

# Package ‘TrustVDJ’

March 16, 2022

**Title** Tools for Immune Repertoire Analysis

**Version** 0.1.0

**Description** A toolkit for read and prepare immune repertoire data. 'TrustVDJ' package focuses on the reading and processing of 'TRUST4' and '10x cellranger' software output results by using 'ReadTrust' and 'Read10x' functions, respectively, and also provides a convenience function 'build\_IMGT\_reference' to download the 'IMGT' database reference and split its sequences by species.

**BugReports** <https://github.com/HatsuneCode/TrustVDJ/issues>

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Collate** 'constants.r' 'utils.r' 'class.r' 'download.r' 'read10x.r'  
'readTrust.r' 'refIMGT.r' 'test.r'

**Imports** Biostrings, data.table, methods, rvest, stats

**Suggests** R.utils

**NeedsCompilation** no

**Author** Lianhao Song [aut, cre] (<<https://orcid.org/0000-0002-3211-3726>>)

**Maintainer** Lianhao Song <[hatsunecode@gmail.com](mailto:hatsunecode@gmail.com)>

**Repository** CRAN

**Date/Publication** 2022-03-16 21:00:02 UTC

## R topics documented:

.Read10x_clonotype	2
.Read10x_consensus	3
.Read10x_contig	3
.ReadAIRR	4
.ReadTrust_BarcodeReport	4
.ReadTrust_Report	5
build_IMGT_reference	6
cbinds	6

consensus-class . . . . .	7
corTest . . . . .	8
df_chain . . . . .	8
Download . . . . .	9
Ifnone . . . . .	10
Ifnull . . . . .	10
Read10x . . . . .	11
ReadTrust . . . . .	12
show,consensus-method . . . . .	14
show,Trust-method . . . . .	14
timer . . . . .	15
Trust-class . . . . .	15
<b>Index</b>	<b>16</b>

---

<i>.Read10x_clonotype</i>	<i>Read 10x clonotype file</i>
---------------------------	--------------------------------

---

## Description

Read 10x clonotype file

## Usage

```
.Read10x_clonotype(clonotype_file, verbose = TRUE)
```

## Arguments

`clonotype_file` character. Path to clonotype file generated by 10x cellranger.  
`verbose` logical. Print progress. Default is TRUE

## Value

A VDJ information data.frame

## Examples

```
clonotype_file = system.file('extdata', '10x_clonotypes.csv.gz', package = 'TrustVDJ')
clonotype = .Read10x_clonotype(clonotype_file = clonotype_file, verbose = FALSE)
head(clonotype)
```

---

`.Read10x_consensus`     *Read 10x consensus file*

---

**Description**

Read 10x consensus file

**Usage**

```
.Read10x_consensus(consensus_file, verbose = TRUE)
```

**Arguments**

`consensus_file`    character. Path to consensus\_annotations file generated by 10x cellranger.  
`verbose`            logical. Print progress. Default is TRUE

**Value**

A VDJ information data.frame

**Examples**

```
consensus_file = system.file('extdata', '10x_consensus_annotations.csv.gz', package = 'TrustVDJ')  
consensus = .Read10x_consensus(consensus_file = consensus_file, verbose = FALSE)  
head(consensus)
```

---

`.Read10x_contig`        *Read 10x contig file*

---

**Description**

Read 10x contig file

**Usage**

```
.Read10x_contig(contig_file, verbose = TRUE)
```

**Arguments**

`contig_file`        character. Path to contig\_annotations file generated by 10x cellranger.  
`verbose`            logical. Print progress. Default is TRUE

**Value**

A VDJ information data.frame

**Examples**

```
contig_file = system.file('extdata', '10x_all_contig_annotations.csv.gz', package = 'TrustVDJ')
contig = .Read10x_contig(contig_file = contig_file, verbose = FALSE)
head(contig)
```

---

```
.ReadAIRR          Read AIRR file.
```

---

**Description**

`.ReadAIRR` reads an AIRR format file from TRUST4/cellranger results or somewhere else. It could be `xx_airr.tsv` or `xx_barcode_airr.tsv` generated by TRUST4 or `airr_rearrangement.tsv` generated by 10x cellranger (> 6.0). (.gz supported)

**Usage**

```
.ReadAIRR(airr_file = NULL, verbose = TRUE)
```

**Arguments**

`airr_file` character. Path to AIRR file, eg. `xx_airr.tsv` or `xx_barcode_airr.tsv`.  
`verbose` logical. Print progress. Default is TRUE

**Value**

A VDJ information data.frame

**Examples**

```
airr_file = system.file('extdata', 'TRUST4_airr.tsv.gz', package = 'TrustVDJ')
airr = .ReadAIRR(airr_file = airr_file, verbose = FALSE)
head(airr)
```

