

# Package: LinTInd (via r-universe)

July 9, 2024

**Title** Lineage tracing by indels

**Version** 1.9.0

**Description** When we combine gene-editing technology and sequencing technology, we need to reconstruct a lineage tree from alleles generated and calculate the similarity between each pair of groups. FindIndel() and IndelForm() function will help you align each read to reference sequence and generate scar form strings respectively. IndelIdents() function will help you to define a scar form for each cell or read. IndelPlot() function will help you to visualize the distribution of deletion and insertion. TagProcess() function will help you to extract indels for each cell or read. TagDist() function will help you to calculate the similarity between each pair of groups across the indwells they contain. BuildTree() function will help you to reconstruct a tree. PlotTree() function will help you to visualize the tree.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**Suggests** knitr, rmarkdown

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**biocViews** SingleCell, CRISPR, Alignment

**Depends** R (>= 4.0), ggplot2, parallel, stats, S4Vectors

**Imports** data.tree, reshape2, networkD3, stringdist, purrr, ape, cowplot, ggnewscale, stringr, dplyr, rlist, pheatmap, Biostrings, pwalgn, IRanges, BiocGenerics(>= 0.36.1), ggtree

**Repository** <https://bioc.r-universe.dev>

**RemoteUrl** <https://github.com/bioc/LinTInd>

**RemoteRef** HEAD

**RemoteSha** fb7ef03630eaa2bd655219f0faa31486c6bc8486

**Contents**

<code>align_to_range</code> . . . . .	2
<code>BuildTree</code> . . . . .	3
<code>cellsinfo</code> . . . . .	4
<code>celltype</code> . . . . .	4
<code>change_form_stat</code> . . . . .	4
<code>cutsite</code> . . . . .	5
<code>data</code> . . . . .	5
<code>example_data</code> . . . . .	6
<code>FindIndel</code> . . . . .	6
<code>IndelForm</code> . . . . .	7
<code>IndelIdents</code> . . . . .	8
<code>IndelPlot</code> . . . . .	9
<code>PlotTree</code> . . . . .	9
<code>ReadCutsite</code> . . . . .	10
<code>ReadFasta</code> . . . . .	11
<code>ref</code> . . . . .	11
<code>scarform</code> . . . . .	12
<code>scarinfo</code> . . . . .	12
<code>scarref</code> . . . . .	12
<code>scarref_all</code> . . . . .	13
<code>tag</code> . . . . .	13
<code>TagDist</code> . . . . .	13
<code>TagProcess</code> . . . . .	14
<code>treeinfo</code> . . . . .	15
<b>Index</b>	<b>16</b>

---

<code>align_to_range</code>	<i>align_to_range</i>
-----------------------------	-----------------------

---

**Description**

Title

**Usage**`align_to_range(p, s, cut)`**Arguments**

<code>p</code>	A base sequence in character format
<code>s</code>	A base sequence in character format
<code>cut</code>	The distance between the starting sites of two fragments

**Value**

A list include two IRanges instances (deletion and insertion)

**Examples**

```
align_to_range(p="AAGG---AAATTCGGAATAAGGAATTT",s="AAGGCCCAAATTT-CGGAATAAGGAATTT",cut=0)
```

---

BuildTree

*BuildTree*

---

**Description**

Generate an array generant tree of a data.tree data structure and save it

**Usage**

```
BuildTree(tag)
```

**Arguments**

tag                    List generate from TagProcess, for more see [TagProcess](#)

**Details**

Title

**Value**

list with two elements, a data.tree data structure and a dataframe of array information for each cell barcode

**Examples**

```
data("example_data",package = "LinTInd")  
treeinfo<-BuildTree(tag)
```

---

cellsinfo	<i>example of cellsinfo</i>
-----------	-----------------------------

---

**Description**

This an example of cellsinfo

**Usage**

```
data("example_data")
```

**Format**

cellsinfo

---

celltype	<i>example of celltype</i>
----------	----------------------------

---

**Description**

This an example of celltype

**Usage**

```
data("example_data")
```

**Format**

celltype

---

change_form_stat	<i>change_form_stat</i>
------------------	-------------------------

---

**Description**

Title

**Usage**

```
change_form_stat(indel)
```

**Arguments**

indel	List include two IRanges instances, contain start and end site of deletions and inserstions
-------	---

**Value**

A scar form string

**Examples**

```
data("example_data", package = "LinTInd")
change_form_stat(cellsinfo$indel[[1]])
```

---

cutsite	<i>example of cutsite</i>
---------	---------------------------

---

**Description**

This an example of cutsite

**Usage**

```
data("example_data")
```

**Format**

cutsite

---

data	<i>example of data</i>
------	------------------------

---

**Description**

This an example of data

**Usage**

```
data("example_data")
```

**Format**

data

example\_data            *example\_data*

---

**Description**

A set of example data

**Usage**

```
data("example_data")
```

**Format**

The format is: chr "example\_data"

