

# Package ‘ImmuneSpaceR’

April 15, 2020

**Type** Package

**Title** A Thin Wrapper around the ImmuneSpace Database

**Version** 1.14.1

**Date** 2014-06-12

**VignetteBuilder** knitr

**Description** Provides a convenient API for accessing data sets within ImmuneSpace ([www.immunespace.org](http://www.immunespace.org)), the data repository and analysis platform of the Human Immunology Project Consortium (HIPC).

**biocViews** DataImport, DataRepresentation, ThirdPartyClient

**URL** <https://github.com/RGLab/ImmuneSpaceR>

**BugReports** <https://github.com/RGLab/ImmuneSpaceR/issues>

**License** GPL-2

**Suggests** knitr, testthat

**Imports** utils, R6, data.table, curl, httr, Rlabkey (>= 2.3.1), Biobase, pheatmap, ggplot2 (>= 3.2.0), scales, stats, gplots, plotly, heatmaply (>= 0.7.0), jsonlite, rmarkdown, preprocessCore, flowCore, flowWorkspace, digest

**RoxygenNote** 7.0.2

**Collate** 'CreateConnection.R' 'ISCon.R' 'ISCon-cytometry.R' 'ISCon-dataset.R' 'ISCon-geneExpression.R' 'ISCon-participantGroup.R' 'ISCon-plot.R' 'ISCon-utils.R' 'ImmuneSpaceR.R' 'netrc.R' 'template.R' 'theme.R' 'utils.R' 'zzz.R'

**git\_url** <https://git.bioconductor.org/packages/ImmuneSpaceR>

**git\_branch** RELEASE\_3\_10

**git\_last\_commit** ef8edfc

**git\_last\_commit\_date** 2020-01-14

**Date/Publication** 2020-04-14

**Author** Greg Finak [aut],  
Renan Sauteraud [aut],  
Mike Jiang [aut],  
Gil Guday [aut],

Leo Dashevskiy [aut],  
 Evan Henrich [aut],  
 Ju Yeong Kim [aut],  
 Lauren Wolfe [aut],  
 Helen Miller [aut],  
 Raphael Gottardo [aut],  
 ImmuneSpace Package Maintainer [cre, cph]

**Maintainer** ImmuneSpace Package Maintainer <immunespace@gmail.com>

## R topics documented:

ImmuneSpaceR-package . . . . .	2
check_netrc . . . . .	3
CreateConnection . . . . .	3
ImmuneSpaceConnection . . . . .	4
interactive_netrc . . . . .	11
ISpalette . . . . .	12
loadConnection . . . . .	13
template_IS . . . . .	13
theme_IS . . . . .	14
write_netrc . . . . .	15
<b>Index</b>	<b>16</b>

---

ImmuneSpaceR-package *A Thin Wrapper Around ImmuneSpace*

---

### Description

ImmuneSpaceR provides a convenient API for accessing data sets within the ImmuneSpace database.

### Details

Uses the Rlabkey package to connect to ImmuneSpace. Implements caching, and convenient methods for accessing data sets.

### See Also

[CreateConnection](#)

---

check_netrc	<i>Check netrc file</i>
-------------	-------------------------

---

**Description**

Check that there is a netrc file with a valid entry for ImmuneSpace.

**Usage**

```
check_netrc()
```

**Details**

In order to connect to ImmuneSpace, you will need a '.netrc' file in your contains a 'machine' name (hostname of ImmuneSpace), and 'login' and 'password'. See [here](<https://www.labkey.org/wiki/home/Documentation/>) for more information. By default Rcurl will look for the file in your home directoty.

If no netrc is available or it is not formatted properly, write\_netrc can be used to write one. Otherwise, when specifying login and password in CreateConnection, a temporary file will be created for that connection.

**Value**

The name of the netrc file

**See Also**

CreateConnection write\_netrc

**Examples**

```
try(check_netrc())
```

---

CreateConnection	<i>CreateConnection</i>
------------------	-------------------------

---

**Description**

Constructor for ImmuneSpaceConnection class.

