

Package ‘IgGeneUsage’

May 8, 2024

Type Package

Title Differential gene usage in immune repertoires

Version 1.19.0

Description Detection of biases in the usage of immunoglobulin (Ig) genes is an important task in immune repertoire profiling. IgGeneUsage detects aberrant Ig gene usage between biological conditions using a probabilistic model which is analyzed computationally by Bayes inference. With this IgGeneUsage also avoids some common problems related to the current practice of null-hypothesis significance testing.

License MIT + file LICENSE

Depends R (>= 4.2.0)

Imports methods, reshape2 (>= 1.4.3), Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.2.0), SummarizedExperiment, tidyr

Suggests BiocStyle, knitr, rmarkdown, testthat (>= 2.1.0), ggplot2, ggforce, ggrepel, patchwork

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Encoding UTF-8

LazyData false

NeedsCompilation yes

biocViews DifferentialExpression, Regression, Genetics, Bayesian, BiomedicalInformatics, ImmunoOncology, MathematicalBiology

BugReports <https://github.com/snaketron/IgGeneUsage/issues>

URL <https://github.com/snaketron/IgGeneUsage>

RoxygenNote 6.1.1

VignetteBuilder knitr

Biarch true

git_url https://git.bioconductor.org/packages/IgGeneUsage
git_branch devel
git_last_commit a4f2b16
git_last_commit_date 2024-04-30
Repository Bioconductor 3.20
Date/Publication 2024-05-08
Author Simo Kitanovski [aut, cre]
Maintainer Simo Kitanovski <simo.kitanovski@uni-due.de>

Contents

IgGeneUsage-package	2
CDR3_Epitopes	3
DGU	4
d_zibb_1	6
d_zibb_2	7
d_zibb_3	8
d_zibb_4	9
d_zibb_5	10
d_zibb_6	11
Ig	12
IGHV_HCV	12
Ig_SE	13
LOO	14
Index	17

IgGeneUsage-package *The R package IgGeneUsage*

Description

IgGeneUsage detects aberrant immunoglobulin (Ig) gene usage between adaptive immune repertoires that belong to different biological conditions using a probabilistic model which is analyzed computationally by Bayes inference.

Details

This package contains functions for:

1. differential Ig gene usage analysis (function DGU)
2. posterior predictive checks (part of results generated by function DGU)
3. leave-one-out cross validation (function LOO)

Author(s)

Authors and maintainers:

- Simo Kitanovski <simokitanovski@uni-due.de> ([ORCID](#))

See Also

Useful links:

- <https://github.com/snaketron/IgGeneUsage>
- Report bugs at <https://github.com/snaketron/IgGeneUsage/issues>

CDR3_Epitopes	<i>Net charge usage in CDR3 sequences of T-cell receptor repertoires disturbed by Influenza-A and CMV</i>
---------------	---

Description

Data of CDR3 sequence from human T-cells receptors (TRB-chain) downloaded from VDJdb. CDR3 sequences annotated to epitopes in Influenza-A and CMV were selected from different publications, as long as the publication contains at least 100 CDR3 sequences. Each publication is considered as a repertoire (sample).

To compute the net CDR3 sequence charge, we consider the amino acids K, R and H as +1 charged, while D and E as -1 charged. Thus, we computed the net charge of a CDR3 sequence by adding up the individual residue charges.

Usage

```
data("CDR3_Epitopes")
```

Format

A data frame with 4 columns: "individual_id", "condition", "gene_name" and "gene_usage_count". The format of the data is suitable to be used as input in IgGeneUsage

```
gene_name = net charge group
```

Source

```
https://vdjdb.cdr3.net/
```

Examples

```
data(CDR3_Epitopes)
head(CDR3_Epitopes)
```

Description

IgGeneUsage detects differential gene usage (DGU) in immune repertoires that belong to two biological conditions.

Usage

```
DGU(ud,
    mcmc_warmup,
    mcmc_steps,
    mcmc_chains,
    mcmc_cores,
    hdi_lvl,
    adapt_delta,
    max_treedepth,
    paired = FALSE)
```

Arguments

ud	Data.frame with 4 or 5 columns: <ul style="list-style-type: none"> • 'individual_id' = character, name of the donor (e.g. Pt1) • 'condition' = character, name of biological conditions (e.g. tumor) • 'gene_name' = character, Ig gene name (e.g. IGHV1-69) • 'gene_usage_count' = number, frequency (=usage) of rearrangements from individual_id x condition x gene_name • [optional] 'replicate' = character or number. Replicate id, if more than one repertoire (biological replicates) is available per individual <p>ud can also be a SummarizedExperiment object. See exemplary data 'data(Ig_SE)' for more information.</p>
mcmc_chains, mcmc_warmup, mcmc_steps, mcmc_cores	Number of MCMC chains (default = 4), number of cores to use (default = 1), length of MCMC chains (default = 1,500), length of adaptive part of MCMC chains (default = 500).
hdi_lvl	Highest density interval (HDI) (default = 0.95).
adapt_delta	MCMC setting (default = 0.95).
max_treedepth	MCMC setting (default = 12).
paired	should a paired samples differential Ig gene analysis be performed (default = FALSE)?

