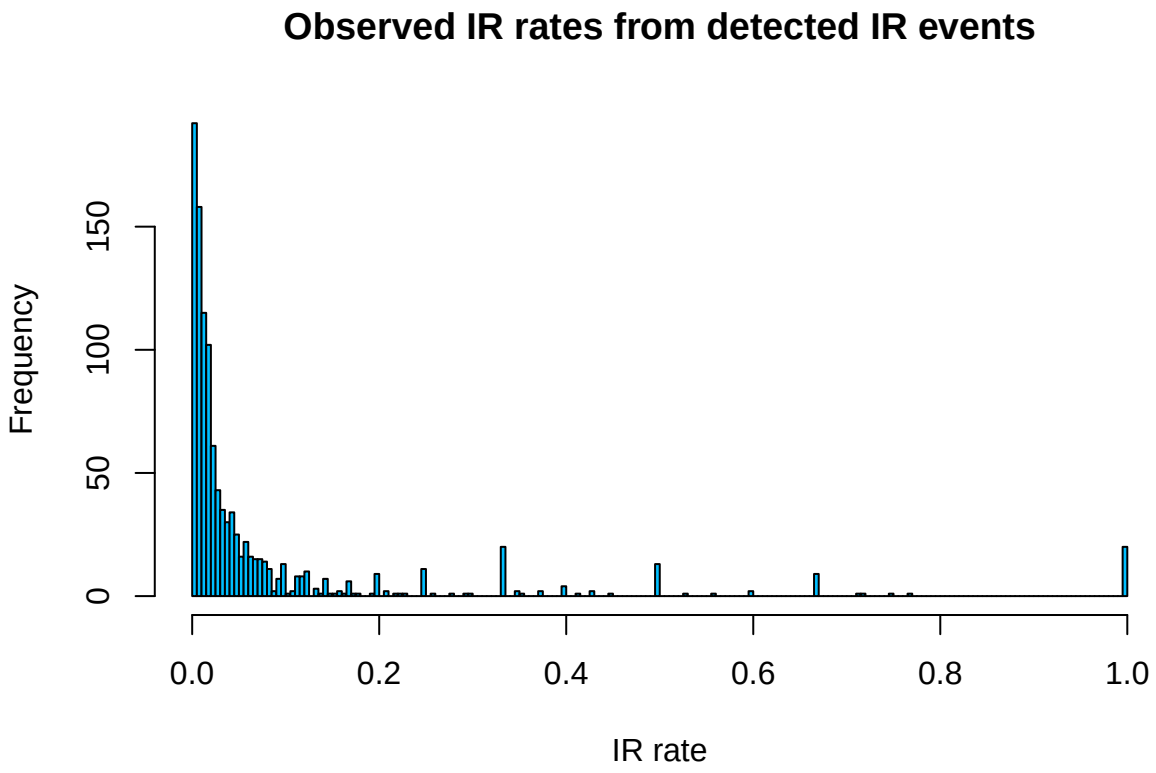
```
## IR rates:

IRratesFile <- system.file("extdata", "IRrates.txt", package = "SIRFindeR")
IRrates <- read.table(file = IRratesFile, header = T)
```

```
# Quick Overview:

head( IRrates )
#>   seqnames start   end width strand         gene_id annotated_intron readCount
#> 1     chr1 12228 12612   385      + ENSG00000223972                0         0
#> 2     chr1 12722 12974   253      + ENSG00000223972                0         0
#> 3     chr1 13053 13220   168      + ENSG00000223972                0         0
#> 4     chr1 30040 30266   227      + ENSG00000243485                0         0
#> 5     chr1 30668 30975   308      + ENSG00000243485                0         0
#> 6     chr1 57654 58699  1046      + ENSG00000240361                0         0
#>   intronicAbundance ratio
#> 1                 0     0
```

```
#> 2                  0     0
#> 3                  0     0
#> 4                  0     0
#> 5                  0     0
#> 6                  0     0
```