**Usage**

```
CreateConnection(  
  study = NULL,  
  login = NULL,  
  password = NULL,  
  verbose = FALSE,  
  onTest = FALSE  
)
```

**Arguments**

study	A "character" vector naming the study.
login	A "character". Optional argument. If there is no netrc file a temporary one can be written by passing login and password of an active ImmuneSpace account.
password	A "character". Optional. The password for the selected login.
verbose	A "logical" whether to print the extra details for troubleshooting.
onTest	A "logical" whether to connect to the test server ( <a href="https://test.immunespace.org/">https://test.immunespace.org/</a> ) instead of the production server ( <a href="https://www.immunespace.org/">https://www.immunespace.org/</a> ).

**Details**

Instantiates an ImmuneSpaceConnection for study The constructor will try to take the values of the various 'labkey.\*' parameters from the global environment. If they don't exist, it will use default values. These are assigned to 'options', which are then used by the ImmuneSpaceConnection class.

**Value**

an instance of an ImmuneSpaceConnection

**See Also**

[ImmuneSpaceConnection](#)

**Examples**

```
## Not run:
# Single study
con <- CreateConnection("SDY269")
# Cross study
con <- CreateConnection("")

## End(Not run)

sdy <- try(CreateConnection("SDY269"))
if (inherits(sdy, "try-error")) {
  warning("Read the Introduction vignette for more information on how to set
up a .netrc file.")
}
```

---

ImmuneSpaceConnection *The ImmuneSpaceConnection class*

---

**Description**

A connection represents a study or a set of studies available on ImmuneSpace. It provides function to download and display the data within these studies.

**Details**

The ImmuneSpaceConnection will initialize itself, and look for a `.netrc` file in `"~/`" the user's home directory. The `.netrc` file should contain a machine, login, and password entry to allow access to ImmuneSpace, where machine is the host name like `"www.immunespace.org"`.

It can also use global variables `labkey.url.base`, and `labkey.url.path`, to access a study. `labkey.url.base` should be `https://www.immunespace.org/`. `labkey.url.path` should be `/Studies/studyname`, where 'studyname' is the accession number of the study.

**Value**

An instance of an ImmuneSpaceConnection for a study in `labkey.url.path`.

**Constructor**

[CreateConnection](#)

**Fields**

`study` A character. The study accession number. Use an empty string (`"`) to create a connection at the project level.

`availableDatasets` A data.table. The table of datasets available in the connection object.

`cache` A list. Stores the data to avoid downloading the same tables multiple times.

`config` A list. Stores configuration of the connection object such as URL, path and username.

**Methods**

`initialize()` Initialize ImmuneSpaceConnection class. See [CreateConnection](#).

`print()` Print ImmuneSpaceConnection class.

`listDatasets(output = c("datasets", "expression"))` Lists the datasets available in the study or studies of the connection.

`listGEMatrices(verbose = FALSE, reload = FALSE, participantIds = NULL)` Lists available gene expression matrices for the connection.

`verbose`: A logical. If TRUE, whether to print the extra details for troubleshooting.

`reload`: A logical. If TRUE, retrieve the table of available gene expression matrices whether a cached version exist or not.

`participantIds`: A character vector of participant ids to filter by. Only matrices with data from `participantIds` will be returned. If NULL, all matrices are returned.

`listGEAnalysis()` Lists available gene expression analysis for the connection.

`listParticipantGroups()` Lists available participant groups on the ImmuneSpace portal.

`listParticipantGEMatrices(group, verbose = FALSE)` Lists available gene expression matrices for participants in group.

`group`: A character or integer. Call `con$listParticipantGroups()` to see available participants groups. Use `group_id` or `group_name` as input.

`verbose`: A logical. If TRUE, whether to print the extra details for troubleshooting.

`listWorkspaces(reload = FALSE)` Lists available workspaces for the connection.

`reload`: A logical. If TRUE, download the table whether a cached version exist or not.

`listGatingSets(reload = FALSE)` Lists available gating sets for the connection.

`reload`: A logical. If TRUE, download the table whether a cached version exist or not.

`summarizeCyto()` Prints a summary of cytometry data for the connection.

`summarizeGatingSet(gatingSet)` Prints a summary of a gating set. Note that this method currently works only in the ImmuneSpace RStudio session.

`gatingSet`: A character. The name of the gating set to summarize.

`loadGatingSet(gatingSet)` Loads a gating set via `flowWorkspace::load_gs` to the current environment. Note that this method currently works only in the ImmuneSpace RStudio Docker session.

