

# Package ‘orthogene’

November 12, 2022

**Type** Package

**Title** Interspecies gene mapping

**Version** 1.5.1

**Description** `orthogene` is an R package for easy mapping of orthologous genes across hundreds of species. It pulls up-to-date gene ortholog mappings across **700+** organisms.

It also provides various utility functions to aggregate/expand common objects (e.g. data.frames, gene expression matrices, lists) using **1:1**, **many:1**, **1:many** or **many:many** gene mappings, both within- and between-species.

**URL** <https://github.com/neurogenomics/orthogene>

**BugReports** <https://github.com/neurogenomics/orthogene/issues>

**License** GPL-3

**Depends** R (>= 4.1)

**VignetteBuilder** knitr

**biocViews** Genetics, ComparativeGenomics, Preprocessing, Phylogenetics, Transcriptomics, GeneExpression

**Imports** dplyr,  
methods,  
stats,  
utils,  
Matrix,  
jsonlite,  
homologene,  
gprofiler2,  
babelgene,  
data.table,  
parallel,  
ggplot2,  
ggpubr,  
patchwork,  
DelayedArray,  
grr,  
repmis,  
ggtree,  
tools

**Suggests** remotes,  
 knitr,  
 BiocStyle,  
 covr,  
 markdown,  
 rmarkdown,  
 here,  
 testthat ( $\geq 3.0.0$ ),  
 piggyback,  
 badger,  
 magick,  
 desc,  
 hrbrthemes,  
 Cairo,  
 yulab.utils,  
 haven,  
 GenomeInfoDbData,  
 ape,  
 phytools,  
 rphylopic,  
 TreeTools,  
 RColorBrewer,  
 ggimage,  
 OmaDB

**RoxygenNote** 7.2.2

**Encoding** UTF-8

**Config/testthat/edition** 3

**Config/rcmdcheck/\_R\_CHECK\_FORCE\_SUGGESTS\_** false

## R topics documented:

orthogene-package . . . . .	3
aggregate_mapped_genes . . . . .	3
all_genes . . . . .	5
convert_orthologs . . . . .	7
create_background . . . . .	10
exp_mouse . . . . .	11
exp_mouse_enst . . . . .	12
gprofiler_namespace . . . . .	12
gprofiler_orgs . . . . .	13
infer_species . . . . .	13
map_genes . . . . .	15
map_orthologs . . . . .	16
map_species . . . . .	17
plot_orthotree . . . . .	18
prepare_tree . . . . .	21
report_orthologs . . . . .	22

<b>Index</b>	<b>25</b>
--------------	-----------

---

orthogene-package

**orthogene:** *Interspecies gene mapping*

---

## Description

**orthogene** is an R package for easy mapping of orthologous genes across hundreds of species.

## Details

It pulls up-to-date interspecies gene ortholog mappings across 700+ organisms.  
It also provides various utility functions to map common objects (e.g. data.frames, gene expression matrices, lists) onto 1:1 gene orthologs from any other species.

## Author(s)

**Maintainer:** Brian Schilder <brian\_schilder@alumni.brown.edu> ([ORCID](#))

## Source

- [GitHub](#) : Source code and Issues submission.
- [Author Site](#) : orthogene was created by Brian M. Schilder.

## See Also

Useful links:

- <https://github.com/neurogenomics/orthogene>
- Report bugs at <https://github.com/neurogenomics/orthogene/issues>

---

aggregate\_mapped\_genes

*Aggregate/expand a gene matrix by gene mappings*

---

## Description

Aggregate/expand a gene matrix (gene\_df) using a gene mapping [data.frame](#) (gene\_map). Importantly, mappings can be performed across a variety of scenarios that can occur during within-species and between-species gene mapping:

- 1 gene : 1 gene
- many genes : 1 gene
- 1 gene : many genes
- many genes : many genes

For more details on how aggregation/expansion is performed, please see: [many2many\\_rows](#).

