

# Package ‘rigvf’

September 10, 2025

**Title** R interface to the IGVF Catalog

**Version** 1.1.3

**Description** The IGVF Catalog provides data on the impact of genomic variants on function. The ‘rigvf’ package provides an interface to the IGVF Catalog, allowing easy integration with Bioconductor resources.

**License** MIT + file LICENSE

**Depends** R (>= 4.1.0)

**Imports** methods, httr2, rjsoncons, dplyr, tidyr, rlang, memoise, cachem, whisker, jsonlite, GenomicRanges, IRanges, Seqinfo

**URL** <https://IGVF.github.io/rigvf>

**BugReports** <https://github.com/IGVF/rigvf/issues>

**Encoding** UTF-8

**biocViews** ThirdPartyClient, Annotation, VariantAnnotation, FunctionalGenomics, GeneRegulation, GenomicVariation, GeneTarget

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), plyranges, plotgardener, org.Hs.eg.db, TxDb.Hsapiens.UCSC.hg38.knownGene

**VignetteBuilder** knitr

**Config/testthat/edition** 3

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

**git\_branch** devel

**git\_last\_commit** 865002e

**git\_last\_commit\_date** 2025-07-24

**Repository** Bioconductor 3.22

**Date/Publication** 2025-09-09

**Author** Martin Morgan [aut] (ORCID: <<https://orcid.org/0000-0002-5874-8148>>),  
Michael Love [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-8401-0545>>),  
NIH NHGRI UM1HG012003 [fnd]

**Maintainer** Michael Love <michaelisaiahlove@gmail.com>

## Contents

|                           |          |
|---------------------------|----------|
| arango . . . . .          | 2        |
| catalog_queries . . . . . | 3        |
| db_queries . . . . .      | 5        |
| rigvf . . . . .           | 7        |
| <b>Index</b>              | <b>8</b> |

---

|        |                                                |
|--------|------------------------------------------------|
| arango | <i>Internal interface to ArangoDB REST API</i> |
|--------|------------------------------------------------|

---

## Description

Objects documented on this page are for developer use. `arango_request()` formulates 'GET' or 'POST' to the ArangoDB API.

`arango_auth()` uses username and password to authenticate against the database.

`arango_collections()` implements the `_api/collection` endpoint for available collections in the IGVF database.

`arango_cursor()` implements the `_api/cursor` endpoint to allow a user-specified query of the IGVF database.

## Usage

```
arango_request(path, ..., body = NULL, jwt_token = NULL)
```

```
arango_auth(
  username = rigvf_config$get("username"),
  password = rigvf_config$get("password")
)
```

```
arango_collections(
  username = rigvf_config$get("username"),
  password = rigvf_config$get("password")
)
```

```
arango_cursor(
  query,
  ...,
  username = rigvf_config$get("username"),
  password = rigvf_config$get("password")
)
```

## Arguments

|      |                                                                                                                                                                                          |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| path | character(1) path to the API end point. Note that database-specific paths are prefixed with <code>/_db/igvf</code> .                                                                     |
| ...  | for <code>arango_request()</code> , named arguments to be used as query parameters. for <code>arango_cursor()</code> , named arguments to be interpreted as bind variables in the query. |
| body | if not NULL, formulate a POST request with JSON body.                                                                                                                                    |

|           |                                                                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| jwt_token | character(1) JWT token obtained via arango_auth().                                                                                                                                                                              |
| username  | character(1) ArangoDB user name. Default: "guest".                                                                                                                                                                              |
| password  | character(1) ArangoDB password. Default: "guestigvfcatalog". A better practice is to use an environment variable to record the password, rather than encoding in a script, so password = Sys.getenv("RIGVF_ARANGODB_PASSWORD"). |
| query     | character(1) the FILE NAME (without extension .aql of the query template.                                                                                                                                                       |

### Details

arango\_auth() is 'memoized', so invoked only once per hour for a particular user and password. The memoised result can be cleared with memoise::forget(arango\_auth).

arango\_cursor() expects queries to be written in package system files in the inst/aql directory. This allows rapid iteration during query development (the package does not need to re-loaded when the query is updated) and some opportunity for language-specific highlighting if supported by the developer's text editor.