`gatingSet`: A character. The name of the gating set to load.

`getDataset(x, original_view = FALSE, reload = FALSE, colFilter = NULL, ...)` Get a dataset from the connection.

`x`: A character. The name of the dataset to download.

`original_view`: A logical. If TRUE, download the original ImmPort view; else, download the default grid view.

`reload`: A logical. If TRUE, download the dataset whether a cached version exist or not.

`colFilter`: A character. A filter as returned by Rlabkey's `makeFilter` function.

`...`: Extra arguments to be passed to `labkey.selectRows`.

`getGEMatrix(matrixName = NULL, cohortType = NULL, outputType = "summary", annotation = "latest", reload = FALSE)` Downloads a probe-level or gene-symbol summarized expression matrix from ImmuneSpace and constructs an ExpressionSet. Use `experimentData()` on the resulting ExpressionSet object to see version info for annotation.

`matrixName`: A character. The name of the gene expression matrix to download.

`cohortType`: A character. The name of a cohortType that has an associated gene expression matrix. Note that if this argument is not NULL, then `matrixName` is ignored. CohortType is a concatenation of "cohort" and "cell type" that allows the user to specify a matrix for the cell type subset of a cohort.

`outputType`: A character. one of 'raw', 'normalized' or 'summary'. If 'raw', returns an expression matrix of non-normalized values by probe. 'normalized' returns normalized values by probe. 'summary' returns normalized values averaged by gene symbol.

`annotation`: A character. one of 'default', 'latest', or 'ImmSig'. Determines which feature annotation set (FAS) is used. 'default' uses the FAS from when the matrix was generated. 'latest' uses a recently updated FAS based on the original. 'ImmSig' is specific to studies involved in the ImmuneSignatures project and uses the annotation from when the meta-study's manuscript was created.

`reload`: A logical. If set to TRUE, the matrix will be downloaded again, even if a cached copy exists in the ImmuneSpaceConnection object.

`verbose`: A logical. If set to TRUE, notes on how the expressionSet object was created will be printed, including normalization, summarization, `feature_annotation_set`, and `alias2symbol` mapping version of `org.Hs.eg.db`.

`getGEAnalysis(...)` Downloads data from the gene expression analysis results table.

`...`: A list of arguments to be passed to `labkey.selectRows`.

`getGEInputs()` Downloads data from the gene expression input samples table.

`getParticipantData(group, dataType, original_view = FALSE, ...)` Returns a data.table with data subset by participant group.

`group`: A character or integer. Call `con$listParticipantGroups()` to see available participants groups. Use `group_id` or `group_name` as input.

`dataType`: A character. Use `con$availableDatasets` to see available dataset names.

`getParticipantGEMatrix(group, outputType = "summary", annotation = "latest", reload = FALSE)`  
 Downloads probe-level or gene-symbol summarized expression matrices for all participants within group from ImmuneSpace and constructs an ExpressionSet containing observations for each participant in group where gene expression data is available.  
 group: A character or integer. Call `con$listParticipantGroups()` to see available participants groups. Use `group_id` or `group_name` as input.  
 outputType: A character. one of 'raw', 'normalized' or 'summary'. If 'raw', returns an expression matrix of non-normalized values by probe. 'normalized' returns normalized values by probe. 'summary' returns normalized values averaged by gene symbol.  
 annotation: A character. one of 'default', 'latest', or 'ImmSig'. Determines which feature annotation set (FAS) is used. 'default' uses the FAS from when the matrix was generated. 'latest' uses a recently updated FAS based on the original. 'ImmSig' is specific to studies involved in the ImmuneSignatures project and uses the annotation from when the meta-study's manuscript was created.  
 reload: A logical. If set to TRUE, matrices will be downloaded again, even if a cached copy exists in the ImmuneSpaceConnection object.

`downloadGEFiles(files, destdir = ".")` Downloads gene expression raw data files.  
 files: A character. Filenames as shown on the `gene_expression_files` dataset.  
 destdir: A character. The local path to store the downloaded files.

`addTreatment(expressionSet)` Adds treatment information to the `phenoData` of an ExpressionSet.  
 expressionSet: An ExpressionSet. The ExpressionSet object that has been downloaded from the connection.

