Package: ISoLDE (via r-universe)

October 11, 2026

Type Package

Title Integrative Statistics of alleLe Dependent Expression

Version 1.41.0

Date 2015-10-29

Description This package provides ISoLDE a new method for identifying imprinted genes. This method is dedicated to data arising from RNA sequencing technologies. The ISoLDE package implements original statistical methodology described in the publication below.

Encoding UTF-8

License GPL (>= 2.0)

Depends R (>= 3.3.0),graphics,grDevices,stats,utils

URL www.r-project.org

LazyData no

NeedsCompilation yes

Author Christelle Reynès [aut, cre], Marine Rohmer [aut], Guilhem Kister [aut]

Maintainer Christelle Reynès <christelle.reynes@igf.cnrs.fr>

biocViews ImmunoOncology, GeneExpression, Transcription, GeneSetEnrichment, Genetics, Sequencing, RNASeq, MultipleComparison, SNP, GeneticVariability, Epigenetics, MathematicalBiology, GeneRegulation

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 12:43:05 UTC

RemoteUrl https://github.com/bioc/ISoLDE

RemoteRef HEAD

RemoteSha d06bfdc444b0ab4671c4b0ab6ebe610fc875fc7c

Contents

ISoLDE-package 2
filteredASRcounts 3
filterT 4
isolde_test 6
normASRcounts 8
rawASRcounts 10
readNormInput 11
readRawInput 12
readTarget 13
target 14

Index **16**

`ISoLDE-package` *INTEGRATIVE STATISTICS OF ALLELE DEPENDENT EXPRESSION*

Description

This package provides a new method for identifying genes with allelic bias. This method is dedicated to data arising from RNA sequencing technologies. The ISoLDE package implements original statistical methodology described in the publication below.

Details

ISoLDE method has been motivated by several literature limitations in taking into account the data specificities and in making the most of biological replicates. It is based on the definition of a new criterion using robust estimation of the data variability. Variability estimation is of high importance in statistical testing procedures because a difference significance can only be assessed with regards to the intrinsic data variability.

Two methods are available to identify allele specific expression: one is based on bootstrap resampling while the second one uses an empirical threshold. The first one is much more satisfying and is likely to give the most reliable results but it can only be applied to data with at least three biological replicates for each reciprocal cross. While strongly recommending to use at least three replicates, the second method implements a robust solution when only two replicates are available.

Author(s)

Christelle Reynès <christelle.reynes@igf.cnrs.fr>,
Marine Rohmer <marine.rohmer@mgx.cnrs.fr>,
Guilhem Kister <guilhem.kister@umontpellier.fr>

References

Reynès, C. et al. (2016): ISoLDE: a new method for identification of allelic imbalance. *Submitted*

filteredASRcounts *NORMALIZED AND FILTERED ALLELE SPECIFIC READ (ASR) COUNTS FOR AN EXPERIENCE WITH MORE THAN TWO REPLICATES*

Description

A data.frame containing the normalized and filtered values of allele specific read counts, for an experiment with more than two replicates. This data.frame is obtained with the filterT function run on the normASRcounts data.frame.

Format

A data.frame.

Details

This data.frame is obtained with the filterT function. Each line represents a feature (e.g. a gene, transcript). Each column represents the number of allele specific sens reads from either the paternal or maternal parent for a given biological replicate, so that you expect to have two columns per biological replicate. Values in the matrix are filtered and normalized (RLE method) ASR counts.

Source

Extract from Bouschet, T. et al. (2016): In vitro corticogenesis from embryonic stem cells recapitulates the in vivo epigenetic control of imprinted gene expression. *Submitted* Subset of 6062 genes (after filtering).

References

Bouschet, T. et al. (2016): In vitro corticogenesis from embryonic stem cells recapitulates the in vivo epigenetic control of imprinted gene expression. *Submitted*

See Also

filterT: a function to filter the ASR counts and produce the filteredASRcounts data.frame.
isolde_test: example of which uses filteredASRcounts.