---

```
.ReadTrust_BarcodeReport
          Read TRUST4 barcode_report file.
```

---

**Description**

`.ReadTrust_BarcodeReport` reads a `barcode_report` file generated by TRUST4. Note that it could be `xx_barcode_report.tsv` but not `xx_report.tsv`. (.gz supported)

**Usage**

```
.ReadTrust_BarcodeReport(barcode_report_file = NULL, verbose = TRUE)
```

**Arguments**

barcode\_report\_file      character. Path to barcode\_report file generated by TRUST4.  
verbose                    logical. Print progress. Default is TRUE

**Value**

A VDJ information data.frame

**Examples**

```
barcode_report_file = system.file('extdata', 'TRUST4_barcode_report.tsv.gz', package = 'TrustVDJ')  
barcode_report = .ReadTrust_BarcodeReport(barcode_report_file, verbose = FALSE)  
head(barcode_report)
```

---

*.ReadTrust\_Report*      *Read TRUST4 report file.*

---

**Description**

*.ReadTrust\_Report* reads a report file generated by TRUST4. Note that it could be xx\_report.tsv but not xx\_barcode\_report.tsv. (.gz supported)

**Usage**

```
.ReadTrust_Report(report_file = NULL, verbose = TRUE)
```

**Arguments**

report\_file      character. Path to report file generated by TRUST4.  
verbose           logical. Print progress. Default is TRUE

**Value**

A VDJ information data.frame

**Examples**

```
report_file = system.file('extdata', 'TRUST4_report.tsv.gz', package = 'TrustVDJ')  
report = .ReadTrust_Report(report_file = report_file, verbose = FALSE)  
head(report)
```

---

build\_IMGT\_reference *Build IMGT database reference*

---

**Description**

Download reference sequences from the IMGT (the international ImMunoGeneTics information system, <http://www.imgt.org>) database and split the sequences by species.

**Usage**

```
build_IMGT_reference(outdir = NULL, verbose = TRUE)
```

**Arguments**

outdir	character. Default getwd()
verbose	logical. Default TRUE

**Value**

if success, return TRUE

**Examples**

```
build_IMGT_reference('IMGT_reference', verbose = FALSE)
```

---

cbinds *Combine Two Data-frame by Columns*

---

**Description**

Combine two data.frame by columns by filling in missing rows from each other based on rownames.

**Usage**

```
cbinds(F1, F2, fill = 0)
```

**Arguments**

F1	data.frame.
F2	data.frame.
fill	character/numeric. Default 0

**Value**

a combined data.frame

**Examples**

```
F1 = data.frame(A = seq(10), B = seq(10), row.names = seq(10))
F2 = data.frame(C = seq(5), D = seq(5), row.names = 3:7)
cbinds(F1, F2)
```

---

consensus-class	<i>The consensus class</i>
-----------------	----------------------------

---

**Description**

The consensus class

**Value**

An object of the consensus class

**Slots**

Vgene character. V gene, eg.: TRAV1\*01

Dgene character. D gene, eg.: TRAD1\*01

Jgene character. J gene, eg.: TRAJ1\*01

Cgene character. C gene, eg.: TRAC1\*01

CDR1dna character. CDR1 nucleic acid sequence, eg.: TCTGAACACAACCGC

CDR2dna character. CDR2 nucleic acid sequence, eg.: TTCCAGAATGAAGCTCAA

CDR3dna character. CDR3 nucleic acid sequence, eg.: TGTGCCAGCAGCCTACGCAACGAGCAGTACTTC

CDR3aa character. CDR3 amino acid sequence, eg.: CASSPTPGEATDTQYF

Count numeric. Read counts, eg.: 200

ID character. Consensus id, eg.: Sample1\_consensus1

CDR3germlineSimilarity numeric. CDR3 germline similarity score, eg.: 80

FullLength logical. Whether the vdj gene is complete, eg.: TRUE

---

corTest	<i>Test for Correlation</i>
---------	-----------------------------

---

**Description**

Correlation analysis for each row (each to each) between two data-frames.

**Usage**

```
corTest(x, y, method = "both", adj_method = "BH", rm0 = TRUE)
```

**Arguments**

x	data.frame.
y	data.frame.
method	character. 'pearson', 'spearman' or 'both'. Default 'both'
adj_method	character. choose one method in p.adjust.methods. Default 'BH'
rm0	logical. whether remove 0 in each analyse. Default TRUE.