**Examples**

```
data("example_data", package = "LinTInd")
```

---

FindIndel            *FindIndel*

---

**Description**

This function can identify indels for each read in input data, and create IRanges instances for deletion and insertion

**Usage**

```
FindIndel(  
  data,  
  scarfull,  
  scar,  
  align_score = NULL,  
  type = NULL,  
  indel.coverage = NULL,  
  cln  
)
```

**Arguments**

data	data frame, include cell barcode, UMI and reads.
scarfull	DNASTring of reference sequence
scar	The cutsite data frame
align_score	The minimum alignment score that matched sequence should get, default in this parameter is the score that the reads which all of the target set were cutted got
type	Group name for this data ("None" in default)
indel.coverage	Choose indels selected scope: "Accurate" (default) means only the indles happened in target site will be idented; "All" means each indel will be detected even they locate on the anchors
cln	The number of threads

**Details**

Title

**Value**

list include IRanges instances (deletion and insertion), a data frame of reads' informations, reference sequenc, dataframe of cut sites

**Examples**

```
data("example_data", package = "LinTInd")
scarinfo<-FindIndel(data=data, scarfull=ref, scar=cutsite, indel.coverage="All", type="test", cln=1)
```

---

IndelForm

*IndelForm*

---

**Description**

Generate scar form strings from scarinfo list for each reads

**Usage**

```
IndelForm(scarinfo, cln)
```

**Arguments**

scarinfo	List generate from FindIndel, for more see <a href="#">FindIndel</a>
cln	The number of threads

**Details**

Title

**Value**

A new list of scarinfo, the scarform of each reads will add in the data frame of reads' informations

**Examples**

```
data("example_data", package = "LinTInd")
IndelForm(scarinfo, cln=1)
```

---

IndelIdents

*IndelIdents*


---

**Description**

Function to define a scarform for each cell(single cell) or each reads(bulk seq, generate 'cell barcode' for each reads)

**Usage**

```
IndelIdents(scarinfo, method.use = NULL, cln)
```

**Arguments**

scarinfo	List generate from IndelForm, for more see <a href="#">IndelForm</a>
method.use	Select how to determine a scar form string for each cell: "reads.num" (default):find the scar with the most reads in the cell; "umi.num":find the scar with the most UMIs in the cell; "consensus":find the consistent sequences in each cell, and then generate scar form strings from the new reads
cln	The number of threads

**Details**

Title

**Value**

The list generate from FindIndel, but in 'Scar' element a new column contain scar form strings

**Examples**

```
data("example_data", package = "LinTInd")
IndelIdents(scarinfo, method.use="umi.num", cln=1)
```



---

IndelPlot	<i>IndelPlot</i>
-----------	------------------

---

**Description**

Return 2 line charts, show the probability of insertion and deletion at each site

**Usage**

```
IndelPlot(cellsinfo)
```

**Arguments**

cellsinfo      List generate from IndelIdents, for more see [IndelIdents](#)

**Details**

Title

**Value**

2 line charts

**Examples**

```
data("example_data", package = "LinTInd")
IndelPlot(cellsinfo = cellsinfo)
```

---

PlotTree	<i>PlotTree</i>
----------	-----------------

---

**Description**

Function to visualise the array generate tree

**Usage**

```
PlotTree(treeinfo, data.extract = NULL, annotation = NULL, prefix = NULL)
```

**Arguments**

treeinfo      List generate from BuildTree, for more see [BuildTree](#)

data.extract    (optional) If "FALSE" (default), will not return the indel's information, if it's "TRUE", the opposite

annotation    (optional) If "TRUE" (default), the annotation of each cell barcodes have to be provided before, and a heatmap of cells' distribution for each array will be return

prefix        (optional) Indels' prefix

**Details**

Title

**Value**

A list include a ggplot object, a dataframe show the distribution of each array contained in each group of cells (optional), and a dataframe to create the ggplot object

**Examples**

```
data("example_data", package = "LinTInd")
plotinfo<-PlotTree(treeinfo = treeinfo, data.extract = "TRUE", annotation = "TRUE")
plotinfo<-PlotTree(treeinfo = treeinfo, data.extract = "TRUE", annotation = "FALSE")
```

---

ReadCutsite

*ReadCutsite*

---

**Description**

Function to create a reference dataframe include each position and its' group

**Usage**

```
ReadCutsite(segref, reftype=NULL)
```

**Arguments**

segref	The cutsite file
reftype	Choose the reference type you want, if reftype="Accurate" (default), there will only the target sites be generated; if reftype="All", each site will be generated

