Package 'svaRetro'

November 12, 2026

Type Package

Title Retrotransposed transcript detection from structural variants

Version 1.18.0

Date 2026-02-18

Description svaRetro contains functions for detecting retrotransposed transcripts (RTs) from structural variant calls. It takes structural variant calls in GRanges of breakend notation and identifies RTs by exon-exon junctions and insertion sites. The candidate RTs are reported by events and annotated with information of the inserted transcripts.

License GPL-3 + file LICENSE

Depends GenomicRanges, rtracklayer, BiocGenerics, StructuralVariantAnnotation, R (>= 4.0)

Imports VariantAnnotation, AnnotationDbi, assertthat, Biostrings, stringr, dplyr, methods, rlang, S4Vectors, Seqinfo, GenomeInfoDb, GenomicFeatures, utils

Suggests TxDb.Hsapiens.UCSC.hg19.knownGene, org.Hs.eg.db, ggplot2, devtools, testthat (>= 2.1.0), roxygen2, knitr, BiocStyle, plyranges, circlize, tictoc, IRanges, stats, SummarizedExperiment, rmarkdown

RoxygenNote 7.1.1

Encoding UTF-8

VignetteBuilder knitr

biocViews DataImport, Sequencing, Annotation, Genetics, VariantAnnotation, Coverage, VariantDetection

BugReports https://github.com/PapenfussLab/svaRetro/issues

git_url https://git.bioconductor.org/packages/svaRetro

git_branch RELEASE_3_23

git_last_commit c3370c7

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Ruining Dong [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1433-0484>)

Maintainer Ruining Dong <lnyidrn@gmail.com>

Contents

.combineMatchingTranscripts 2
.scoreByTranscripts 3
.txs2genesym 3
rtDetect 4
svaRetro 5
%na% 5
%null% 5

Index **7**

`.combineMatchingTranscripts` *Combining matching transcripts*

Description

Combining matching transcripts

Usage

```
.combineMatchingTranscripts(gr, names)
```

Arguments

`gr`	A GRanges object
`names`	A vector of granges names.

Details

This is an internal function used to merge all overlapping transcripts of a breakpoint into one vector.

Value

A list of vectors. Each vector is named with the name of the corresponding granges.

.scoreByTranscripts *Ranking matching transcripts*

Description

Ranking matching transcripts

Usage

```
.scoreByTranscripts(genes, transcripts.col)
```

Arguments

genes	TxDb object of genes. hg19 and hg38 are supported in the current version.
transcripts.col	A vector of transcript names.

Details

This is an internal function which returns overlapping transcript names with ranking scores. The ranking score is the proportion of exon-exon fusions (intronic deletion events) detected for a given transcript.

Value

A dataframe with two columns, tx_name and score.

.txs2genesym *Adding gene symbol annotations*

Description

Adding gene symbol annotations

Usage

```
.txs2genesym(txs, unique.genesyms = TRUE)
```

Arguments

txs	A list of transcript ids in UCSC format.
unique.genesyms	TRUE or FALSE. If TRUE, the converted gene symbols will remove duplicates.

Details

This is an internal function which takes a list of txs in UCSC id format as input and convert the txs to gene symbol.

Value

A list of names in gene symbols

`rtDetect` *Detecting retrotranscript insertion in nuclear genomes.*

Description

Detecting retrotranscript insertion in nuclear genomes.

Usage

```
rtDetect(gr, genes, maxgap = 100, minscore = 0.4)
```

Arguments

`gr`	A GRanges object
`genes`	TxDb object of genes. hg19 and hg38 are supported in the current version.
`maxgap`	The maxium distance allowed on the reference genome between the paired exon boundries.
`minscore`	The minimum proportion of intronic deletions of a transcript should be identified.

Details

This function searches for retroposed transcripts by identifying breakpoints supporting intronic deletions and fusions between exons and remote loci. Only BND notations are supported at the current stage.

Value

A GRangesList object, named insSite and rt, reporting breakpoints supporting insert sites and retroposed transcripts respectively. 'exon' and 'txs' in the metadata columns report exon_id and transcript_name from the 'genes' object.

Examples

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
genes <- TxDb.Hsapiens.UCSC.hg19.knownGene
vcf.file <- system.file("extdata", "diploidSV.vcf",
                        package = "svaRetro")
vcf <- VariantAnnotation::readVcf(vcf.file, "hg19")
gr <- breakpointRanges(vcf, nominalPosition=TRUE)
rt <- rtDetect(gr, genes, maxgap=30, minscore=0.6)
```

svaRetro *svaRetro: a package for retrotransposed transcript detection*

Description

svaRetro contains functions for detecting retrotransposed transcripts from structural variant calls.

Details

For more details on the features of StructuralVariantAnnotation, read the vignette: ‘browseVignettes(package = "svaRetro")‘

`%na%` *Replaces the NA values in a with corresponding values in b*

Description

Replaces the NA values in a with corresponding values in b

Usage

```
a %na% b
```

Arguments

a, b	objects to be tested or coerced.

Value

The altered object.

`%null%` *Uses b if a is NULL*

Description

Uses b if a is NULL

Usage

```
a %null% b
```

Arguments

a, b	objects to be tested or coerced.

Value

An un-null object.

Index

∗ **internal**
.combineMatchingTranscripts, 2
.scoreByTranscripts, 3
.txs2genesym, 3
.combineMatchingTranscripts, 2
.scoreByTranscripts, 3
.txs2genesym, 3
%na%, 5
%null%, 5

rtDetect, 4

svaRetro, 5