Examples

```
data(filteredASRcounts)
```

filterT *FILTERING GENES BEFORE STATISTICAL ANALYSIS*

Description

Filter lowly expressed genes (or transcripts) according to a data driven threshold, before any statistical analysis. This step is not mandatory but strongly recommended.

Usage

```
filterT(rawASRcounts, normASRcounts, target, tol_filter = 0,
bias)
```

Arguments

rawASRcounts	the data.frame containing raw counts (obtained with the readRawInput function or any data.frame following rawASRcounts format specifications). Raw count data.frame is required when filtering on raw or on normalized data when the normalized data do not contain 0 counts. (For simplicity purpose, we call '0 count' any value of zero in a count file).
normASRcounts	the data.frame containing normalized counts (obtained with the readNormInput function or any data.frame following normASRcounts format specifications). We strongly recommend to filter on normalized ASR counts.
target	the data.frame containing the target meta data (obtained with the readTarget function or any data.frame following target format specifications).
tol_filter	a value between 0 and 100 allowing to introduce tolerance rate into filtering step: if tol_filter = 25 all genes having less than 25% of their counts from at least one parental (or strain) origin below the threshold are selected (the default value 0 means all raw counts from at least one parental (or strain) origin must be above threshold, 100 means that no filtering is applied).
bias	The kind of allele expression bias you want to study. It must be one of "parental" or "strain".

Details

Filtering in statistical analysis is recommended to avoid considering genes (or transcript) without enough information, and thus to avoid a too strong effect of multiple test correction.

The aim of our filtering method is to eliminate from analysis not enough quantified genes, that is genes having mostly counts of 0 or near 0 for each replicate in at least one condition (parent, strain). In this purpose, the filterT function searches for the distribution of counts of a gene in a condition when most of read counts are 0 for this condition. This distribution allows to define a threshold. Hence, genes having less counts than this threshold are eliminated.

The filtering step is not mandatory but strongly recommended.

Value

A list of two `data.frame`:

`filteredASRcounts`
: This `data.frame` contains ASR counts that have successfully passed the filtering step.

`removedASRcounts`
: This `data.frame` contains ASR counts that have been removed by the filtering step.

Each line represents a feature (e.g. a gene, transcript). Each column represents the number of allele-specific sens reads from either the paternal or maternal parent for a given biological replicate, so that you expect to have two columns per biological replicate.

Note

`filterT` output on normalized data is the typical input for `isolde_test`.

Note

A minimal filtering step will always be performed while applying the `isolde_test` function. It consists of eliminating all genes not satisfying these two conditions:
- At least one of the two medians (of paternal or maternal ASR counts) is different from 0;
- There is at least one ASR count (different from 0) in each cross.

Author(s)

Marine Rohmer <marine.rohmer@mgx.cnrs.fr>,
Christelle Reynès <christelle.reynes@igf.cnrs.fr>

References

Reynès, C. et al. (2016): ISoLDE: a new method for identification of allelic imbalance. *Submitted*

Examples

```
# Loading all required data.frames
data(rawASRcounts)
data(normASRcounts)
data(target)

# Filtering genes from the ASR count data.frame in parental bias study
res_filterT <- filterT(rawASRcounts = rawASRcounts,
                       normASRcounts = normASRcounts,
                       target = target, bias="parental")
filteredASRcounts <- res_filterT$filteredASRcounts
removedASRcounts <- res_filterT$removedASRcounts
```

isolde_test *Statistical analysis of Allele specific read (ASR) counts*

Description

The main function of the ISoLDE package. Performs statistical test to identify genes with allelic bias and produces both graphical and textual outputs.

