

Package ‘immReferent’

December 6, 2025

Title An Interface for Immune Receptor and HLA Gene IMGT Reference Data

Version 0.99.5

Description Provides a consistent interface for downloading, storing, and accessing IMGT immune receptor (TCR/BCR) and HLA sequences (both nucleotide and protein). This package serves as a core dependency for immunogenomics packages, ensuring reliable and high-quality sequence access.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

biocViews Software, Annotation, Sequencing

Depends R (>= 4.5.0)

Imports Biostrings, httr, jsonlite, methods, rvest, tibble, yaml

Suggests BiocManager, BiocStyle, knitr, mockery, spelling, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://github.com/BorchLab/immReferent/>

BugReports <https://github.com/BorchLab/immReferent/issues>

git_url <https://git.bioconductor.org/packages/immReferent>

git_branch devel

git_last_commit 9933f32

git_last_commit_date 2025-10-14

Repository Bioconductor 3.23

Date/Publication 2025-12-05

Author Nick Borcharding [aut, cre]

Maintainer Nick Borcharding <ncborch@gmail.com>

Contents

immReferent-package	2
getIMGT	3
getOGRDB	4
is_imgt_available	6
is_ogrdb_available	6
listIMGT	7
listOGRDB	7
loadIMGT	8
loadOGRDB	9
refreshIMGT	10
refreshOGRDB	11
Index	13

immReferent-package	<i>immReferent: An Interface for Immune Receptor and HLA Gene Reference Data</i>
---------------------	--

Description

immReferent provides a stable, reproducible, and lightweight interface to reference sequences for immune receptors (TCR/BCR) and HLA genes sourced from IMGT, IPD-IMGT/HLA, and the AIRR-C’s OGRDB. It centralizes downloading, caching, and querying of curated nucleotide and protein sequences, and plays a foundational role in computational immunology workflows.

Details

The package is designed as a common reference layer across immunoinformatics tools, ensuring consistent provenance and offline reproducibility via caching.

Core functionality

- Download and parse receptor and HLA sequences from IMGT and OGRDB.
- Local caching to support offline, reproducible analysis.
- Query by gene, allele, species, locus, and sequence type/format.
- Interoperability with Bioconductor classes such as `Biostrings::DNASTringSet` and `Biostrings::AAStringSet`.

Supported data sources

- IMGT — The international ImMunoGeneTics information system: <https://www.imgt.org/>
- IPD-IMGT/HLA — The HLA Database: <https://www.ebi.ac.uk/ipd/imgt/hla/>
- OGRDB — Open Germline Receptor Database (AIRR-C): <https://ogrdb.airr-community.org/>

Getting started

```
browseVignettes("immReferent")
```

Attribution and Licensing

Data obtained from IMGT and OGRDB must be cited according to their terms. IMGT data are distributed under a [CC BY-NC-ND 4.0](#) license. Proper attribution is required, and derivative or commercial use is restricted per IMGT policy. Always review the current licensing and citation requirements of each resource prior to use.

#' @seealso <https://github.com/BorchLab/immReferent>
<https://github.com/BorchLab/Ibex/immReferent>

Author(s)

Maintainer: Nick Borcharding <ncborch@gmail.com>

See Also

Useful links:

- <https://github.com/BorchLab/immReferent/>
- Report bugs at <https://github.com/BorchLab/immReferent/issues>

getIMGT

Download and Load Immune Receptor and HLA Sequences from IMGT

Description

This is the main function to download and load reference sequences from IMGT and the IPD-IMGT/HLA database. It handles caching of downloaded files.

Usage

```
getIMGT(  
  species = "human",  
  gene,  
  type = c("NUC", "PROT"),  
  refresh = FALSE,  
  suppressMessages = FALSE  
)
```

Arguments

species	The species for which to download data. Required for TCR/BCR queries. Currently supported: "human", "mouse", "rat", "rabbit", "pig", "dog", "rhesus_monkey", "cyno monkey". Defaults to "human" for HLA.
gene	The gene or locus to download. For TCR/BCR, this can be a specific chain (e.g., "IGHV", "TRBJ") or a group (e.g., "IGH", "TCR"). For HLA, use "HLA".

type	The type of sequence to retrieve. Either "NUC" for nucleotide or "PROT" for protein sequences. This primarily distinguishes between VDJ nucleotide and V-region amino acid sequences for TCR/BCR genes.
refresh	Logical. If 'TRUE', forces a re-download of the data even if it exists in the cache.
suppressMessages	Logical. If 'TRUE', suppresses the license and other informational messages.

