Package: immReferent (via r-universe)

November 2, 2026

Title An Interface for Immune Receptor and HLA Gene Reference Data

Version 1.0.0

Description Provides a consistent interface for downloading, storing, and accessing immune receptor (TCR/BCR) and HLA sequences from IMGT, IPD-IMGT/HLA, and OGRDB (AIRR-C). Supports export to popular analysis tools including MiXCR, TRUST4, Cell Ranger, and IgBLAST. This package serves as a core dependency for immunogenomics packages, ensuring reliable and high-quality sequence access with local caching for reproducibility.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

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

Config/pak/sysreqs libxml2-dev libssl-dev zlib1g-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 13:05:59 UTC

RemoteUrl https://github.com/bioc/immReferent

RemoteRef RELEASE_3_23

RemoteSha 5f85d75251beb23f48b480f4f542a8d36696999b

Contents

immReferent-package . . . 2
exportCellRanger . . . 3
exportIgBLAST . . . 5
exportMiXCR . . . 6
exportTRUST4 . . . 8
getIMGT . . . 9
getOGRDB . . . 10
is_imgt_available . . . 12
is_ogrdb_available . . . 13
listIMGT . . . 14
listOGRDB . . . 14
loadIMGT . . . 15
loadOGRDB . . . 16
refreshIMGT . . . 17
refreshOGRDB . . . 19

Index **21**

`immReferent-package` *immReferent: An Interface for Immune Receptor and HLA Gene Reference Data*

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
- Export to popular analysis tools (MiXCR, TRUST4, Cell Ranger, IgBLAST)
- Interoperability with Bioconductor classes such as `DNAStringSet` and `AAStringSet`

Data retrieval functions

- `getIMGT`: Download sequences from IMGT
- `getOGRDB`: Download sequences from OGRDB

- loadIMGT, loadOGRDB: Load cached sequences
- refreshIMGT, refreshOGRDB: Force re-download

Export functions

- exportMiXCR: Export for MiXCR analysis
- exportTRUST4: Export for TRUST4 analysis
- exportCellRanger: Export for 10x Cell Ranger VDJ
- exportIgBLAST: Export for IgBLAST analysis

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.

Author(s)

Maintainer: Nick Borcherding <ncborch@gmail.com>

See Also

https://github.com/BorchLab/immReferent

---

exportCellRanger *Export Reference Sequences to Cell Ranger VDJ Format*

Description

Exports a DNAStringSet to FASTA format suitable for creating a custom Cell Ranger VDJ reference. The function generates a FASTA file with properly formatted headers for use with `cellranger mkvdjref`.

Usage

```
exportCellRanger(sequences, output_file, gene_type = NULL)
```

Arguments

sequences	A DNAStringSet object containing immune receptor sequences. Sequence names must follow standard IG/TR gene nomenclature (e.g., "IGHV1-2*01"). Can be obtained from getIMGT or getOGRDB.
output_file	Character string specifying the path to the output FASTA file. The parent directory will be created if it does not exist.
gene_type	Character string specifying the type of gene region. One of "V", "D", "J", or "C". If NULL (default), the function will attempt to infer the type from sequence names.

Details

Cell Ranger's mkvdjref command expects FASTA files with specific header formats. This function creates a FASTA file that can be used as input to build a custom VDJ reference.

Note: For a complete Cell Ranger VDJ reference, you also need a GTF file with gene annotations. This function only creates the FASTA component.

This function works with sequences from both **IMGT** (via getIMGT) and **OGRDB** (via getOGRDB).

Value

Character string with the path to the created file, returned invisibly.

See Also

getIMGT, getOGRDB for obtaining sequences

exportMiXCR, exportTRUST4, exportIgBLAST for other export formats

https://www.10xgenomics.com/support/software/cell-ranger/latest/analysis/inputs/cr-5p-references for Cell Ranger documentation

Examples

```
# Create a small example DNAStringSet
seqs <- Biostrings::DNAStringSet(c(
  "ATGCGATCGATCGATCG",
  "ATGCGATCGATCG"
))
names(seqs) <- c("IGHV1-2*01", "IGHV1-3*01")

# Export to temporary file
output_file <- tempfile(fileext = ".fa")
exportCellRanger(seqs, output_file)

# View the result
cat(readLines(output_file), sep = "\n")

# Clean up
unlink(output_file)
```

exportIgBLAST *Export Reference Sequences to IgBLAST Format*

Description

Exports a DNAStringSet to FASTA files formatted for use with IgBLAST. The function creates separate FASTA files for V, D, and J gene segments with simplified headers compatible with IgBLAST's requirements.