Details

The main input of `IgGeneUsage` is a table with Ig gene usage frequencies for a set of repertoires that belong to one of two biological condition. For the DGU analysis between two biological conditions, `IgGeneUsage` employs a Bayesian hierarchical model for zero-inflated beta-binomial (ZIBB) regression (see vignette 'User Manual: `IgGeneUsage`').

Value

<code>dgu</code>	DGU statistics for each gene: 1) <code>es</code> = effect size on DGU (mean, median standard error (<code>se</code>), standard deviation (<code>sd</code>), L (low boundary of HDI), H (high boundary of HDI); 2) <code>contrast</code> = direction of the effect; 3) <code>pmax</code> = probability of DGU. This summary is only available if the input data contains at least two conditions
<code>gu</code>	gene usage (GU) summary of each gene in each condition
<code>fit</code>	stanfit object
<code>ppc</code>	two types of posterior predictive checks: 1) repertoire- specific, 2) condition-specific
<code>ud</code>	processed gene usage data used for the model

Author(s)

Simo Kitanovski <simo.kitanovski@uni-due.de>

See Also

`LOO`, `Ig`, `IGHV_Epitopes`, `IGHV_HCV`, `Ig_SE`, `d_zibb_1`, `d_zibb_2`, `d_zibb_3`

Examples

```
# input data
data(d_zibb_2)
head(d_zibb_2)

# run differential gene usage (DGU)
M <- DGU(ud = d_zibb_2,
        mcmc_warmup = 350,
        mcmc_steps = 1500,
        mcmc_chains = 2,
        mcmc_cores = 1,
        hdi_lvl = 0.95,
        adapt_delta = 0.8,
        max_treedepth = 10,
        paired = FALSE)

# look at M elements
names(M)

# look at DGU results
head(M$dgu)
```

```
# look at posterior predictive checks (PPC)
head(M$ppc)
```

d_zibb_1	<i>Simulated Ig gene usage data</i>
----------	-------------------------------------

Description

A small example dataset that has the following features:

- 1 conditions
- 5 individuals (samples)
- 15 Ig genes

This dataset was simulated from zero-inflated beta-binomial (ZIBB) distribution. Simulation code is available in `inst/scripts/d_zibb_1.R`

Usage

```
data("d_zibb_1", package = "IgGeneUsage")
```

Format

A data frame with 4 columns:

- "individual_id"
- "condition"
- "gene_name"
- "gene_name_count"

This format is accepted by `IgGeneUsage`.

Source

Simulation code is provided in `inst/scripts/d_zibb_1.R`

Examples

```
data("d_zibb_1", package = "IgGeneUsage")
head(d_zibb_1)
```

d_zibb_2

Simulated Ig gene usage data

Description

A small example dataset that has the following features:

- 1 conditions
- 5 individuals (samples)
- 3 biological replicates per individual
- 15 Ig genes

This dataset was simulated from zero-inflated beta-binomial (ZIBB) distribution. Simulation code is available in `inst/scripts/d_zibb_2.R`

Usage

```
data("d_zibb_2", package = "IgGeneUsage")
```

Format

A data frame with columns:

- "individual_id"
- "condition"
- "gene_name"
- "replicate"
- "gene_name_count"

This format is accepted by `IgGeneUsage`.

Source

Simulation code is provided in `inst/scripts/d_zibb_2.R`

Examples

```
data("d_zibb_2", package = "IgGeneUsage")  
head(d_zibb_2)
```

`d_zibb_3`*Simulated Ig gene usage data*

Description

A small example dataset that has the following features:

- 3 conditions
- 5 samples per condition
- 8 Ig genes

This dataset was simulated from zero-inflated beta-binomial (ZIBB) distribution. Simulation code is available in `inst/scripts/d_zibb_3.R`

Usage

```
data("d_zibb_3", package = "IgGeneUsage")
```

Format

A data frame with columns:

- "individual_id"
- "condition"
- "gene_name"
- "gene_name_count"

This format is accepted by `IgGeneUsage`.