Value

A 'DNAStringSet' or 'AAStringSet' object containing the requested sequences.

Examples

```
if(is_imgt_available()) {
  # Download human IGHV nucleotide sequences
  ighv_nuc <- getIMGT(species = "human",
                     gene = "IGHV",
                     type = "NUC")

  # Download all HLA protein sequences
  hla_prot <- getIMGT(gene = "HLA",
                    type = "PROT")

  # Download all mouse TRB genes
  trb_mouse <- getIMGT(species = "mouse",
                      gene = "TRB",
                      type = "NUC")
}
```

getOGRDB	<i>Download and Load Immune Receptor Germline Sequences from OGRDB (AIRR)</i>
----------	---

Description

Downloads AIRR-compliant germline sets (or FASTA) from OGRDB and returns sequences as 'DNAStringSet' (NUC) or attempts AA translation for V genes.

Usage

```
getOGRDB(
  species = "human",
  locus = c("IGH", "IGK", "IGL"),
  set_name = NULL,
  type = c("NUC", "PROT"),
  format = c("FASTA_GAPPED", "FASTA_UNGAPPED", "AIRR"),
```

```

    version = c("published", "latest"),
    species_subgroup = NULL,
    refresh = FALSE,
    suppressMessages = FALSE
  )

```

Arguments

species	Species string. Accepts "human"/"Homo sapiens"/"mouse"/"Mus musculus".
locus	Either a locus short code ("IGH","IGK","IGL", etc.) OR NULL if you pass a 'set_name' explicitly.
set_name	Optional explicit OGRDB set name (e.g., "IGH_VDJ"). If provided, overrides 'locus'.
type	"NUC" (default) or "PROT". PROT will translate V-gene CDS; only supported for FASTA or AIRR records that include a valid CDS.
format	"FASTA_GAPPED", "FASTA_UNGAPPED", or "AIRR". Default "FASTA_GAPPED".
version	"published" (default), "latest", or a specific revision number as character/number.
species_subgroup	Optional subgroup (e.g., a mouse strain like "C57BL/6"). If it contains '/', OGRDB requires it encoded as "%252f".
refresh	If TRUE, redownload even if cached.
suppressMessages	If TRUE, be quiet.

Value

'DNAStringSet' for NUC; if 'type="PROT"', returns 'AAStringSet' where possible.

Examples

```

if (is_ogrdb_available()) {
  # Download human IGH nucleotide sequences (gapped FASTA)
  igh_nuc <- getOGRDB(species = "human",
                     locus   = "IGH",
                     type    = "NUC",
                     format  = "FASTA_GAPPED")

  # Download human IGK sequences in AIRR JSON format
  igk_airr <- getOGRDB(species = "human",
                     locus   = "IGK",
                     type    = "NUC",
                     format  = "AIRR")

  # Download human IGL sequences and translate to AA
  igl_prot <- getOGRDB(species = "human",
                     locus   = "IGL",
                     type    = "PROT",
                     format  = "FASTA_UNGAPPED")
}

```

```
# Example using an explicit set name (instead of locus)
igh_explicit <- getOGRDB(species = "human",
                        set_name = "IGH_VDJ",
                        type      = "NUC",
                        format    = "FASTA_GAPPED")
}
```

is_imgt_available	<i>Check if IMGT Website is Available</i>
-------------------	---

Description

This function sends a lightweight HEAD request to the main IMGT page to check if the service is online and accessible.

Usage

```
is_imgt_available()
```

Value

A logical value: 'TRUE' if the IMGT website is accessible, 'FALSE' otherwise.

Examples

```
is_imgt_available()
```

is_ogrdb_available	<i>Check if OGRDB Website is Available</i>
--------------------	--

Description

This function sends a lightweight HEAD request to the main OGRDB page to check if the service is online and accessible.