Usage

```
exportIgBLAST(
  sequences,
  output_dir,
  organism = "custom",
  receptor_type = c("ig", "tcr")
)
```

Arguments

sequences	A DNAStringSet object containing immune receptor sequences. Sequence names must follow standard IG/TR gene nomenclature (e.g., "IGHV1-2*01"). Can be obtained from getIMGT or getOGRDB.
output_dir	Character string specifying the directory where output files will be written. The directory will be created if it does not exist.
organism	Character string specifying the organism name for the output files. Used in file naming. Default is "custom".
receptor_type	Character string specifying the receptor type. One of "ig" for immunoglobulin or "tcr" for T-cell receptor. Default is "ig".

Details

IgBLAST requires FASTA files with simplified headers containing only the gene/allele name. This function mimics the output of IgBLAST's edit_imgt_file.pl script, which truncates IMGT headers to keep only the allele designation.

Output files follow the naming convention used by IgBLAST:

- <organism>_<receptor_type>_v.fasta
- <organism>_<receptor_type>_d.fasta
- <organism>_<receptor_type>_j.fasta

After exporting, use makeblastdb with the -parse_seqids flag to create the BLAST database:

```
makeblastdb -parse_seqids -dbtype nucl -in <fasta_file> -out <db_name>
```

This function works with sequences from both **IMGT** (via getIMGT) and **OGRDB** (via getOGRDB).

Value

A named list containing the paths to the created files, returned invisibly. The list may contain elements `v_genes`, `d_genes`, and `j_genes` depending on which segment types were found in the input sequences.

See Also

`getIMGT`, `getOGRDB` for obtaining sequences

`exportMiXCR`, `exportTRUST4`, `exportCellRanger` for other export formats

`https://ncbi.github.io/igblast/` for IgBLAST documentation

Examples

```
# Create a small example DNAStringSet
seqs <- Biostrings::DNAStringSet(c(
  "ATGCGATCGATCGATCG",
  "ATGCGATCGATCG",
  "ATGCGATC"
))
names(seqs) <- c("IGHV1-2*01", "IGHD1-1*01", "IGHJ1*01")

# Export to temporary directory
output_dir <- tempdir()
files <- exportIgBLAST(seqs, output_dir, organism = "human", receptor_type = "ig")
print(files)

# Clean up
unlink(unlist(files))
```

---

exportMiXCR *Export Reference Sequences to MiXCR Format*

Description

Exports a `DNAStringSet` or `AAStringSet` to FASTA files formatted for use with MiXCR's `buildLibrary` command. The function creates separate FASTA files for V, D, J, and C gene segments.

Usage

```
exportMiXCR(
  sequences,
  output_dir,
  chain = c("IGH", "IGK", "IGL", "TRA", "TRB", "TRD", "TRG")
)
```

Arguments

sequences	A DNAStringSet or AAStringSet object containing immune receptor sequences. Sequence names must follow standard IG/TR gene nomenclature (e.g., "IGHV1-2*01", "TRBJ2-1*01"). Can be obtained from getIMGT or getOGRDB.
output_dir	Character string specifying the directory where output files will be written. The directory will be created if it does not exist.
chain	Character string specifying the chain type for the output files. Must be one of "IGH", "IGK", "IGL", "TRA", "TRB", "TRD", or "TRG".

Details

MiXCR expects FASTA files with simple headers containing only the gene name. The function filters sequences by gene type (V, D, J, C) based on the gene name pattern and writes separate files for each segment type.

Output files follow the naming convention:

- v-genes.<chain>.fasta
- d-genes.<chain>.fasta
- j-genes.<chain>.fasta
- c-genes.<chain>.fasta

This function works with sequences from both **IMGT** (via getIMGT) and **OGRDB** (via getOGRDB).

Value

A named list containing the paths to the created files, returned invisibly. The list may contain elements v_genes, d_genes, j_genes, and c_genes depending on which segment types were found in the input sequences.