`mapSampleNames(EM = NULL, colType = "participant_id")` Changes the sampleNames of an ExpressionSet fetched by `getGEMatrix` using the information in the `phenodData` slot.  
 EM: An ExpressionSet, as returned by `getGEMatrix`.  
 colType: A character. The type of column names. Valid options are 'expsample\_accession' and 'participant\_id'.

`plot(...)` Visualizes a selected dataset. This method is used by the DataExplorer module on the ImmuneSpace portal.  
 dataset: A character. The name of the dataset to plot, as displayed by the `listDataset` method.  
 normalize\_to\_baseline: A logical. If TRUE, the values are plotted as log2 fold-change from baseline.  
 type: A character. The type of plot. Valid choices are 'auto', 'heatmap', 'boxplot', 'lineplot', 'violinplot'. If set to 'auto', the function will select an appropriate plot type for the selected data.  
 filter: A filter as created by the `makeFilter` function from Rlabkey.  
 facet: The faceting for ggplot2 based plots. Valid choices are 'grid' and 'wrap'.  
 text\_size: The size of all text elements in the plot.  
 legend: A character. Columns of the dataset or demographics to be added as legend on the heatmap. This argument is ignored if the plot type isn't heatmap.  
 show\_virus\_strain: A logical. Should all the virus strains be shown or should the values be averaged. Only used when `dataset = 'hai'`.  
 interactive: A logical. If TRUE, an interactive plot will be created. The default is FALSE.  
 ...: Extra argument to be passed to ggplot. e.g: `shape = 'Age'`, `color = 'Race'`.

`clearCache()` Clears the cache. Removes downloaded datasets and expression matrices.

## Methods

### Public methods:

- `ISCon$listWorkspaces()`
- `ISCon$listGatingSets()`
- `ISCon$summarizeCyto()`
- `ISCon$summarizeGatingSet()`
- `ISCon$loadGatingSet()`
- `ISCon$listDatasets()`
- `ISCon$getDataset()`
- `ISCon$listGEMatrices()`
- `ISCon$listGEAnalysis()`
- `ISCon$getGEMatrix()`
- `ISCon$getGEAnalysis()`
- `ISCon$getGEInputs()`
- `ISCon$getGEFiles()`
- `ISCon$downloadGEFiles()`
- `ISCon$addTreatment()`
- `ISCon$mapSampleNames()`
- `ISCon$listParticipantGroups()`
- `ISCon$getParticipantData()`
- `ISCon$listParticipantGEMatrices()`
- `ISCon$getParticipantGEMatrix()`
- `ISCon$plot()`
- `ISCon$print()`
- `ISCon$clearCache()`
- `ISCon$new()`
- `ISCon$clone()`

### Method `listWorkspaces()`:

*Usage:*

```
ISCon$listWorkspaces(reload = FALSE)
```

### Method `listGatingSets()`:

*Usage:*

```
ISCon$listGatingSets(reload = FALSE)
```

### Method `summarizeCyto()`:

*Usage:*

```
ISCon$summarizeCyto()
```

### Method `summarizeGatingSet()`:

*Usage:*

```
ISCon$summarizeGatingSet(gatingSet)
```

### Method `loadGatingSet()`:

*Usage:*

```
ISCon$loadGatingSet(gatingSet)
```

**Method** listDatasets():*Usage:*

```
ISCon$listDatasets(output = c("datasets", "expression"))
```

**Method** getDataset():*Usage:*

```
ISCon$getDataset(  
  x,  
  original_view = FALSE,  
  reload = FALSE,  
  colFilter = NULL,  
  transformMethod = "none",  
  ...  
)
```

**Method** listGEMatrices():*Usage:*

```
ISCon$listGEMatrices(verbose = FALSE, reload = FALSE, participantIds = NULL)
```

**Method** listGEAnalysis():*Usage:*

```
ISCon$listGEAnalysis()
```

**Method** getGEMatrix():*Usage:*

```
ISCon$getGEMatrix(  
  matrixName = NULL,  
  cohortType = NULL,  
  outputType = "summary",  
  annotation = "latest",  
  reload = FALSE,  
  verbose = FALSE  
)
```