Usage

```
isolde_test(bias, method = "default", asr_counts, target,
            nboot = 5000, pcore = 75, graph = TRUE, ext = "pdf",
            text = TRUE, split_files = FALSE, prefix =
            "ISoLDE_result", outdir = "")
```

Arguments

bias	The kind of bias you want to study. It must be one of "parental" or "strain".
method	specifies the statistical method to use for testing. It must be one of "default" or "threshold". Default behaviour is to adapt to the number of replicates: when at least three biological replicates for each reciprocal cross are available the bootstrap resampling method is used, else the threshold method is applied. It is possible to force isolde_test to use the threshold method even when more than three replicates are available. In this case method must be set to "threshold". It is *not possible* to force a bootstrap method with less than three replicates.
asr_counts	the data.frame containing the ASR counts to be tested. These data should be normalized and filtered (see the filterT function), although the function can run with non-normalized and non-filtered data (not recommended).
target	the target data.frame (obtained by the readTarget function).
nboot	specifies how many resampling steps to do for the bootstrap method. This option is not considered if "threshold" value is set for method. Low values of nboot leads to less relevent results (default to 5000).
pcore	a value between 0 and 100 (default to 75) which specifies the proportion of cores (in percent) to be used for the bootstrap method.
graph	if TRUE (default) graphical outputs are produced (both on device and file).
ext	specifies the extension of the graphical file output (does not work if graph = FALSE). It must be one of "pdf" (default), "png" or "eps".
text	if TRUE (default) textual output files are produced.
split_files	if text = TRUE, specifies if your want to have all genes in one same output file (FALSE, default) or four separate files according to the result: ASE, biallelic, undetermined or filtered (TRUE).
prefix	specifies the prefix for all output file names (default to "ISoLDE_result").
outdir	specifies the path where to write the output file(s) (default to current directory).

Details

Before using this function, your data should be normalized and filtered (see the `filterT` function for filtering) although the function can run with non-normalized and/or non-filtered data.

The method depends on your minimum number of replicates for each reciprocal cross.

If only one replicate is found, the test can not be achieved and exits.

method="default" : If more than two replicates per cross, the method takes advantage of having enough information by using bootstrap resampling to identify genes with allelic bias.

If only two replicates are found in at least one cross, there is too few information to obtain reliable distributions from resampling. Genes with allelic bias are identified thanks to empirically defined thresholds.

method="threshold" : The empirical method will be processed instead of the bootstrap one, even if more than two replicates per cross are found.

Note that in differential RNA-seq analysis, at least three replicates are strongly recommended, as variability estimation quality is a key factor in statistical analysis.

More details in Reynès, C. et al. (2016) ISoLDE: a new method for identification of allelic imbalance. *Submitted*

Value

`listASE` a `data.frame` with one row per gene (or transcript) identified as having an allelic bias and five columns:
- "names" contains gene (or transcript) names such as `asr_counts` row names,
- "criterion" contains the criterion value (see vignette or Reynès et al. (2016)),
- "diff_prop" the criterion numerator which contains the difference between proportions of either parents or strain origins,
- "variability" the criterion denominator which quantifies the gene (or transcript) variability between replicates,
- "origin" specifies the bias direction either "P" or "M" for parental bias or one of specified strain names for strain bias.

`listBA` a `data.frame` with one row per gene (or transcript) identified as biallelically expressed and four columns corresponding to the first four ones in listASE.

`listUN` a `data.frame` with one row per gene (or transcript) with undetermined status and six columns. The first five columns are the same as `listASE`, the last one may take three values:
- "FLAG_consistency" for genes no statistical evidence of neither bias nor biallelic expression but whose parental or strain bias is always in the same direction across replicates,
- "FLAG_significance" for genes with statistical evidence of bias but with dis-

crepancies in bias direction across replicates,
- "NO_FLAG" for other undetermined genes.

`listFILT` a `data.frame` containing names of genes that have failed the minimal filtering step and thus that have not been considered during the statistical test.

ASE, BA and UN lists are sorted according to their criterion value.

Note

The bootstrap resampling step is performed many times (default to 5000). Hence, the function may run for a long time if performing the bootstrap method (until several minutes).

Note

A minimal filtering step will always be performed while applying the `isolde_test` function. It consists of eliminating all genes not satisfying these two conditions:
- At least one of the two medians (of paternal or maternal ASR counts) is different from 0;
- There is at least one ASR count (different from 0) in each cross.