See Also

getIMGT, getOGRDB for obtaining sequences

exportTRUST4, exportCellRanger, exportIgBLAST for other export formats

https://mixcr.com/mixcr/guides/create-custom-library/ for MiXCR documentation

Examples

```
# Create a small example DNAStringSet
seqs <- Biostrings::DNAStringSet(c(
  "ATGCGATCGATCGATCG",
  "ATGCGATCGATCG",
  "ATGCGATC",
  "ATGCGATCGATCGATCGATCG"
))
names(seqs) <- c("IGHV1-2*01", "IGHD1-1*01", "IGHJ1*01", "IGHC*01")

# Export to temporary directory
output_dir <- tempdir()
```

```
files <- exportMiXCR(seqs, output_dir, chain = "IGH")
print(files)

# Clean up
unlink(unlist(files))
```

`exportTRUST4` *Export Reference Sequences to TRUST4 Format*

Description

Exports a `DNAStringSet` to a FASTA file formatted for use with TRUST4. The output follows the format produced by TRUST4's `BuildImgtAnnot.pl` script.

Usage

```
exportTRUST4(sequences, output_file, include_constant = TRUE)
```

Arguments

`sequences`	A `DNAStringSet` object containing immune receptor sequences. Sequence names must follow standard IG/TR gene nomenclature (e.g., `"IGHV1-2*01"`). Can be obtained from `getIMGT` or `getOGRDB`.
`output_file`	Character string specifying the path to the output FASTA file. The parent directory will be created if it does not exist.
`include_constant`	Logical. If `TRUE` (default), include constant region sequences. TRUST4's `IMGT+C.fa` file includes constant regions.

Details

TRUST4 expects FASTA files with headers containing only the allele name (e.g., `>IGHV1-2*01`). The function reformats sequence headers to match the output of TRUST4's `BuildImgtAnnot.pl` script.

TRUST4 uses this reference for the `--ref` parameter in its analysis pipeline.

This function works with sequences from both **IMGT** (via `getIMGT`) and **OGRDB** (via `getOGRDB`).

Value

Character string with the path to the created file, returned invisibly.

See Also

`getIMGT`, `getOGRDB` for obtaining sequences

`exportMiXCR`, `exportCellRanger`, `exportIgBLAST` for other export formats

`https://github.com/liulab-dfci/TRUST4` for TRUST4 documentation

Examples

```
# Create a small example DNAStringSet
seqs <- Biostrings::DNAStringSet(c(
  "ATGCGATCGATCGATCG",
  "ATGCGATCGATCG",
  "ATGCGATC"
))
names(seqs) <- c("IGHV1-2*01", "IGHJ1*01", "IGHC*01")

# Export to temporary file
output_file <- tempfile(fileext = ".fa")
exportTRUST4(seqs, output_file)

# View the result
cat(readLines(output_file), sep = "\n")

# Clean up
unlink(output_file)
```

`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`	Character string specifying the species for which to download data. Required for TCR/BCR queries. Currently supported species: `"human"`, `"mouse"`, `"rat"`, `"rabbit"`, `"pig"`, `"dog"`, `"rhesus_monkey"`, `"cyno_monkey"`. Defaults to `"human"` for HLA queries.
`gene`	Character string specifying 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 — Character string specifying 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. Default is FALSE.

suppressMessages — Logical. If TRUE, suppresses the license and other informational messages. Default is FALSE.

Value

A DNAStringSet object (when type = "NUC") or AAStringSet object (when type = "PROT") containing the requested sequences.

See Also

loadIMGT, refreshIMGT for convenience wrappers

getOGRDB for OGRDB/AIRR-C germline sequences

exportMiXCR, exportTRUST4, exportCellRanger, exportIgBLAST for exporting sequences to analysis tools

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 — *Download and Load Immune Receptor Germline Sequences from OGRDB*

Description

Downloads AIRR-compliant germline sets (or FASTA) from OGRDB (Open Germline Receptor Database) and returns sequences as a DNAStringSet or AAStringSet.