Source

Simulation code is provided in `inst/scripts/d_zibb_3.R`

Examples

```
data("d_zibb_3", package = "IgGeneUsage")  
head(d_zibb_3)
```

d_zibb_4

Simulated Ig gene usage data

Description

A small example dataset that has the following features:

- 2 conditions
- 7 individuals per condition
- 4 replicates per individual
- 8 Ig genes

This dataset was simulated from zero-inflated beta-binomial (ZIBB) distribution. Simulation code is available in `inst/scripts/d_zibb_4.R`

Usage

```
data("d_zibb_4", package = "IgGeneUsage")
```

Format

A data frame with columns:

- "individual_id"
- "condition"
- "gene_name"
- "replicate"
- "gene_name_count"

This format is accepted by `IgGeneUsage`.

Source

Simulation code is provided in `inst/scripts/d_zibb_4.R`

Examples

```
data("d_zibb_4", package = "IgGeneUsage")  
head(d_zibb_4)
```

`d_zibb_5`*Simulated Ig gene usage data*

Description

A small example of paired-sample IRRs with these features:

- 3 conditions
- 6 individuals with one IRRs per condition
- 10 Ig genes

This dataset was simulated from zero-inflated beta-binomial (ZIBB) distribution. Simulation code is available in `inst/scripts/d_zibb_5.R`

Usage

```
data("d_zibb_5", package = "IgGeneUsage")
```

Format

A data frame with columns:

- "individual_id"
- "condition"
- "gene_name"
- "gene_name_count"

This format is accepted by `IgGeneUsage`.

Source

Simulation code is provided in `inst/scripts/d_zibb_5.R`

Examples

```
data("d_zibb_5", package = "IgGeneUsage")  
head(d_zibb_5)
```

`d_zibb_6`*Simulated Ig gene usage data*

Description

A small example of paired-sample *with replicates* IRRs with these features:

- 3 conditions
- 9 individuals with one IRRs per condition
- 10 Ig genes
- 4 replicates per individual

This dataset was simulated from zero-inflated beta-binomial (ZIBB) distribution. Simulation code is available in `inst/scripts/d_zibb_6.R`

Usage

```
data("d_zibb_6", package = "IgGeneUsage")
```

Format

A data frame with columns:

- "individual_id"
- "condition"
- "gene_name"
- "replicate"
- "gene_name_count"

This format is accepted by `IgGeneUsage`.

Source

Simulation code is provided in `inst/scripts/d_zibb_6.R`

Examples

```
data("d_zibb_6", package = "IgGeneUsage")  
head(d_zibb_6)
```

Ig

IGHV gene family usage in vaccine-challenged B-cell repertoires

Description

A small example database subset from study evaluating vaccine-induced changes in B-cell populations publicly provided by R-package alakazam (version 0.2.11). It contains IGHV gene family usage, reported in four B-cell populations (samples IgM, IgD, IgG and IgA) across two timepoints (conditions = -1 hour and +7 days).

Usage

```
data("Ig")
```

Format

A data frame with 4 columns: "sample_id", "condition", "gene_name" and "gene_usage_count". The format of the data is suitable to be used as input in IgGeneUsage

Source

R package: alakazam version 0.2.11

References

Laserson U and Vigneault F, et al. High-resolution antibody dynamics of vaccine-induced immune responses. Proc Natl Acad Sci USA. 2014 111:4928-33.

Examples

```
data(Ig)  
head(Ig)
```

IGHV_HCV

IGHV gene usage in HCV+ and healthy individuals

Description

Publicly available dataset of IGHV segment usage in memory B-cells of 22 HCV+ individuals and 7 healthy donors.

Usage

```
data("IGHV_HCV")
```

Format

A data frame with 4 columns: "individual_id", "condition", "gene_name" and "gene_usage_count". The format of the data is suitable to be used as input in IgGeneUsage

Source

Tucci, Felicia A., et al. "Biased IGH VDJ gene repertoire and clonal expansions in B cells of chronically hepatitis C virus–infected individuals." *Blood* 131.5 (2018): 546-557.

Examples

```
data(IGHV_HCV)
head(IGHV_HCV)
```

Ig_SE	<i>IGHV gene family usage in vaccine-challenged B-cell repertoires (SummarizedExperiment object)</i>
-------	--

Description

A small example database subset from study evaluating vaccine-induced changes in B-cell populations publicly provided by R-package alakazam (version 0.2.11). It contains IGHV gene family usage, reported in four B-cell populations (samples IgM, IgD, IgG and IgA) across two timepoints (conditions = -1 hour and +7 days).

Usage

```
data("Ig_SE")
```

Format

A SummarizedExperiment object with 1) assay data (rows = gene name, columns = repertoires) and 2) column data.frame in which the sample names and the corresponding biological condition labels are noted.

Source

R package: alakazam version 0.2.11

References

Laserson U and Vigneault F, et al. High-resolution antibody dynamics of vaccine-induced immune responses. *Proc Natl Acad Sci USA*. 2014 111:4928-33.