**Method** getGEAnalysis():*Usage:*

```
ISCon$getGEAnalysis(...)
```

**Method** getGEInputs():*Usage:*

```
ISCon$getGEInputs()
```

**Method** getGEFiles():*Usage:*

```
ISCon$getGEFiles(files, destdir = ".", quiet = FALSE)
```

**Method** downloadGEFiles():*Usage:*

```
ISCon$downloadGEFiles(files, destdir = ".")
```

**Method** addTreatment():*Usage:*

ISCon\$addTreatment(expressionSet)

**Method** mapSampleNames():*Usage:*

ISCon\$mapSampleNames(EM = NULL, colType = "participant\_id")

**Method** listParticipantGroups():*Usage:*

ISCon\$listParticipantGroups()

**Method** getParticipantData():*Usage:*

```
ISCon$getParticipantData(  
  group,  
  dataType,  
  original_view = FALSE,  
  reload = FALSE,  
  colFilter = NULL,  
  transformMethod = "none",  
  ...  
)
```

**Method** listParticipantGEMatrices():*Usage:*

ISCon\$listParticipantGEMatrices(group, verbose = FALSE)

**Method** getParticipantGEMatrix():*Usage:*

```
ISCon$getParticipantGEMatrix(  
  group,  
  outputType = "summary",  
  annotation = "latest",  
  reload = FALSE  
)
```

**Method** plot():*Usage:*

ISCon\$plot(...)

**Method** print():*Usage:*

ISCon\$print()

**Method** clearCache():*Usage:*

ISCon\$clearCache()

**Method** new():

*Usage:*

```
ISCon$new(
  study = NULL,
  login = NULL,
  password = NULL,
  verbose = FALSE,
  onTest = FALSE
)
```

**Method** clone(): The objects of this class are cloneable with this method.

*Usage:*

```
ISCon$clone(deep = FALSE)
```

*Arguments:*

deep Whether to make a deep clone.

**See Also**

[CreateConnection ImmuneSpaceR-package](#)

**Examples**

```
## Not run:
# Create a connection (Initiate a ImmuneSpaceConnection object)
sdy269 <- CreateConnection("SDY269")

# Print the connection object
sdy269

# Retrieve the HAI dataset
HAI <- sdy269$getDataset("hai")

# Fetch a summarized gene expression matrix with latest annotation
LAIV <- sdy269$getGEMatrix("LAIV_2008")

# Visualize the ELISA dataset
sdy269$plot("elisa")

## End(Not run)

sdy <- try(CreateConnection("SDY269"))
if (inherits(sdy, "try-error")) {
  warning("Read the Introduction vignette for more information on how to set
up a .netrc file.")
}
```

---

interactive\_netrc

*Interactively write a netrc file*

---

**Description**

Write a netrc file that is valid for accessing ImmuneSpace

**Usage**

```
interactive_netrc()
```

**Value**

A netrc file that is verified to connect to ImmuneSpace

**Examples**

```
## Not run:  
interactive_netrc()  
  
## End(Not run)
```

---

ISpalette

*ImmuneSpace palette*

---

**Description**

Create a color gradient of the selected length that matches the ImmuneSpace theme.

**Usage**

```
ISpalette(n)
```

**Arguments**

n                    A numeric. The length of the desired palette.

**Value**

A character vector colors in hexadecimal code of length n.

**Examples**

```
plot(1:10, col = ISpalette(10), cex = 10, pch = 16)
```

---

loadConnection	<i>Save/Load an ImmuneSpaceConnection object from disk</i>
----------------	------------------------------------------------------------

---

### Description

Connection can hold a lot of data in cache. If a lot of work has been done (e.g: lots of downloaded datasets and gene-expression matrices), it can be useful to save the connection for later work or even offline use.

### Usage

```
loadConnection(file)

saveConnection(con, file)
```

### Arguments

file	The file name to be saved to or loaded from
con	An ImmuneSpaceConnection. The connection to save to file. To be loaded later using loadConnection.

