

# Package ‘airr’

May 27, 2020

**Type** Package

**Version** 1.3.0

**Date** 2020-05-26

**Title** AIRR Data Representation Reference Library

**Description** Schema definitions and read, write and validation tools for data formatted in accordance with the AIRR Data Representation schemas defined by the AIRR Community <<http://docs.airr-community.org>>.

**License** CC BY 4.0

**URL** <http://docs.airr-community.org>

**BugReports** <https://github.com/airr-community/airr-standards/issues>

**LazyData** true

**BuildVignettes** true

**VignetteBuilder** knitr

**Encoding** UTF-8

**Depends** R (>= 3.1.2)

**Imports** methods, readr, stats, stringi, yaml

**Suggests** knitr, rmarkdown, testthat

**RoxygenNote** 7.1.0

**NeedsCompilation** no

**Author** Jason Vander Heiden [aut, cre],  
Susanna Marquez [aut],  
Scott Christley [aut],  
AIRR Community [cph]

**Maintainer** Jason Vander Heiden <[jason.vanderheiden@gmail.com](mailto:jason.vanderheiden@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-05-27 05:30:18 UTC

## R topics documented:

ExampleData . . . . .	2
load_schema . . . . .	2
read_airr . . . . .	3
Schema-class . . . . .	4
validate_airr . . . . .	6
write_airr . . . . .	6
<b>Index</b>	<b>8</b>

---

ExampleData	<i>Example AIRR data</i>
-------------	--------------------------

---

### Description

Example data files compliant with the the AIRR Data Representation standards.

### Format

extdata/rearrangement-example.tsv.gz: Rearrangement TSV file.

### Examples

```
# Get path to the rearrangement-example file
file <- system.file("extdata", "rearrangement-example.tsv.gz", package="airr")

# Load data file
df <- read_rearrangement(file)
```

---

load_schema	<i>Load a schema definition</i>
-------------	---------------------------------

---

### Description

load\_schema loads an AIRR object definition from the internal definition set.

### Usage

```
load_schema(definition)
```

### Arguments

definition      name of the schema definition.

**Details**

Valid definitions include:

- "Rearrangement"
- "Alignment"
- "Study"
- "Subject"
- "Diagnosis"
- "Sample"
- "CellProcessing"
- "NucleicAcidProcessing"
- "RawSequenceData"
- "SoftwareProcessing"

**Value**

A [Schema](#) object for the definition.

**See Also**

See [Schema](#) for the return object.

**Examples**

```
# Load the Rearrangement definition
schema <- load_schema("Rearrangement")

# Load the Alignment definition
schema <- load_schema("Alignment")
```

---

read_airr	<i>Read an AIRR TSV</i>
-----------	-------------------------

---

**Description**

read\_airr reads a TSV containing AIRR records.

**Usage**

```
read_airr(file, base = c("1", "0"), schema = RearrangementSchema, ...)

read_rearrangement(file, base = c("1", "0"), ...)

read_alignment(file, base = c("1", "0"), ...)
```

**Arguments**

file	input file path.
base	starting index for positional fields in the input file. If "1", then these fields will not be modified. If "0", then fields ending in "_start" and "_end" are 0-based half-open intervals (python style) in the input file and will be converted to 1-based closed-intervals (R style).
schema	Schema object defining the output format.
...	additional arguments to pass to <a href="#">read_delim</a> .

**Details**

`read_rearrangement` reads an AIRR TSV containing Rearrangement data.

`read_alignment` reads an AIRR TSV containing Alignment data.

**Value**

A data.frame of the TSV file with appropriate type and position conversion for fields defined in the specification.

**See Also**

See [Schema](#) for the AIRR schema object definition. See [write\\_airr](#) for writing AIRR data.

**Examples**

```
# Get path to the rearrangement-example file
file <- system.file("extdata", "rearrangement-example.tsv.gz", package="airr")

# Load data file
df <- read_rearrangement(file)
```

---

Schema-class

*S4 class defining an AIRR standard schema*

---

**Description**

Schema defines a common data structure for AIRR Data Representation standards.