Usage

```
is_ogrdb_available()
```

Value

A logical value: 'TRUE' if the IMGT website is accessible, 'FALSE' otherwise.

Examples

```
is_imgt_available()
```

listIMGT	<i>List Datasets in the Local Cache</i>
----------	---

Description

Scans the cache directory and returns a list of available datasets.

Usage

```
listIMGT()
```

Value

A character vector of file paths for the cached datasets.

Examples

```
# List all files in the cache
cached_files <- listIMGT()

# To see the structure, you can print the first few
head(cached_files)
```

listOGRDB	<i>List OGRDB Datasets in Local Cache</i>
-----------	---

Description

Scans the cache directory and returns a list of available datasets.

Usage

```
listOGRDB()
```

Value

A character vector of absolute file paths to cached OGRDB files (length zero if none). Paths are typically under the package cache directory (e.g., `file.path(.get_cache_dir(), "<Species>", "ogrdb")`).

Examples

```
if (is_ogrdb_available()) {
  # List cached OGRDB files
  cached_files <- listOGRDB()
  head(cached_files)
}
```

loadIMGT

*Load Cached IMGT/HLA Sequences***Description**

Loads sequences from the local cache without attempting to download. This function relies on ‘getIMGT(refresh = FALSE)’. If the data is not found in the cache, it will be downloaded unless an internet connection is unavailable.

Usage

```
loadIMGT(
  species = "human",
  gene,
  type = c("NUC", "PROT"),
  suppressMessages = FALSE
)
```

Arguments

species	The species for which to download data. Required for TCR/BCR queries. Currently supported: "human", "mouse", "rat", "rabbit", "pig", "dog", "rhesus_monkey", "cyno monkey". Defaults to "human" for HLA.
gene	The gene or locus to download. For TCR/BCR, this can be a specific chain (e.g., "IGHV", "TRBJ") or a group (e.g., "IGH", "TCR"). For HLA, use "HLA".
type	The type of sequence to retrieve. Either "NUC" for nucleotide or "PROT" for protein sequences. This primarily distinguishes between VDJ nucleotide and V-region amino acid sequences for TCR/BCR genes.
suppressMessages	Logical. If ‘TRUE’, suppresses the license and other informational messages.

Value

A ‘DNASet’ or ‘AAStringSet’ object.

Examples

```
if(is_imgt_available()) {
  # First, download a file to ensure it's in the cache
  getIMGT(species = "human", gene = "IGHV", type = "NUC", suppressMessages = TRUE)
  # Now, load it from the cache
  ighv_cached <- loadIMGT(species = "human", gene = "IGHV", type = "NUC")
}
```

loadOGRDB	<i>Load Cached OGRDB Sequences</i>
-----------	------------------------------------

Description

Loads sequences from the local cache without attempting to download. This function relies on 'getOGRDB(refresh = FALSE)'. If the data is not found in the cache, it will be downloaded unless an internet connection is unavailable.

Usage

```
loadOGRDB(
  species = "human",
  locus = c("IGH", "IGK", "IGL"),
  set_name = NULL,
  type = c("NUC", "PROT"),
  format = c("FASTA_GAPPED", "FASTA_UNGAPPED", "AIRR"),
  version = c("published", "latest"),
  species_subgroup = NULL,
  suppressMessages = FALSE
)
```

Arguments

species	Species string. Accepts "human"/"Homo sapiens"/"mouse"/"Mus musculus".
locus	Either a locus short code ("IGH","IGK","IGL", etc.) OR NULL if you pass a 'set_name' explicitly.
set_name	Optional explicit OGRDB set name (e.g., "IGH_VDJ"). If provided, overrides 'locus'.
type	"NUC" (default) or "PROT". PROT will translate V-gene CDS; only supported for FASTA or AIRR records that include a valid CDS.
format	"FASTA_GAPPED", "FASTA_UNGAPPED", or "AIRR". Default "FASTA_GAPPED".
version	"published" (default), "latest", or a specific revision number as character/number.
species_subgroup	Optional subgroup (e.g., a mouse strain like "C57BL/6"). If it contains '/', OGRDB requires it encoded as "%252f".
suppressMessages	If TRUE, be quiet.