### Value

arango\_request() returns the JSON response as a character(1) vector.

arango\_auth() returns a JWT token to be used for authentication in subsequent calls.

arango\_collections() returns a tibble with columns name, type (either 'node' or 'edge'), and count.

arango\_cursor() returns the JSON character(1) 'result' of the query.

### Examples

```
## available queries
templates <- system.file(package = "rigvf", "aql")
dir(templates)
```

---

|                 |                                            |
|-----------------|--------------------------------------------|
| catalog_queries | <i>Query the IGVF Catalog via REST API</i> |
|-----------------|--------------------------------------------|

---

### Description

This page documents functions using the IGVF REST API, documented at <https://api.catalog.igvf.org/#>.

Note that functions will only return a limited number of responses, see limit and page arguments below for control over number of responses.

gene\_variants() locates variants associated with a gene. Only one of gene\_id, hgnc, gene\_name, or alias should be specified.

variant\_genes() locates genes associated with a variant. Only one of spdi, hgvs, rsid, variant\_id, or chr + position should be specified.

gene\_elements() locates elements associated with a gene.

elements() locates genomic elements based on a genomic range query.

element\_genes() locates genomic elements and associated genes based on a genomic range query.

**Usage**

```
gene_variants(
  gene_id = NULL,
  hgnc = NULL,
  gene_name = NULL,
  alias = NULL,
  organism = "Homo sapiens",
  log10pvalue = NULL,
  effect_size = NULL,
  page = 0L,
  limit = 25L,
  verbose = FALSE
)
```

```
variant_genes(
  spdi = NULL,
  hgvs = NULL,
  rsid = NULL,
  variant_id = NULL,
  chr = NULL,
  position = NULL,
  organism = "Homo sapiens",
  log10pvalue = NULL,
  effect_size = NULL,
  page = 0L,
  limit = 25L,
  verbose = FALSE
)
```

```
gene_elements(gene_id = NULL, page = 0L, limit = 25L, verbose = FALSE)
```

```
elements(range = NULL, page = 0L, limit = 25L)
```

```
element_genes(range = NULL, page = 0L, limit = 25L, verbose = FALSE)
```

**Arguments**

|             |                                                                                                                                                                            |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| gene_id     | character(1) Ensembl gene identifier, e.g., "ENSG00000106633"                                                                                                              |
| hgnc        | character(1) HGNC identifier                                                                                                                                               |
| gene_name   | character(1) Gene symbol, e.g., "GCK"                                                                                                                                      |
| alias       | character(1) Gene alias                                                                                                                                                    |
| organism    | character(1) Either 'Homo sapiens' (default) or 'Mus musculus'                                                                                                             |
| log10pvalue | character(1) The following can be used to set thresholds on the negative log10pvalue: gt (>), gte (>=), lt (<), lte (<=), with a ":" following and a value, e.g., "gt:5.0" |
| effect_size | character(1) Optional string used for thresholding on the effect size of the variant on the gene. See 'log10pvalue'. E.g., "gt:0.5"                                        |
| page        | integer(1) when there are more response items than limit, offers pagination. starts on page 0L, next is 1L, ...                                                            |
| limit       | integer(1) the limit parameter controls the page size and can not exceed 1000                                                                                              |
| verbose     | logical(1) return additional information about variants and genes                                                                                                          |

|            |                                                                 |
|------------|-----------------------------------------------------------------|
| spdi       | character(1) SPDI of variant                                    |
| hgvs       | character(1) HGVS of variant                                    |
| rsid       | character(1) RSID of variant                                    |
| variant_id | character(1) IGVF variant ID                                    |
| chr        | character(1) UCSC-style chromosome name of variant, e.g. "chr1" |
| position   | character(1) 0-based position of variant                        |
| range      | the query GRanges (expects 1-based start position)              |

### Value

gene\_variants() returns a tibble describing variants associated with the gene; use verbose = TRUE to retrieve more extensive information.

variant\_genes() returns a tibble describing genes associated with a variant; use verbose = TRUE to retrieve more extensive information.

gene\_elements() returns a tibble describing elements associated with the gene; use verbose = TRUE to retrieve more extensive information.

elements() returns a GRanges object describing elements.

element\_genes() returns a tibble describing genomic element and gene pairs.