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	Character string specifying the species. Accepts "human", "Homo sapiens", "mouse", or "Mus musculus". Default is "human".
locus	Character string specifying the locus short code. One of "IGH", "IGK", or "IGL". Can be NULL if you pass a set_name explicitly.
set_name	Optional character string specifying an explicit OGRDB set name (e.g., "IGH_VDJ"). If provided, overrides locus.
type	Character string specifying the sequence type. Either "NUC" (default) for nucleotide or "PROT" for protein. "PROT" will translate V-gene CDS; only supported for FASTA or AIRR records that include a valid CDS.
format	Character string specifying the download format. One of "FASTA_GAPPED" (default), "FASTA_UNGAPPED", or "AIRR".
version	Character string specifying the version. Either "published" (default) or "latest".
species_subgroup	Optional character string specifying a subgroup (e.g., a mouse strain like "C57BL/6"). If it contains /, OGRDB requires it encoded as %252f.
refresh	Logical. If TRUE, forces re-download even if cached. Default is FALSE.
suppressMessages	Logical. If TRUE, suppresses informational messages. Default is FALSE.

Details

OGRDB (Open Germline Receptor Database) is the AIRR Community's curated repository of germline receptor sequences. It complements IMGT with additional species support and standardized AIRR JSON format.

The function supports multiple download formats:

- FASTA_GAPPED: FASTA with IMGT gaps preserved
- FASTA_UNGAPPED: FASTA without gaps
- AIRR: AIRR-C compliant JSON format

Value

A DNAStringSet object (when type = "NUC") or AAStringSet object (when type = "PROT") containing the requested sequences.

See Also

loadOGRDB, refreshOGRDB for convenience wrappers

getIMGT for IMGT sequences

exportMiXCR, exportTRUST4, exportCellRanger, exportIgBLAST for exporting sequences to analysis tools

https://ogrdb.airr-community.org/ for OGRDB documentation

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 *Check if IMGT Website is Available*

Description

Sends a lightweight HEAD request to the main IMGT page to check if the service is online and accessible. This function is used to conditionally run examples and tests that require an internet connection.

Usage

```
is_imgt_available()
```

Value

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

See Also

is_ogrdb_available for checking OGRDB availability

getIMGT which uses this function

Examples

```
is_imgt_available()
```

---

is_ogrdb_available *Check if OGRDB Website is Available*

Description

Sends a lightweight HEAD request to the OGRDB API to check if the service is online and accessible. This function is used to conditionally run examples and tests that require an internet connection.

Usage

```
is_ogrdb_available()
```

Value

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

See Also

is_imgt_available for checking IMGT availability

getOGRDB which uses this function

Examples

```
is_ogrdb_available()
```

listIMGT *List Datasets in the Local Cache*

Description

Scans the cache directory and returns a list of available datasets that have been downloaded.

Usage

```
listIMGT()
```

Value

A character vector of absolute file paths for the cached datasets. Returns an empty character vector if the cache directory does not exist or contains no files.

See Also

`getIMGT` for downloading sequences

`listOGRDB` for listing OGRDB cached files

Examples

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

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

listOGRDB *List OGRDB Datasets in Local Cache*

Description

Scans the cache directory and returns a list of available OGRDB datasets that have been downloaded.

Usage

```
listOGRDB()
```

Value

A character vector of absolute file paths to cached OGRDB files. Returns an empty character vector if no OGRDB files have been cached. Paths are typically under the package cache directory (e.g., `~/.immReferent/<Species>/ogrdb/`).