Author(s)

Christelle Reynès <christelle.reynes@igf.cnrs.fr>,
Marine Rohmer <marine.rohmer@mgx.cnrs.fr>

References

Reynès, C. et al. (2016): ISoLDE: a new method for identification of allelic imbalance. *Submitted*

Examples

```
  # Loading all required data.frames
  data(filteredASRcounts)
  data(target)
  # Statistical analysis (forcing the threshold option)
  isolde_res <- isolde_test(bias = "parental", method = "threshold",
asr_counts = filteredASRcounts, target = target, ext = "pdf",
prefix = "ISoLDE_test")
```

---

normASRcounts *NORMALIZED ALLELE SPECIFIC READ (ASR) COUNTS FOR AN EXPERIENCE WITH MORE THAN TWO REPLICATES*

Description

normASRcounts_file.txt: A tab-delimited text file containing the normalized values of ASR counts for an experiment with more than two replicates.
normASRcounts.rda: the normASRcounts_file.txt loaded into a data.frame by the `readNormInput` function.

Format

normASRcounts_file.txt: A tab-delimited file.
normASRcounts.rda: A data.frame.

Details

Each line represents a feature (e.g. a gene or a transcript).
Each column represents the number of allele-specific sens reads from either
the paternal or maternal parent for a given biological replicate, so that
you expect to have two columns per biological replicate.
Values in the matrix are normalized (RLE method) ASR counts.
In case of double input, columns must be in the same order in both raw and normalized ASR counts files.
The normASRcounts_file.txt file should be read and checked by the `readNormInput` function.

Note

A minimum of two biological replicates per cross is mandatory, however, we strongly recommend to use more than two replicates per cross. This enables a better estimation of variability and to use the bootstrap method to perform the statistical test (see the `isolde_test` function).

Source

Extract from Bouschet, T. et al. (2016): In vitro corticogenesis from embryonic stem cells recapitulates the in vivo epigenetic control of imprinted gene expression. *Submitted* Subset of 6062 genes (after filtering).

References

Bouschet, T. et al. (2016): In vitro corticogenesis from embryonic stem cells recapitulates the in vivo epigenetic control of imprinted gene expression. *Submitted*

See Also

`readNormInput` example of which uses the `normASRcounts` file.

Examples

```
# normASRcounts_file.txt
normfile <- system.file("extdata", "normASRcounts_file.txt",
package = "ISoLDE")
normASRcounts <- readNormInput(norm_file = normfile, del = "tab",
rownames = TRUE, colnames = TRUE)

# normASRcounts.rda
data(normASRcounts)
```

rawASRcounts *RAW ALLELE SPECIFIC READ (ASR) COUNTS FOR AN EXPERIENCE WITH MORE THAN TWO REPLICATES*

Description

rawASRcounts_file.txt: A tab-delimited text file containing the raw values of ASR counts for an experiment with more than two replicates.
rawASRcounts.rda: the rawASRcounts_file.txt loaded into a data.frame by the `readRawInput` function.

Format

rawASRcounts_file.txt: A tab-delimited file.
rawASRcounts.rda: A data.frame.

Details

Each line represents a feature (e.g. a gene or a transcript).
Each column represents the number of allele-specific sens reads from either
the paternal or maternal parent for a given biological replicate, so that
you expect to have two columns per biological replicate.
Values in the matrix are raw allele-specific read counts.
In case of double input, columns must be in the same order in both raw and normalized ASR counts files.
The rawASRcounts_file.txt file should be read and checked by the `readRawInput` function.

Note

A minimum of two biological replicates per cross is mandatory, however, we strongly recommend to use more than two replicates per cross. This enables a better estimation of variability and to use the bootstrap method to perform the statistical test (see the `isolde_test` function).

Source

Extract from Bouschet, T. et al. (2016): In vitro corticogenesis from embryonic stem cells recapitulates the in vivo epigenetic control of imprinted gene expression. *Submitted* Subset of 6062 genes (after filtering).