## Usage

```
aggregate_mapped_genes(
  gene_df,
  gene_map = NULL,
  input_col = "input_gene",
  output_col = "ortholog_gene",
  input_species = "human",
  output_species = input_species,
  method = c("gprofiler", "homologene", "babelgene"),
  agg_fun = "sum",
  agg_method = c("monocle3", "stats"),
  aggregate_orthologs = TRUE,
  transpose = FALSE,
  mthreshold = 1,
  target = "ENSG",
  numeric_ns = "",
  as_integers = FALSE,
  as_sparse = TRUE,
  as_DelayedArray = FALSE,
  dropNA = TRUE,
  sort_rows = FALSE,
  verbose = TRUE
)
```

## Arguments

gene_df	Input matrix where row names are genes.
gene_map	<p>A <a href="#">data.frame</a> that maps the current gene names to new gene names. This function's behaviour will adapt to different situations as follows:</p> <ul style="list-style-type: none"> <li>• <code>gene_map=&lt;data.frame&gt;</code> : When a <code>data.frame</code> containing the gene key:value columns (specified by <code>input_col</code> and <code>output_col</code>, respectively) is provided, this will be used to perform aggregation/expansion.</li> <li>• <code>gene_map=NULL</code> and <code>input_species!=output_species</code> : A <code>gene_map</code> is automatically generated by <a href="#">map_orthologs</a> to perform inter-species gene aggregation/expansion.</li> <li>• <code>gene_map=NULL</code> and <code>input_species==output_species</code> : A <code>gene_map</code> is automatically generated by <a href="#">map_genes</a> to perform within-species gene symbol standardization and aggregation/expansion.</li> </ul>
input_col	Column name within <code>gene_map</code> with gene names matching the row names of X.
output_col	Column name within <code>gene_map</code> with gene names that you wish you map the row names of X onto.
input_species	Name of the input species (e.g., "mouse", "fly"). Use <a href="#">map_species</a> to return a full list of available species.
output_species	Name of the output species (e.g. "human", "chicken"). Use <a href="#">map_species</a> to return a full list of available species.
method	<p>R package to use for gene mapping:</p> <ul style="list-style-type: none"> <li>• "gprofiler" : Slower but more species and genes.</li> <li>• "homologene" : Faster but fewer species and genes.</li> </ul>

- "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.

agg_fun	Aggregation function.
agg_method	Aggregation method.
aggregate_orthologs	[Optional] After performing an initial round of many:many aggregation/expansion with <a href="#">many2many_rows</a> , ensure each orthologous gene only appears in one row by using the <a href="#">aggregate_rows</a> function (default: TRUE).
transpose	Transpose gene_df before mapping genes.
mthreshold	maximum number of results per initial alias to show. Shows all by default.
target	target namespace.
numeric_ns	namespace to use for fully numeric IDs ( <a href="#">list of available namespaces</a> ).
as_integers	Force all values in the matrix to become integers, by applying <a href="#">floor</a> (default: FALSE).
as_sparse	Convert aggregated matrix to sparse matrix.
as_DelayedArray	Convert aggregated matrix to <a href="#">DelayedArray</a> .
dropNA	Drop genes assigned to NA in groupings.
sort_rows	Sort gene_df rows alphanumerically.
verbose	Print messages.

## Value

Aggregated matrix

## Examples

```
#### Aggregate within species: gene synonyms ####
data("exp_mouse_enst")
X_agg <- aggregate_mapped_genes(gene_df = exp_mouse_enst,
                               input_species = "mouse")

#### Aggregate across species: gene orthologs ####
data("exp_mouse")
X_agg2 <- aggregate_mapped_genes(gene_df = exp_mouse,
                                input_species = "mouse",
                                output_species = "human",
                                method="homologene")
```

---

all\_genes

*Get all genes*


---

## Description

Return all known genes from a given species.

**Usage**

```
all_genes(
  species,
  method = c("gprofiler", "homologene", "babelgene"),
  ensure_filter_nas = FALSE,
  run_map_species = TRUE,
  verbose = TRUE,
  ...
)
```

**Arguments**

species	Species to get all genes for. Will first be standardised with <code>map_species</code> .
method	R package to use for gene mapping: <ul style="list-style-type: none"> <li>• "gprofiler" : Slower but more species and genes.</li> <li>• "homologene" : Faster but fewer species and genes.</li> <li>• "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.</li> </ul>
ensure_filter_nas	Perform an extra check to remove genes that are NAs of any kind.
run_map_species	Standardise species names with <a href="#">map_species</a> first (Default: TRUE).
verbose	Print messages.
...	Additional arguments to be passed to <a href="#">gorth</a> or <a href="#">homologene</a> .