Examples

```
# inspect the data
data(Ig_SE)

# repertoire information: must have the two columns: 'condition' and
'individual_id'

SummarizedExperiment::colData(Ig_SE)

# assay counts (gene frequency usage)
SummarizedExperiment::assay(x = Ig_SE)
```

LOO	<i>Leave-one-out analysis for quantitative evaluation of the probability of DGU</i>
-----	---

Description

IgGeneUsage detects differential gene usage (DGU) in immune repertoires that belong to two biological conditions.

To quantify the robustness of the estimated probability of DGU (pmax), IgGeneUsage has a built-in procedure for a fully Bayesian leave-one-out (LOO) analysis. In each LOO step we discard the data of one of the repertoires, and use the remaining data to analyze for DGU with IgGeneUsage. In each step we record pmax for all genes. Finally, we evaluate the variability of pmax for a given across the different LOO steps. Low variability in pmax: robust DGU; high variability: unrobust DGU.

For datasets that include many repertoires (e.g. 100) LOO can be computationally costly.

Usage

```
LOO(ud,
    mcmc_warmup,
    mcmc_steps,
    mcmc_chains,
    mcmc_cores,
    hdi_lvl,
    adapt_delta,
    max_treedepth,
    paired = FALSE)
```

Arguments

ud	Data.frame with 4 columns: <ul style="list-style-type: none"> • 'individual_id' = character, name of the donor (e.g. Pt1) • 'condition' = character, name of biological conditions (e.g. tumor) • 'gene_name' = character, Ig gene name (e.g. IGHV1-69)
----	--

- 'gene_usage_count' = number, frequency (=usage) of rearrangements from individual_id x condition x gene_name
- [optional] 'replicate' = character or number. Replicate id, if more than one repertoire (biological replicates) is available per individual

ud can also be a SummarizedExperiment object. See dataset 'data(Ig_SE)' for more information.

mcmc_chains, mcmc_warmup, mcmc_steps, mcmc_cores

Number of MCMC chains (default = 4), number of cores to use (default = 1), length of MCMC chains (default = 1,500), length of adaptive part of MCMC chains (default = 500).

hdi_lvl Highest density interval (HDI) (default = 0.95).

adapt_delta MCMC setting (default = 0.95).

max_treedepth MCMC setting (default = 12).

paired should a paired samples differential Ig gene analysis be performed (default = FALSE)?

Details

IgGeneUsage invokes the function DGU in each LOO step. For more details see help for DGU or vignette 'User Manual: IgGeneUsage'.

Value

loo DGU statistics for each Ig gene for specific LOO step:

- es = effect size statistics: mean, median, standard error (se), standard deviation (sd), L/H (low/high boundary of HDI)
- contrast = direction of the effect
- pmax = DGU probability
- loo_id (LOO step)
- Neff (effective sample size), Rhat (potential scale reduction factor)

Author(s)

Simo Kitanovski <simo.kitanovski@uni-due.de>

See Also

DGU, Ig, IGHV_Epitopes, IGHV_HCV, Ig_SE, d_zibb_1, d_zibb_2, d_zibb_3

Examples

```
# input data:
data("Ig", package = "IgGeneUsage")
head(Ig)

# run leave-one-out (LOO)
L <- LOO(ud = Ig,
```

```
mcmc_warmup = 500,  
mcmc_steps = 2000,  
mcmc_chains = 3,  
mcmc_cores = 1,  
hdi_lvl = 0.95,  
adapt_delta = 0.99,  
max_treedepth = 10,  
paired = FALSE)  
  
# how many LOOs?  
names(L)  
  
# elements in first LOO, see vignette about how to extract results  
names(L[[1]])
```


Index

- * **CDR3_Epitopes**
 - CDR3_Epitopes, [3](#)
- * **IGHV_HCV**
 - IGHV_HCV, [12](#)
- * **Ig_SE**
 - Ig_SE, [13](#)
- * **Ig**
 - Ig, [12](#)
- * **d_zibb_1**
 - d_zibb_1, [6](#)
- * **d_zibb_2**
 - d_zibb_2, [7](#)
- * **d_zibb_3**
 - d_zibb_3, [8](#)
- * **d_zibb_4**
 - d_zibb_4, [9](#)
- * **d_zibb_5**
 - d_zibb_5, [10](#)
- * **d_zibb_6**
 - d_zibb_6, [11](#)

CDR3_Epitopes, [3](#)

d_zibb_1, [6](#)
d_zibb_2, [7](#)
d_zibb_3, [8](#)
d_zibb_4, [9](#)
d_zibb_5, [10](#)
d_zibb_6, [11](#)
DGU, [4](#)

Ig, [12](#)
Ig_SE, [13](#)
IgGeneUsage (IgGeneUsage-package), [2](#)
IgGeneUsage-package, [2](#)
IGHV_HCV, [12](#)

LOO, [14](#)