**Value**

a correlation results data.frame

**Examples**

```
treatment = data.frame(S1 = sample(10, 5), S2 = sample(10, 5), S3 = sample(10, 5))
control   = data.frame(S4 = sample(20, 5), S5 = sample(20, 5), S6 = sample(10, 5))
result    = corTest(treatment, control, method = 'pearson')
head(result)
```

---

df_chain	<i>data.frame a single chain information</i>
----------	--

---

**Description**

data.frame a single chain information

**Usage**

```
df_chain(chain)
```

**Arguments**

chain	list. trust4 single chain information in a list
-------	---



**Value**

a data.frame named by chainName

**Examples**

```
df_chain(list('V', 'D', 'J', 'C', 'CDR3nt', 'CDR3aa', '60', 'id1', '98', '1'))
```

---

 Download

*Download files retryable*


---

**Description**

Download files retryable

**Usage**

```
Download(
  URLs,
  names = NULL,
  method = NULL,
  sleep = NULL,
  outdir = NULL,
  verbose = TRUE
)
```

**Arguments**

URLs	character/list. URLs to be downloaded.
names	character/list. file names. Default seq(URLs)
method	character. Method to be used for downloading files, equal to download.file. Default 'libcurl'
sleep	numeric. retry interval (second). Default 2
outdir	character. output directory.
verbose	logical. Default TRUE

**Value**

if success, return TRUE

**Examples**

```
URLs = paste0('http://www.imgt.org/download/', c('V-QUEST/IMGT_V-QUEST_reference_directory',
  'GENE-DB/IMGTGENEDB-ReferenceSequences.fasta-nt-WithGaps-F+ORF+inframeP'))
Download(URLs, c('vdj_species.html', 'IMGT_download.fa'))
```

---

Ifnone	<i>Default for NULL and NA value</i>
--------	--------------------------------------

---

**Description**

set default value for object, including NULL and NA and length 0.

**Usage**

```
x %|||% y
```

**Arguments**

x	character/numeric/factor/list. An object which could be checked by <code>is.na()</code> .
y	ANY. A default value

**Value**

%||%: x unless NULL, NA nor `length(x) == 0`, otherwise y

**Examples**

```
1 %|||% 1
NA %|||% 1
NULL %|||% 1
```

---

Ifnull	<i>Default for NULL value</i>
--------	-------------------------------

---

**Description**

set default value for object, equal to %||% in rlang package

**Usage**

```
x %||% y
```

**Arguments**

x	ANY. An object
y	ANY. A default value

**Value**

%||%: x unless NULL, otherwise y

**Examples**

```

1    %||% 1
NA   %||% 1
NULL %||% 1

```

---

Read10x

*Read AIRR/10x report files*


---

**Description**

Read10x reads AIRR file and/or contig/consensus/clonotype file generated by 10x cellranger (> 6.0). Generally AIRR file: `airr_rearrangement.tsv` (from cellranger); contig files: `all_contig_annotations.csv`, `filtered_contig_annotations.csv`; consensus file: `consensus_annotations.csv`; clonotype file: `clonotypes.csv`. (.gz supported)

1. AIRR + filtered\_contig: Read10x will read AIRR and add it 'fwr.., cdr.. and full\_length' column based on filtered\_contig file.
2. only one file: Read10x will return a data.frame for this file.
3. AIRR/contig + consensus/clonotype: Read10x will ignore consensus/clonotype file when either AIRR or contig file is given. Note that when AIRR + all\_contig, only contigs in AIRR will be return.
4. consensus + clonotype (no AIRR nor contig): Read10x will ignore clonotype file when consensus file is given. (Don't worry about the information of inkt/mait\_evidence in clonotype, these can be reproduced in downstream analysis.)

**Usage**

```

Read10x(
  airr_file = NULL,
  contig_file = NULL,
  consensus_file = NULL,
  clonotype_file = NULL,
  verbose = TRUE
)

```

**Arguments**

`airr_file` character. Path to AIRR file.

`contig_file` character. Path to contig\_annotations file generated by 10x cellranger.

`consensus_file` character. Path to consensus\_annotations file generated by 10x cellranger.

`clonotype_file` character. Path to clonotypes file generated by 10x cellranger.

`verbose` logical. Print progress. Default is TRUE.

**Value**

A VDJ information data.frame

**Examples**

```
# file paths
airr_file = system.file('extdata', '10x_airr_rearrangement.tsv.gz', package = 'TrustVDJ')
contig_file = system.file('extdata', '10x_all_contig_annotations.csv.gz', package = 'TrustVDJ')
#or contig_file =
#system.file('extdata', '10x_filtered_contig_annotations.csv.gz', package = 'TrustVDJ')
consensus_file = system.file('extdata', '10x_consensus_annotations.csv.gz', package = 'TrustVDJ')
clonotype_file = system.file('extdata', '10x_clonotypes.csv.gz', package = 'TrustVDJ')

# both AIRR and contig
data = Read10x(airr_file = airr_file, contig_file = contig_file)
head(data)

# only AIRR
data = Read10x(airr_file = airr_file)
head(data)

# only contig
data = Read10x(contig_file = contig_file)
head(data)

# only consensus
data = Read10x(consensus_file = consensus_file)
head(data)

# only clonotype
data = Read10x(clonotype_file = clonotype_file)
head(data)
```