**Usage**

```
## S4 method for signature 'Schema'
names(x)

## S4 method for signature 'Schema,character'
x[i]
```

```
## S4 method for signature 'Schema'  
x$name  
  
AlignmentSchema  
  
RearrangementSchema
```

### Arguments

x	Schema object.
i	field name.
name	field name.

### Format

A Schema object.  
An object of class Schema of length 1.  
An object of class Schema of length 1.

### Details

The following predefined Schema objects are defined:  
AlignmentSchema: AIRR Alignment Schema.  
RearrangementSchema: AIRR Rearrangement Schema.

### Slots

required character vector of required fields.  
optional character vector of non-required fields.  
properties list of field definitions.  
info list schema information.

### See Also

See [load\\_schema](#) for loading a Schema from the definition set. See [read\\_airr](#), [write\\_airr](#) and [validate\\_airr](#) schema operators.

---

validate_airr	<i>Validate AIRR data</i>
---------------	---------------------------

---

**Description**

validate\_airr validates compliance of the contents of a data.frame to the AIRR data standards.

**Usage**

```
validate_airr(data, schema = RearrangementSchema)
```

**Arguments**

data	data.frame to validate.
schema	Schema object defining the data standard.

**Value**

Returns TRUE if the input data is compliant and FALSE if not.

**Examples**

```
# Get path to the rearrangement-example file
file <- system.file("extdata", "rearrangement-example.tsv.gz", package="airr")

# Load data file
df <- read_rearrangement(file)

# Validate a data.frame against the Rearrangement schema
validate_airr(df, schema=RearrangementSchema)
```

---

write_airr	<i>Write an AIRR TSV</i>
------------	--------------------------

---

**Description**

write\_airr writes a TSV containing AIRR formatted records.

**Usage**

```
write_airr(data, file, base = c("1", "0"), schema = RearrangementSchema, ...)
```

```
write_rearrangement(data, file, base = c("1", "0"), ...)
```

```
write_alignment(data, file, base = c("1", "0"), ...)
```

### Arguments

data	data.frame of Rearrangement data.
file	output file name.
base	starting index for positional fields in the output file. Fields in the input data are assumed to be 1-based closed-intervals (R style). If "1", then these fields will not be modified. If "0", then fields ending in <code>_start</code> and <code>_end</code> will be converted to 0-based half-open intervals (python style) in the output file.
schema	Schema object defining the output format.
...	additional arguments to pass to <a href="#">write_delim</a> .

### Details

`write_rearrangement` writes a data.frame containing AIRR Rearrangement data to TSV.

`write_alignment` writes a data.frame containing AIRR Alignment data to TSV.

### See Also

See [Schema](#) for the AIRR schema object definition. See [read\\_airr](#) for reading to AIRR files.

### Examples

```
# Get path to the rearrangement-example file
file <- system.file("extdata", "rearrangement-example.tsv.gz", package="airr")

# Load data file
df <- read_rearrangement(file)

# Write a Rearrangement data file
outfile <- file.path(tempdir(), "output.tsv")
write_rearrangement(df, outfile)
```

# Index

## \*Topic **datasets**

- Schema-class, [4](#)
- [, Schema, character-method (Schema-class), [4](#)
- [, Schema-method (Schema-class), [4](#)
- AlignmentSchema (Schema-class), [4](#)
- ExampleData, [2](#)
- load\_schema, [2](#), [5](#)
- names, Schema-method (Schema-class), [4](#)
- read\_airr, [3](#), [5](#), [7](#)
- read\_alignment (read\_airr), [3](#)
- read\_delim, [4](#)
- read\_rearrangement (read\_airr), [3](#)
- RearrangementSchema (Schema-class), [4](#)
- Schema, [3](#), [4](#), [7](#)
- Schema (Schema-class), [4](#)
- Schema-class, [4](#)
- Schema-method (Schema-class), [4](#)
- validate\_airr, [5](#), [6](#)
- write\_airr, [4](#), [5](#), [6](#)
- write\_alignment (write\_airr), [6](#)
- write\_delim, [7](#)
- write\_rearrangement (write\_airr), [6](#)