### Examples

```

rigvf::gene_variants(gene_name = "GCK")

rigvf::gene_variants(gene_name = "GCK", effect_size="gt:0.5")

rigvf::gene_variants(gene_name = "GCK", verbose = TRUE)

rigvf::variant_genes(spdi = "NC_000001.11:920568:G:A")

res <- rigvf::gene_elements(gene_id = "ENSG00000187961")
res
res |>
  dplyr::select(elements) |>
  tidyr::unnest_wider(elements)

rng <- GenomicRanges::GRanges("chr1", IRanges::IRanges(1157520,1158189))

rigvf::elements(range = rng)

rigvf::element_genes(range = rng)

```

### Description

edges() and nodes() identify edges or nodes in the data base.

db\_gene\_variants() locates variants associated with a (Ensembl) gene identifier.

db\_gene\_elements() locates genomic elements associated with a (Ensembl) gene identifier.

**Usage**

```

db_edges(
  username = rigvf_config$get("username"),
  password = rigvf_config$get("password")
)

db_nodes(
  username = rigvf_config$get("username"),
  password = rigvf_config$get("password")
)

db_gene_variants(
  gene_id,
  threshold,
  username = rigvf_config$get("username"),
  password = rigvf_config$get("password")
)

db_gene_elements(
  gene_id,
  threshold,
  username = rigvf_config$get("username"),
  password = rigvf_config$get("password")
)

```

**Arguments**

|           |                                                                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| username  | character(1) ArangoDB user name. Default: "guest".                                                                                                                                                                              |
| password  | character(1) ArangoDB password. Default: "guestigvfcatalog". A better practice is to use an environment variable to record the password, rather than encoding in a script, so password = Sys.getenv("RIGVF_ARANGODB_PASSWORD"). |
| gene_id   | character(1) Ensembl gene identifier                                                                                                                                                                                            |
| threshold | numeric(1) minimum association statistic, minus log10 p-value for variants, and score for elements                                                                                                                              |

**Value**

edges() and nodes() return a tibble with the edge or node name and count of occurrences in the database.

db\_gene\_variants() returns a tibble summarizing variants associated with the gene.

db\_gene\_elements() returns a tibble summarizing genomic elements associated with the gene.

**Examples**

```

db_edges()

db_nodes()

db_gene_variants("ENSG00000106633", threshold = 4.0)

db_gene_elements("ENSG00000106633", threshold = 0.5)

```

---

`rigvf`*Package configuration global variable management*

---

**Description**

Objects documented on this page are for developer use.

`rigvf_config` provides a simple interface to manage 'package global' variables via `rigvf_config$get()`, `rigvf_config$set()`, etc. The default username and password provide guest access.

**Usage**

```
rigvf_config
```

**Format**

`rigvf_config` is a list of functions for listing (`ls()`) and manipulating (`get()`, `set()`, `unset()`) package-global variables.

# Index

## \* datasets

rigvf, [7](#)

arango, [2](#)

arango\_auth (arango), [2](#)

arango\_collections (arango), [2](#)

arango\_cursor (arango), [2](#)

arango\_request (arango), [2](#)

catalog\_queries, [3](#)

db\_edges (db\_queries), [5](#)

db\_gene\_elements (db\_queries), [5](#)

db\_gene\_variants (db\_queries), [5](#)

db\_nodes (db\_queries), [5](#)

db\_queries, [5](#)

element\_genes (catalog\_queries), [3](#)

elements (catalog\_queries), [3](#)

gene\_elements (catalog\_queries), [3](#)

gene\_variants (catalog\_queries), [3](#)

rigvf, [7](#)

rigvf\_config (rigvf), [7](#)

variant\_genes (catalog\_queries), [3](#)