---

ReadTrust

*Read AIRR/TRUST4 report files*

---

**Description**

ReadTrust reads AIRR file and/or barcode\_report/report file generated by TRUST4. Generally AIRR file: airr.tsv, barcode\_airr.tsv (from TRUST4); barcode\_report file: barcode\_report.tsv; report file: report.tsv. (.gz supported)

1. AIRR + barcode\_report: ReadTrust will read AIRR and add it a 'cdr3\_germline\_similarity' column based on barcode\_report.
2. only one file: ReadTrust will return a data.frame for this file.
3. AIRR/barcode\_report + report: ReadTrust will ignore report file when either AIRR or barcode\_report file is given.

**Usage**

```
ReadTrust(  
  airr_file = NULL,  
  barcode_report_file = NULL,  
  report_file = NULL,  
  verbose = TRUE  
)
```

**Arguments**

`airr_file` character. Path to AIRR file.

`barcode_report_file` character. Path to barcode\_report file generated by TRUST4.

`report_file` character. Path to report file generated by TRUST4.

`verbose` logical. Print progress. Default is TRUE.

**Value**

A VDJ information data.frame

**Examples**

```
# file paths  
airr_file = system.file('extdata', 'TRUST4_airr.tsv.gz', package = 'TrustVDJ')  
barcode_report_file = system.file('extdata', 'TRUST4_barcode_report.tsv.gz', package = 'TrustVDJ')  
report_file = system.file('extdata', 'TRUST4_report.tsv.gz', package = 'TrustVDJ')  
  
# both AIRR and barcode_report  
data = ReadTrust(airr_file = airr_file, barcode_report_file = barcode_report_file)  
head(data)  
  
# only AIRR  
data = ReadTrust(airr_file = airr_file)  
head(data)  
  
# only barcode_report  
data = ReadTrust(barcode_report_file = barcode_report_file)  
head(data)  
  
# only report  
data = ReadTrust(report_file = report_file)  
head(data)
```

---

show, consensus-method *Overview of the consensus class*

---

**Description**

Overview of the consensus class

**Usage**

```
## S4 method for signature 'consensus'  
show(object)
```

**Arguments**

object	class.
consensus	class. An object of the consensus class

**Value**

Brief information about an consensus object

---

show, Trust-method *Overview of the Trust class*

---

**Description**

Overview of the Trust class

**Usage**

```
## S4 method for signature 'Trust'  
show(object)
```

**Arguments**

object	class.
Trust	class. An object of the Trust class

**Value**

Brief information about an consensus object

---

timer	<i>Time record</i>
-------	--------------------

---

**Description**

Time record

**Usage**

timer()

**Value**

character. Time now

**Examples**

timer()

---

Trust-class	<i>The Trust class</i>
-------------	------------------------

---

**Description**

The Trust object is the center of each single-cell immune repertoire analysis. slots are listed below:

**Value**

An object of the trust4 class

**Slots**

barcode character. Cell barcode in single-cell sequencing, eg: Sample1\_ATGCCAGAACGACT.

celltype character. Inferred cell type, such as: abT, gdT or B.

Achain consensus. confident TCR/BCR Alpha-chain object.

Bchain consensus. confident TCR/BCR Beta-chain object.

Achain2 list. secondary TCR/BCR Alpha-chain objects.

Bchain2 list. secondary TCR/BCR Beta-chain objects.

# Index

[.Read10x\\_clonotype](#), [2](#)  
[.Read10x\\_consensus](#), [3](#)  
[.Read10x\\_contig](#), [3](#)  
[.ReadAIRR](#), [4](#)  
[.ReadTrust\\_BarcodeReport](#), [4](#)  
[.ReadTrust\\_Report](#), [5](#)

[build\\_IMGT\\_reference](#), [6](#)

[cbinds](#), [6](#)  
[consensus \(consensus-class\)](#), [7](#)  
[consensus-class](#), [7](#)  
[corTest](#), [8](#)

[df\\_chain](#), [8](#)  
[Download](#), [9](#)

[Ifnone](#), [10](#)  
[Ifnull](#), [10](#)

[Read10x](#), [11](#)  
[ReadTrust](#), [12](#)

[show, consensus-method](#), [14](#)  
[show, Trust-method](#), [14](#)

[timer](#), [15](#)  
[Trust \(Trust-class\)](#), [15](#)  
[Trust-class](#), [15](#)