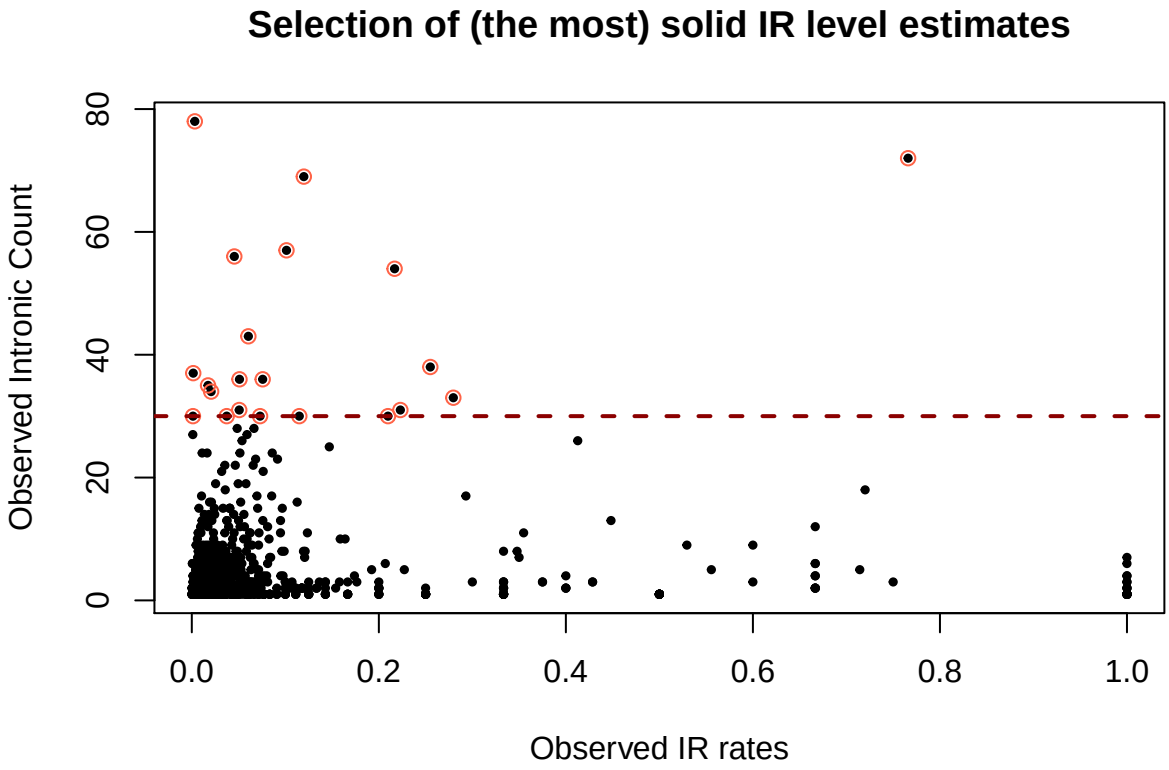
Overall observed distribution of IR rates for "detected" IR events:

```
hist(main = "Observed IR rates from detected IR events",
     xlab = "IR rate",x = IRrates$ratio[IRrates$ratio>0], breaks = seq(0,1,0.005),
     col = "deepskyblue")
```



Select most reliable estimates (eg: having at least 30 read counts - red dots):

```
reliabilityThr <- 30

reliableFeatures <- which(IRrates$intronicAbundance >= reliabilityThr)

cat("There are", length(reliableFeatures), "well-supported values. \n")
#> There are 21 well-supported values.

plot(main = "Selection of (the most) solid IR level estimates",
     xlab = "Observed IR rates",
     x = IRrates$ratio[IRrates$ratio>0],
     ylab = "Observed Intronic Count",
     y = IRrates$intronicAbundance[IRrates$ratio>0],
     pch = 19, cex = 0.5)
points(x = IRrates$ratio[reliableFeatures],
       y = IRrates$intronicAbundance[reliableFeatures], col = "tomato")
abline(h = reliabilityThr, lty = "dashed", lwd = 2, col = "darkred")
```



A summary of reliable IR levels obtained using long read data:

```
summary(IRrates$ratio[reliableFeatures])
#>      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
#> 0.001232 0.037313 0.072993 0.129717 0.209790 0.765957
```

References

1. *Broseus and Ritchie* ***SIRFindeR: stable and accurate quantification of intron retention levels*** *biorXiv. (2020)*
2. *Dobin et al.* ***STAR: ultrafast universal RNA-seq aligner.*** *Bioinformatics. (2013)*
3. *Li.* ***Minimap2: pairwise alignment for nucleotide sequences.*** *Bioinformatics. (2013)*
4. *Broseus et al.* ***TALC: Transcript-level Aware Long Read Correction.*** *Bioinformatics. (2020)*