*NOTE:* To return only the most "popular" interspecies ortholog mappings, supply `mtthreshold=1` here AND set `method="gprofiler"` above. This procedure tends to yield a greater number of returned genes but at the cost of many of them not being true biological 1:1 orthologs.

For more details, please see [here](#).

**Details**

References [homologeneData](#) or [gconvert](#).

**Value**

Table with all gene symbols from the given species.

**Examples**

```
genome_mouse <- all_genes(species = "mouse")
genome_human <- all_genes(species = "human")
```

---

convert_orthologs	<i>Map genes from one species to another</i>
-------------------	--

---

## Description

Currently supports ortholog mapping between any pair of 700+ species.  
 Use [map\\_species](#) to return a full list of available organisms.

## Usage

```
convert_orthologs(
  gene_df,
  gene_input = "rownames",
  gene_output = "rownames",
  standardise_genes = FALSE,
  input_species,
  output_species = "human",
  method = c("gprofiler", "homologene", "babelgene"),
  drop_nonorths = TRUE,
  non121_strategy = "drop_both_species",
  agg_fun = NULL,
  mthreshold = Inf,
  as_sparse = FALSE,
  as_DelayedArray = FALSE,
  sort_rows = FALSE,
  verbose = TRUE,
  ...
)
```

## Arguments

gene_df	<p>Data object containing the genes (see gene_input for options on how the genes can be stored within the object).          Can be one of the following formats:</p> <ul style="list-style-type: none"> <li>• <code>matrix</code> : A sparse or dense matrix.</li> <li>• <code>data.frame</code> : A <code>data.frame</code>, <code>data.table</code>. or <code>tibble</code>.</li> <li>• <code>codelist</code> : A list or character vector.</li> </ul> <p>Genes, transcripts, proteins, SNPs, or genomic ranges can be provided in any format (HGNC, Ensembl, RefSeq, UniProt, etc.) and will be automatically converted to gene symbols unless specified otherwise with the <code>...</code> arguments.  <i>Note:</i> If you set <code>method="homologene"</code>, you must either supply genes in gene symbol format (e.g. "Sox2") OR set <code>standardise_genes=TRUE</code>.</p>
gene_input	<p>Which aspect of gene_df to get gene names from:</p> <ul style="list-style-type: none"> <li>• <code>"rownames"</code> : From row names of <code>data.frame/matrix</code>.</li> </ul>

	<ul style="list-style-type: none"> <li>• "colnames" : From column names of data.frame/matrix.</li> <li>• &lt;column name&gt; : From a column in gene_df, e.g. "gene_names".</li> </ul>
gene_output	<p>How to return genes. Options include:</p> <ul style="list-style-type: none"> <li>• "rownames" : As row names of gene_df.</li> <li>• "colnames" : As column names of gene_df.</li> <li>• "columns" : As new columns "input_gene", "ortholog_gene" (and "input_gene_standard" if standardise_genes=TRUE) in gene_df.</li> <li>• "dict" : As a dictionary (named list) where the names are input_gene and the values are ortholog_gene.</li> <li>• "dict_rev" : As a reversed dictionary (named list) where the names are ortholog_gene and the values are input_gene.</li> </ul>
standardise_genes	<p>If TRUE AND gene_output="columns", a new column "input_gene_standard" will be added to gene_df containing standardised HGNC symbols identified by <a href="#">gorth</a>.</p>
input_species	<p>Name of the input species (e.g., "mouse", "fly"). Use <a href="#">map_species</a> to return a full list of available species.</p>
output_species	<p>Name of the output species (e.g. "human", "chicken"). Use <a href="#">map_species</a> to return a full list of available species.</p>
method	<p>R package to use for gene mapping:</p> <ul style="list-style-type: none"> <li>• "gprofiler" : Slower but more species and genes.</li> <li>• "homologene" : Faster but fewer species and genes.</li> <li>• "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.</li> </ul>
drop_nonorths	<p>Drop genes that don't have an ortholog in the output_species.</p>
non121_strategy	<p>How to handle genes that don't have 1:1 mappings between input_species:output_species. Options include:</p> <ul style="list-style-type: none"> <li>• "drop_both_species" or "dbs" or 1 : Drop genes that have duplicate mappings in either the input_species or output_species (DEFAULT).</li> <li>• "drop_input_species" or "dis" or 2 : Only drop genes that have duplicate mappings in the input_species.</li> <li>• "drop_output_species" or "dos" or 3 : Only drop genes that have duplicate mappings in the output_species.</li> <li>• "keep_both_species" or "kbs" or 4 : Keep all genes regardless of whether they have duplicate mappings in either species.</li> </ul>