**Details**

Title

**Value**

reference dataframe

**Examples**

```
data("example_data", package = "LinTInd")
ReadCutsite(cutsite)
ReadCutsite(cutsite, reftype="All")
```

---

ReadFasta

*ReadFasta*

---

**Description**

Function to read fasta file to DNString object

**Usage**

```
ReadFasta(filename)
```

**Arguments**

filename      The input fasta file name

**Details**

Title

**Value**

A DNString object

**Examples**

```
fafile=paste0(system.file("extdata",package = 'LinTInd'),"/V3.fasta")
ReadFasta(fafile)
```

---

ref

*example of ref*

---

**Description**

This an example of ref

**Usage**

```
data("example_data")
```

**Format**

ref

---

scarform	<i>example of scarform</i>
----------	----------------------------

---

**Description**

This an example of scarform

**Usage**

```
data("example_data")
```

**Format**

scarform

---

scarinfo	<i>example of scarinfo</i>
----------	----------------------------

---

**Description**

This an example of scarinfo

**Usage**

```
data("example_data")
```

**Format**

scarinfo

---

scarref	<i>example of scarref</i>
---------	---------------------------

---

**Description**

This an example of scarref

**Usage**

```
data("example_data")
```

**Format**

scarref

---

scarref_all	<i>example of scarref_all</i>
-------------	-------------------------------

---

**Description**

This an example of scarref\_all

**Usage**

```
data("example_data")
```

**Format**

scarref\_all

---

tag	<i>example of tag</i>
-----	-----------------------

---

**Description**

This an example of tag

**Usage**

```
data("example_data")
```

**Format**

tag

---

TagDist	<i>TagDist</i>
---------	----------------

---

**Description**

If the cell barcode and the anntation of each cell are provided, this function can calculate the relationship between each cell type in three way

**Usage**

```
TagDist(tag, method = NULL)
```

**Arguments**

tag	List generate from TagProcess, for more see <a href="#">TagProcess</a>
method	Denote which method to use: <ul style="list-style-type: none"> <li>• "Jaccard"(default): calculate the weighted jaccard similarity of indels between each pair of groups;</li> <li>• "P": right-tailed test, compare the Indels intersection level with the hypothetical result generated from random editing, and the former is expected to be significantly higher than the latter;</li> <li>• "spearman": Spearman correlation of indels between each pair of groups</li> </ul>

**Details**

Title

**Value**

2 figures are saved to show the distribution of INDEL and the relationship between groups respectively, the matrix of the relationship between groups is returned

**Examples**

```
tag_dist=TagDist(tag,method = "spearman")
tag_dist=TagDist(tag)
tag_dist=TagDist(tag,method = "Jaccard")
tag_dist=TagDist(tag,method = "P")
tag_dist=TagDist(tag,method = "spearman")
```

---

TagProcess

*TagProcess*

---

**Description**

Split each indel from scar form strings and map indel information to cell barcodes

**Usage**

```
TagProcess(data, Cells = NULL, prefix = NULL)
```

**Arguments**

data	List generate from IndelIdents, for more see <a href="#">IndelIdents</a>
Cells	(optional) Dataframe of cells' annotation, with two columns: "Cell.BC" and "Cell.type"
prefix	(optional) Indels' prefix

**Details**

Title

**Value**

List with two dataframes: Indels for each cell barcode and cells' annotation

**Examples**

```
data("example_data", package = "LinTInd")
TagProcess(cellsinfo$info, Cells=celltype)
```

---

treeinfo

*example of treeinfo*

---

**Description**

This an example of treeinfo

**Usage**

```
data("example_data")
```

**Format**

treeinfo

# Index

## \* datasets

- cellsinfo, [4](#)
- celltype, [4](#)
- cutsite, [5](#)
- data, [5](#)
- example\_data, [6](#)
- ref, [11](#)
- scarform, [12](#)
- scarinfo, [12](#)
- scarref, [12](#)
- scarref\_all, [13](#)
- tag, [13](#)
- treeinfo, [15](#)

[align\\_to\\_range, 2](#)

[BuildTree, 3, 9](#)

[cellsinfo, 4](#)

[celltype, 4](#)

[change\\_form\\_stat, 4](#)

[cutsite, 5](#)

[data, 5](#)

[example\\_data, 6](#)

[FindIndel, 6, 7](#)

[IndelForm, 7, 8](#)

[IndelIdents, 8, 9, 14](#)

[IndelPlot, 9](#)

[PlotTree, 9](#)

[ReadCutsite, 10](#)

[ReadFasta, 11](#)

[ref, 11](#)

[scarform, 12](#)

[scarinfo, 12](#)

[scarref, 12](#)

[scarref\\_all, 13](#)

[tag, 13](#)

[TagDist, 13](#)

[TagProcess, 3, 14, 14](#)

[treeinfo, 15](#)