Value

The same object type as getOGRDB(): a DNASTringSet (when type = "NUC") or an AAStringSet (when type = "PROT"), loaded from the local cache if present (and downloaded on first use if needed).

Examples

```
if (is_ogrdb_available()) {
  # First, ensure the file is cached
  getOGRDB(species = "human", locus = "IGH",
            type = "NUC", format = "FASTA_GAPPED",
            suppressMessages = TRUE)

  # Now load from cache only
  igh_cached <- loadOGRDB(species = "human",
                        locus   = "IGH",
                        type    = "NUC",
                        format   = "FASTA_GAPPED")
}
```

refreshIMGT

Force Re-download of IMGT/HLA Sequences

Description

A convenience wrapper for ‘getIMGT(..., refresh = TRUE)’ to ensure that the local cache is updated with the latest versions of the requested sequences.

Usage

```
refreshIMGT(
  species = "human",
  gene,
  type = c("NUC", "PROT"),
  suppressMessages = FALSE
)
```

Arguments

species	The species for which to download data. Required for TCR/BCR queries. Currently supported: "human", "mouse", "rat", "rabbit", "pig", "dog", "rhesus_monkey", "cyno monkey". Defaults to "human" for HLA.
gene	The gene or locus to download. For TCR/BCR, this can be a specific chain (e.g., "IGHV", "TRBJ") or a group (e.g., "IGH", "TCR"). For HLA, use "HLA".
type	The type of sequence to retrieve. Either "NUC" for nucleotide or "PROT" for protein sequences. This primarily distinguishes between VDJ nucleotide and V-region amino acid sequences for TCR/BCR genes.
suppressMessages	Logical. If ‘TRUE’, suppresses the license and other informational messages.

Value

A ‘DNAStrngSet’ or ‘AAStringSet’ object.

Examples

```
if(is_imgt_available()) {
  # Force a re-download of human IGHV protein sequences
  ighv_prot_fresh <- refreshIMGT(species = "human", gene = "IGHV", type = "PROT")
}
```

refreshOGRDB

Force Re-download of OGRDB Sequences

Description

A convenience wrapper for ‘getOGRDB(..., refresh = TRUE)’ to ensure that the local cache is updated with the latest versions of the requested sequences.

Usage

```
refreshOGRDB(
  species = "human",
  locus = c("IGH", "IGK", "IGL"),
  set_name = NULL,
  type = c("NUC", "PROT"),
  format = c("FASTA_GAPPED", "FASTA_UNGAPPED", "AIRR"),
  version = c("published", "latest"),
  species_subgroup = NULL,
  suppressMessages = FALSE
)
```

Arguments

species	Species string. Accepts "human"/"Homo sapiens"/"mouse"/"Mus musculus".
locus	Either a locus short code ("IGH","IGK","IGL", etc.) OR NULL if you pass a ‘set_name’ explicitly.
set_name	Optional explicit OGRDB set name (e.g., "IGH_VDJ"). If provided, overrides ‘locus’.
type	"NUC" (default) or "PROT". PROT will translate V-gene CDS; only supported for FASTA or AIRR records that include a valid CDS.
format	"FASTA_GAPPED", "FASTA_UNGAPPED", or "AIRR". Default "FASTA_GAPPED".
version	"published" (default), "latest", or a specific revision number as character/number.
species_subgroup	Optional subgroup (e.g., a mouse strain like "C57BL/6"). If it contains ‘/’, OGRDB requires it encoded as "%252f".
suppressMessages	If TRUE, be quiet.

Index

* **package**

immReferent-package, [2](#)

getIMGT, [3](#)

getOGRDB, [4](#)

immReferent (immReferent-package), [2](#)

immReferent-package, [2](#)

is_imgt_available, [6](#)

is_ogrdb_available, [6](#)

listIMGT, [7](#)

listOGRDB, [7](#)

loadIMGT, [8](#)

loadOGRDB, [9](#)

refreshIMGT, [10](#)

refreshOGRDB, [11](#)