Package ‘Isosceles’

July 27, 2023

Title Isoform Single-Cell and Long-read Expression Suite

Version 0.0.1.9000

Description Transcript detection and quantification from long reads.

Depends R (>= 4.2.0),
SingleCellExperiment (>= 1.18.0)

Imports utils (>= 4.2.0),
methods (>= 4.2.0),
stats (>= 4.2.0),
rlang (>= 1.0.4),
assertthat (>= 0.2.1),
magrittr (>= 2.0.3),
tibble (>= 3.1.7),
tidyselect (>= 1.1.2),
dplyr (>= 1.0.9),
tidyr (>= 1.2.0),
glue (>= 1.6.2),
digest (>= 0.6.29),
Rcpp (>= 1.0.9),
Matrix (>= 1.4-1),
BiocParallel (>= 1.30.3),
BiocNeighbors (>= 1.14.0),
S4Vectors (>= 0.34.0),
BiocGenerics (>= 0.42.0),
Biostrings (>= 2.64.0),
BSgenome (>= 1.64.0),
GenomeInfoDb (>= 1.32.2),
IRanges (>= 2.30.0),
GenomicRanges (>= 1.48.0),
Rsamtools (>= 2.12.0),
GenomicAlignments (>= 1.32.1),
rtracklayer (>= 1.56.1),
GenomicFeatures (>= 1.48.3),
SummarizedExperiment (>= 1.26.1),
igraph (>= 1.3.4),
scuttle (>= 1.6.2),

fastmatch (>= 1.1-3)

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Suggests testthat (>= 3.0.0),
tools (>= 4.2.0),
knitr (>= 1.39),
rmarkdown (>= 2.14),
BiocStyle (>= 2.24.0),
pheatmap (>= 1.0.12),
viridis (>= 0.6.2),
ggbio (>= 1.44.1),
ggplot2 (>= 3.3.6),
biovizBase (>= 1.44.0),
dittoSeq (>= 1.8.1),
scran (>= 1.24.0),
scater (>= 1.24.0),
bluster (>= 1.6.0)

Config/testthat/edition 3

LinkingTo Rcpp (>= 1.0.9),
RcppArmadillo (>= 0.11.2.0.0)

VignetteBuilder knitr

R topics documented:

`Isosceles-package` *Isosceles: Isoform Single-Cell and Long-read Expression Suite*

Description

Transcript detection and quantification from long reads

Author(s)

Tim Sterne-Weiler sternewt@gene.com

Michal Kabza michal.kabza@contractors.roche.com

`export_gtf` *Data export to a GTF file*

Description

Export transcripts from a SummarizedExperiment to a GTF file

Usage

```
export_gtf(se, file)
```

Arguments

`se`	A transcript-level SummarizedExperiment object returned by the `prepare_transcript_se` function
`file`	A string specifying the output file path

Value

Nothing is returned

extract_read_structures *Read structure extraction from BAM files*

Description

Extract non-redundant read structures from one or multiple BAM files

Usage

```
extract_read_structures(bam_files, chunk_size = 1e+06, ncpu = 1)
```

Arguments

bam_files A character vector containing BAM file paths

chunk_size An integer scalar specifying the chunk size for reading the BAM files

ncpu An integer scalar specifying the number of cores to use for multicore parallelization

Value

A data frame containing non-redundant read structure data obtained from the BAM files

merge_sc_neighbors *Merging neighboring cell TCC values in scRNA-Seq data*

Description

Prepares a TCC SummarizedExperiment object where count values from the nearest k neighbors are added to the count values of each cell

Usage

```
merge_sc_neighbors(se_tcc, pca_mat, k = 10, use_annoy = FALSE, ncpu = 1)
```

Arguments

se_tcc A TCC SummarizedExperiment object returned by the prepare_tcc_se function

pca_mat A matrix containing PCA coordinates of each cell

k An integer scalar specifying the number of nearest neighbors to use

use_annoy A logical scalar indicating whether to use the Annoy algorithm for approximate nearest neighbor identification (recommended for big datasets)

ncpu An integer scalar specifying the number of cores to use for multicore parallelization

Value

A SummarizedExperiment object containing merged TCC data

---

prepare_gene_se *Prepare a gene-level SummarizedExperiment object*

Description

Prepares a gene-level SummarizedExperiment from TCC data

Usage

```
prepare_gene_se(se_tcc)
```

Arguments

`se_tcc` A TCC SummarizedExperiment object returned by a function from the `Isosceles-package`

Value

A SummarizedExperiment object containing gene annotation and quantification data

---

prepare_pseudobulk_se *Prepare a pseudobulk TCC SummarizedExperiment object*

Description

Prepares a pseudobulk TCC SummarizedExperiment from TCC data and given cell labels

Usage

```
prepare_pseudobulk_se(se_tcc, cell_labels)
```

Arguments

`se_tcc` A TCC SummarizedExperiment object returned by the `prepare_tcc_se` function

`cell_labels` A vector or a factor containing cell labels acting as a grouping variable

Value

A pseudobulk SummarizedExperiment object containing TCC annotation and quantification data

prepare_psi_se *Prepare a PSI SummarizedExperiment object*

Description

Prepares a PSI (Percent Spliced In) SummarizedExperiment object for the given transcript-level SummarizedExperiment object. PSI values are calculated for the following types of regions:

- **TSS** - transcription start sites
- **TES** - transcription end sites
- **CE** - core exonic regions
- **RI** - retained intronic regions
- **A5** - 5’ alternative exonic regions
- **A3** - 3’ alternative exonic regions

TSS and TES positions are calculated based on transcripts’ binned start and end coordinates extracted from their identifiers

Usage

```
prepare_psi_se(se, ncpu = 1)
```

Arguments

`se`	A transcript-level SummarizedExperiment object returned by the `prepare_transcript_se` function
`ncpu`	An integer scalar specifying the number of cores to use for multicore parallelization

Value

A SummarizedExperiment object containing PSI annotation and quantification data

prepare_tcc_se *Prepare a TCC SummarizedExperiment object*

Description

Prepares a TCC (Transcript Compatibility Counts) SummarizedExperiment object for the given BAM files and transcript set

Usage

```
prepare_tcc_se(
  bam_files,
  transcript_data,
  run_mode = "strict",
  min_read_count = 1,
  min_relative_expression = 0.1,
  extend_spliced_transcripts = 100,
  is_single_cell = FALSE,
  barcode_tag = "BC",
  chunk_size = 1e+06,
  ncpu = 1
)
```

Arguments

`bam_files` A named character vector containing BAM file paths

`transcript_data` A named list containing transcript data returned by the `prepare_transcripts` function

`run_mode` A string specifying the mode for choosing the transcript set ('strict', 'de_novo_strict', 'de_novo_loose' or 'de_novo_full')

`min_read_count` An integer scalar specifying the read count threshold for transcripts extracted from the BAM files

`min_relative_expression` A numeric scalar specifying the relative expression threshold for transcripts extracted from the BAM files

`extend_spliced_transcripts` An integer scalar specifying the number of base pairs by which transcript starts and ends are extended for spliced read compatibility search

`is_single_cell` A logical scalar specifying if the BAM files contain single cell data

`barcode_tag` A string specifying the name of the BAM file tag containing cell barcodes

`chunk_size` An integer scalar specifying the chunk size for reading the BAM files

`ncpu` An integer scalar specifying the number of cores to use for multicore parallelization

Value

A SummarizedExperiment object containing TCC annotation and quantification data

prepare_transcripts *Transcript data preparation*

Description

Prepare transcript data (reference and extracted from the BAM files) for further analysis

Usage

```
prepare_transcripts(
  gtf_file,
  genome_fasta_file,
  bam_parsed,
  is_technical = FALSE,
  min_intron_length = 30,
  known_intron_motifs = c("GT-AG"),
  rescue_annotated_introns = FALSE,
  known_intron_granges = NULL,
  min_bam_splice_read_count = 2,
  min_bam_splice_fraction = 0.1,
  bin_size = 50
)
```

Arguments

`gtf_file` A string containing a GTF file path

`genome_fasta_file` A string containing a genome FASTA file path

`bam_parsed` A data frame containing non-redundant read structure data returned by the `extract_read_structures` function. If NULL, only reference transcripts are used

`is_technical` A boolean scalar specifying if the GTF file describes technical sequences (e.g. SIRV or ERCC) rather than originating from Ensembl / GENCODE

`min_intron_length` An integer scalar specifying the minimal length of introns to assign strand to

`known_intron_motifs` A character vector specifying the known intron motifs

`rescue_annotated_introns` A logical scalar specifying if introns found in genome annotations should be kept even if they don't have known intron motifs

`known_intron_granges` A GRanges object storing known intron positions (e.g. from short read data) used for transcript classification. If set to NULL, only introns from reference annotations are used

`min_bam_splice_read_count` An integer scalar specifying the read count threshold for splice sites confirmed by aligned reads

`min_bam_splice_fraction`
A numeric scalar specifying the minimum connectivity fraction to a known splice site for splice sites confirmed by aligned reads

`bin_size` An integer scalar specifying the bin size for transcript start and end position binning

Value

A named list containing following elements:

tx_df a data frame storing extracted transcript data

tx_granges a GRanges object storing genomic positions of extracted transcript

tx_exon_granges_list a GRangesList object storing exon genomic positions of extracted transcript

tx_intron_granges_list a GRangesList object storing intron genomic positions of extracted transcript

---

`prepare_transcript_se` *Prepare a transcript-level SummarizedExperiment object*

Description

Prepares a transcript-level SummarizedExperiment from TCC data using the EM algorithm

Usage

```
prepare_transcript_se(
  se_tcc,
  em.maxiter = 250,
  em.conv = 0.01,
  use_length_normalization = TRUE,
  ncpu = 1
)
```

Arguments

`se_tcc` A TCC SummarizedExperiment object returned by a function from the `Isosceles-package`

`em.maxiter` An integer scalar specifying the maximum number of EM iterations

`em.conv` A numeric scalar specifying the EM convergence threshold

`use_length_normalization`
A logical scalar specifying if normalization using effective transcript lengths should be used during EM

`ncpu` An integer scalar specifying the number of cores to use for multicore parallelization

Value

A SummarizedExperiment object containing transcript annotation and quantification data

Index