References

Bouschet, T. et al. (2016): In vitro corticogenesis from embryonic stem cells recapitulates the in vivo epigenetic control of imprinted gene expression. *Submitted*

See Also

`readRawInput` example of which uses the `rawASRcounts` file.

Examples

```
# rawASRcounts_file.txt
rawfile <- system.file("extdata", "rawASRcounts_file.txt",
package = "ISoLDE")
rawASRcounts <- readRawInput(raw_file = rawfile, del = "tab",
colnames = TRUE, rownames = TRUE)

# rawASRcounts.rda
data(rawASRcounts)
```

`readNormInput` *READ THE NORMALIZED DATA FILE*

Description

Checks and loads into a `data.frame` the input file containing normalized allele-specific read (ASR) counts so that it can be input into `filterT` and `isolde_test`.

Usage

```
readNormInput(norm_file, del = "\t", rownames = TRUE, colnames =
TRUE, dec = ".")
```

Arguments

`norm_file`	A character-delimited input file containing normalized counts such as described in `normASRcounts_file`.
`del`	Specifies the delimiter for the input file, usually a semi-colon ";", a coma "," or a tabulation "\t". (default : "\t"). Note : None of your data values must contain this delimiter (be specially careful in gene names).
`rownames`	Specifies if the file contains some row names to consider. Possible values: TRUE or FALSE (default: TRUE).
`colnames`	Specifies if the file contains some column names to consider. Possible values: TRUE or FALSE (default: TRUE).
`dec`	Specifies the character used in the file for decimal mark (default : ".").

Value

A `data.frame` containing normalized ASR counts from your input file.

Author(s)

Marine Rohmer <marine.rohmer@mgx.cnrs.fr>,
Christelle Reynès <christelle.reynes@igf.cnrs.fr>

See Also

normASRcounts_file.txt: the normalized ASR count file on which to run the readNormInput function.
readRawInput: a similar function for raw (non-normalized) ASR count files.

Examples

```
# character-delimited input file containing normalized ASR counts
normfile <- system.file("extdata", "normASRcounts_file.txt",
package = "ISoLDE")
# loading it into a data.frame using the readNormInput function
nbreadnorm <- readNormInput(norm_file = normfile, del = "tab",
rownames = TRUE, colnames = TRUE, dec = ".")
```

---

readRawInput *READ THE RAW DATA FILE*

Description

Checks and loads into a data.frame the input file containing raw allele specific read (ASR) counts so that it can be input into filterT.

Usage

```
readRawInput(raw_file, del = "\t", rownames = TRUE, colnames =
    TRUE)
```

Arguments

raw_file	A character-delimited input file containing raw ASR counts such as described in rawASRcounts_file.
del	Specifies the delimiter for the input file, usually a semi-colon ";", a coma "," or a tabulation "\t". (default : "\t"). Note : None of your data values must contain this delimiter (be specially careful in gene names).
rownames	Specifies if the file contains some row names to consider. Possible values: TRUE or FALSE (default: TRUE).
colnames	Specifies if the file contains some column names to consider. Possible values: TRUE or FALSE (default: TRUE).

Details

Raw ASR counts are only required for the filtering step (with the filterT function) in case the normalized data do not contain 0 counts anymore. If you do not want to perform the filtering step or if you still have 0 counts in your normalized file, you do not need to load raw ASR counts. (For simplicity purpose, we call '0 count' any value of zero in a count file).

Value

A `data.frame` containing raw ASR counts from your input file.

Author(s)

Marine Rohmer <marine.rohmer@mgx.cnrs.fr>,
Christelle Reynès <christelle.reynes@igf.cnrs.fr>

See Also

`rawASRcounts_file.txt`: the raw ASR count file on which to run the `readRawInput` function.
`readNormInput`: a similar function for normalized ASR count file.

Examples

```
# character-delimited input file containing raw ASR counts
rawfile <- system.file("extdata", "rawASRcounts_file.txt",
package = "ISoLDE")
# loading it into a data.frame using the readRawInput function
nbread <- readRawInput(raw_file = rawfile, del = "tab",
rownames = TRUE, colnames = TRUE)
```

---

readTarget *READ THE TARGET FILE*

Description

Checks and loads into a `data.frame` your target input file.