See Also

getOGRDB for downloading sequences

listIMGT for listing IMGT cached files

Examples

```
# 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 is a convenience wrapper for `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	Character string specifying the species for which to download data. Required for TCR/BCR queries. Currently supported species: `"human"`, `"mouse"`, `"rat"`, `"rabbit"`, `"pig"`, `"dog"`, `"rhesus_monkey"`, `"cyno_monkey"`. Defaults to `"human"` for HLA queries.
gene	Character string specifying 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	Character string specifying 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. Default is `FALSE`.

Value

A DNAStringSet object (when type = "NUC") or AAStringSet object (when type = "PROT") containing the requested sequences.

See Also

getIMGT for the main download function

refreshIMGT to force re-download

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 *Load Cached OGRDB Sequences*

Description

Loads sequences from the local cache without attempting to download. This function is a convenience wrapper for 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	Character string specifying the species. Accepts "human", "Homo sapiens", "mouse", or "Mus musculus". Default is "human".
locus	Character string specifying the locus short code. One of "IGH", "IGK", or "IGL". Can be NULL if you pass a set_name explicitly.

`set_name` Optional character string specifying an explicit OGRDB set name (e.g., `"IGH_VDJ"`). If provided, overrides `locus`.

`type` Character string specifying the sequence type. Either `"NUC"` (default) for nucleotide or `"PROT"` for protein. `"PROT"` will translate V-gene CDS; only supported for FASTA or AIRR records that include a valid CDS.

`format` Character string specifying the download format. One of `"FASTA_GAPPED"` (default), `"FASTA_UNGAPPED"`, or `"AIRR"`.

`version` Character string specifying the version. Either `"published"` (default) or `"latest"`.

`species_subgroup` Optional character string specifying a subgroup (e.g., a mouse strain like `"C57BL/6"`). If it contains `/`, OGRDB requires it encoded as `%252f`.

`suppressMessages` Logical. If `TRUE`, suppresses informational messages. Default is `FALSE`.

Value

A `DNAStringSet` object (when `type = "NUC"`) or `AAStringSet` object (when `type = "PROT"`) containing the requested sequences.

See Also

`getOGRDB` for the main download function

`refreshOGRDB` to force re-download

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	Character string specifying the species for which to download data. Required for TCR/BCR queries. Currently supported species: "human", "mouse", "rat", "rabbit", "pig", "dog", "rhesus_monkey", "cyno_monkey". Defaults to "human" for HLA queries.
gene	Character string specifying 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	Character string specifying 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. Default is FALSE.

Value

A DNAStringSet object (when type = "NUC") or AAStringSet object (when type = "PROT") containing the requested sequences.

See Also

getIMGT for the main download function

loadIMGT to load from cache without downloading

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`	Character string specifying the species. Accepts `"human"`, `"Homo sapiens"`, `"mouse"`, or `"Mus musculus"`. Default is `"human"`.
`locus`	Character string specifying the locus short code. One of `"IGH"`, `"IGK"`, or `"IGL"`. Can be `NULL` if you pass a `set_name` explicitly.
`set_name`	Optional character string specifying an explicit OGRDB set name (e.g., `"IGH_VDJ"`). If provided, overrides `locus`.
`type`	Character string specifying the sequence type. Either `"NUC"` (default) for nucleotide or `"PROT"` for protein. `"PROT"` will translate V-gene CDS; only supported for FASTA or AIRR records that include a valid CDS.
`format`	Character string specifying the download format. One of `"FASTA_GAPPED"` (default), `"FASTA_UNGAPPED"`, or `"AIRR"`.
`version`	Character string specifying the version. Either `"published"` (default) or `"latest"`.
`species_subgroup`	Optional character string specifying a subgroup (e.g., a mouse strain like `"C57BL/6"`). If it contains `/`, OGRDB requires it encoded as `%252f`.
`suppressMessages`	Logical. If `TRUE`, suppresses informational messages. Default is `FALSE`.

Value

A `DNAStringSet` object (when `type = "NUC"`) or `AAStringSet` object (when `type = "PROT"`) containing the requested sequences.

See Also

getOGRDB for the main download function

loadOGRDB to load from cache without downloading

Examples

```
if (is_ogrdb_available()) {
  # Force a re-download of the human IGK sequences
  igk_fresh <- refreshOGRDB(species = "human",
                            locus   = "IGK",
                            type    = "NUC",
                            format  = "FASTA_GAPPED")
}
```

Index

∗ **package**
immReferent-package, 2

AAStringSet, *2*, *6, 7*, *10*, *12*, *16–19*

DNAStringSet, *2–8*, *10*, *12*, *16–19*

exportCellRanger, 3, *3*, *6–8*, *10*, *12*
exportIgBLAST, *3, 4*, 5, *7, 8*, *10*, *12*
exportMiXCR, *3, 4*, 6, *6*, *8*, *10*, *12*
exportTRUST4, *3, 4*, *6, 7*, 8, *10*, *12*

getIMGT, *2*, *4–8*, 9, *12–14*, *16*, *18*
getOGRDB, *2*, *4–8*, 10, *10*, *13*, *15*, *17*, *20*

immReferent *(*immReferent-package*)*, 2
immReferent-package, 2
is_imgt_available, 12, *13*
is_ogrdb_available, 13, *13*

listIMGT, 14, *15*
listOGRDB, 14, *14*
loadIMGT, *3*, *10*, 15, *18*
loadOGRDB, *3*, *12*, 16, *20*

refreshIMGT, *3*, *10*, *16*, 17
refreshOGRDB, *3*, *12*, *17*, 19