### Value

An ImmuneSpaceConnection object

### Examples

```
# Sample saved connection with pre-downloaded expression matrices and datasets
saved <- system.file("extdata/saved_con.rds", package = "ImmuneSpaceR")
new_con <- loadConnection(saved)
new_con
names(new_con$cache)
## Not run:
saveConnection(new_con, tempfile())

## End(Not run)
```

---

template_IS	<i>template_IS</i>
-------------	--------------------

---

### Description

A HTML template for knitted reports that matches ImmuneSpace's graphic style. It is based on [html\\_document](#) from the **rmarkdown** package with css, theme, and template parameters disabled.

### Usage

```
template_IS(...)
```

**Arguments**

... See [html\\_document](#)

**Details**

See the documentation for [html\\_document](#) or the [online documentation](#) for additional details on using the `html_document` format. Compared to `html_document`, it:

- uses a custom css stylesheet
- does not use bootstrap themes

**Value**

R Markdown output format to pass to [render](#)

**Examples**

```
## Not run:
library(ImmuneSpaceR)
rmarkdown::render("input.Rmd", template_IS())
rmarkdown::render("input.Rmd", template_IS(toc = TRUE))

## End(Not run)
template_IS()
```

---

theme\_IS

*theme\_IS*

---

**Description**

Theme that matches ImmuneSpace's graphic style. The theme modifies the background, the grid lines, the axis, and the colors used by continuous and gradient scales.

**Usage**

```
theme_IS(base_size = 12)
```

**Arguments**

`base_size` A numeric. Base font size.

**Details**

List of modified ggplot2 elements: `panel.background`, `panel.grid.major`, `panel.grid.minor`, `axis.ticks`, `axis.line.x`, `axis.line.y`, `plot.title`, and `strip.background`.

The default `scale_fill_gradient`, `scale_fill_continuous`, `scale_colour_gradient` and `scale_colour_continuous` are also replaced by a custom scale.

**Value**

A theme object

**Examples**

```
library(ggplot2)
p <- ggplot(data = mtcars) + geom_point(aes(x = mpg, y = cyl, color = hp)) + facet_grid(vs ~ am)
p + theme_IS()
```

---

`write_netrc`*Write a netrc file*

---

**Description**

Write a netrc file that is valid for accessing ImmuneSpace

**Usage**

```
write_netrc(login, password, machine = "www.immunespace.org", file = NULL)
```

**Arguments**

<code>login</code>	A character. The email address used for logging in on ImmuneSpace.
<code>password</code>	A character. The password associated with the login.
<code>machine</code>	A character. The server to connect.
<code>file</code>	A character. The credentials will be written into that file. If left NULL, the netrc will be written into a temporary file.

**Value**

A character vector containing the file paths for netrc

**Examples**

```
write_netrc("immunespaceuser@gmail.com", "mypassword")
```

# Index

addTreatmentt (ImmuneSpaceConnection), 4

check\_netrc, 3

CreateConnection, 2, 3, 5, 11

getDataset (ImmuneSpaceConnection), 4

getGEAnalysis (ImmuneSpaceConnection), 4

getGEFiles (ImmuneSpaceConnection), 4

getGEInputs (ImmuneSpaceConnection), 4

getGEMatrix (ImmuneSpaceConnection), 4

getParticipantData  
(ImmuneSpaceConnection), 4

getParticipantGEMatrix  
(ImmuneSpaceConnection), 4

html\_document, 13, 14

ImmuneSpaceConnection, 4, 4

ImmuneSpaceR (ImmuneSpaceR-package), 2

ImmuneSpaceR-package, 2

interactive\_netrc, 11

ISCon (ImmuneSpaceConnection), 4

ISpalette, 12

listDatasets (ImmuneSpaceConnection), 4

listGatingSets (ImmuneSpaceConnection),  
4

listGEAnalysis (ImmuneSpaceConnection),  
4

listGEMatrices (ImmuneSpaceConnection),  
4

listParticipantGroups  
(ImmuneSpaceConnection), 4

listParticipantMatrices  
(ImmuneSpaceConnection), 4

listWorkspaces (ImmuneSpaceConnection),  
4

loadConnection, 13

loadGatingSet (ImmuneSpaceConnection), 4

mapSampleNames (ImmuneSpaceConnection),  
4

render, 14

saveConnection (loadConnection), 13

summarizeCyto (ImmuneSpaceConnection), 4

summarizeGatingSet  
(ImmuneSpaceConnection), 4

template\_IS, 13

theme\_IS, 14

write\_netrc, 15