Usage

```
readTarget(target_file, asr_counts, del = "\t")
```

Arguments

`target_file`	A character-delimited text input file, containing metadata about ASR counts files (see `target_file.txt`).
`asr_counts`	The `data.frame` containing values of ASR counts (obtained either by the `readRawInput` or the `readNormInput` function). It is used to perform checks on compatibility with the target file.
`del`	Specifies the delimiter for the target input file, usually a semi-colon ";", a coma "," or a tabulation "\t". (default : "\t"). Note : None of your data values must contain this delimiter (be specially careful in gene names).

Details

See `target_file.txt` for more details about the `target_file` format.

Value

a data.frame containing the target.

Author(s)

Marine Rohmer <marine.rohmer@mgx.cnrs.fr>,
Christelle Reynès <christelle.reynes@igf.cnrs.fr>

See Also

target_file.txt: the metadata file on which to run the readTarget function.

Examples

```
# Target input file
targetfile <- system.file("extdata", "target_file.txt",
package = "ISoLDE")
# The data.frame containing ASR counts is also required
data(rawASRcounts)
# Load into a data.frame and check the target file
target <- readTarget(target_file = targetfile,
asr_counts = rawASRcounts, del = "\t")
```

---

target *METADATA ABOUT THE ASR COUNT DATA.*

Description

target_file.txt: A tab-delimited file describing your input data (raw and / or normalized allele specific read (ASR) count file(s)).
Each line of the target file corresponds to a column of the rawASRcounts and / or normASRcounts data.frames. Lines of target file MUST be in the same order as the columns in ASR count data. Each line contains four values, separated by a character (e.g. a tabulation) : samples, parent, strain and replicate (see the Details section for more information).
target.rda: The target_file.txt file loaded into a data.frame by the link{readTarget} function.

Format

target_file.txt: A tab-delimited text file.
target.rda: A data.drame.

Details

Details of the three columns :
sample : the biological sample name. A same sample name has to appear twice in the target file : one line for the maternal allele and one line for the paternal allele.
allele : the parental origin of the ASR count. Two possible values: maternal or paternal.
strain : the strain origin of the ASR count. Exactly two different values have to be provided in the

whole file.

The first line of the target file has to contain these column names in the same order.
These metadata are required for both `filterT` and `isolde_test` functions.

Factice example:
sample,parent,strain
samp1,maternal,str1
samp1,paternal,str2
samp2,maternal,str1
samp2,paternal,str2
samp3,maternal,str1
samp3,paternal,str2
samp4,maternal,str1
samp4,paternal,str2

Author(s)

Marine Rohmer <marine.rohmer@mgx.cnrs.fr>,
Christelle Reynès <christelle.reynes@igf.cnrs.fr>

See Also

`readTarget` is a function to load into a `data.frame` and check the input target file.

Index

∗ **datasets**
filteredASRcounts, 3
normASRcounts, 8
rawASRcounts, 10

filteredASRcounts, 3, *3*
filterT, *3*, 4, *4–7*, *11*, *12*, *15*

ISoLDE *(*ISoLDE-package*)*, 2
ISoLDE-package, 2
isolde_test, *3*, *5*, 6, *6*, *8–11*, *15*

normASRcounts, *3*, *4*, 8, *9*, *14*
normASRcounts_file, *11*
normASRcounts_file *(*normASRcounts*)*, 8
normASRcounts_file.txt, *12*

rawASRcounts, *4*, 10, *10*, *14*
rawASRcounts_file, *12*
rawASRcounts_file *(*rawASRcounts*)*, 10
rawASRcounts_file.txt, *13*
readNormInput, *4*, *8*, *9*, 11, *12*, *13*
readRawInput, *4*, *10*, 12, *12*, *13*
readTarget, *4*, *6*, 13, *14*

target, *4*, 14
target_file *(*target*)*, 14
target_file.txt, *13*, *14*