	<ul style="list-style-type: none"> <li>• "keep_popular" or "kp" or 5 : Return only the most "popular" interspecies ortholog mappings. This procedure tends to yield a greater number of returned genes but at the cost of many of them not being true biological 1:1 orthologs.</li> <li>• "sum", "mean", "median", "min" or "max" : When gene_df is a matrix and gene_output="rownames", these options will aggregate many-to-one gene mappings (input_species-to-output_species) after dropping any duplicate genes in the output_species.</li> </ul>
agg_fun	Aggregation function passed to <a href="#">aggregate_mapped_genes</a> . Set to NULL to skip aggregation step (default).
mtreehold	Maximum number of ortholog names per gene to show. Passed to <a href="#">gorth</a> . Only used when method="gprofiler" ( <i>DEFAULT</i> : Inf).
as_sparse	Convert gene_df to a sparse matrix. Only works if gene_df is one of the following classes: <ul style="list-style-type: none"> <li>• matrix</li> <li>• Matrix</li> <li>• data.frame</li> <li>• data.table</li> <li>• tibble</li> </ul> If gene_df is a sparse matrix to begin with, it will be returned as a sparse matrix (so long as gene_output= "rownames" or "colnames").
as_DelayedArray	Convert aggregated matrix to <a href="#">DelayedArray</a> .
sort_rows	Sort gene_df rows alphanumerically.
verbose	Print messages.
...	Additional arguments to be passed to <a href="#">gorth</a> or <a href="#">homologene</a> .

*NOTE:* To return only the most "popular" interspecies ortholog mappings, supply mtreehold=1 here AND set method="gprofiler" above. This procedure tends to yield a greater number of returned genes but at the cost of many of them not being true biological 1:1 orthologs.

For more details, please see [here](#).

## Value

gene\_df with orthologs converted to the output\_species.  
Instead returned as a dictionary (named list) if gene\_output="dict" or "dict\_rev".

## Examples

```
data("exp_mouse")
gene_df <- convert_orthologs(
  gene_df = exp_mouse,
  input_species = "mouse"
)
```

---

create_background	<i>Create gene background</i>
-------------------	-------------------------------

---

## Description

Create a gene background as the union/intersect of all orthologs between input species (species1 and species2), and the output\_species. This can be useful when generating random lists of background genes to test against in analyses with data from multiple species (e.g. enrichment of mouse cell-type markers gene sets in human GWAS-derived gene sets).

## Usage

```
create_background(
  species1,
  species2,
  output_species = "human",
  as_output_species = TRUE,
  use_intersect = TRUE,
  bg = NULL,
  gene_map = NULL,
  method = "homologene",
  non121_strategy = "drop_both_species",
  verbose = TRUE
)
```

## Arguments

species1	First species.
species2	Second species.
output_species	Species to convert all genes from species1 and species2 to first. Default="human", but can be to either any species supported by <b>orthogene</b> , including species1 or species2.
as_output_species	Return background gene list as output_species orthologs, instead of the gene names of the original input species.
use_intersect	When species1 and species2 are both different from output_species, this argument will determine whether to use the intersect (TRUE) or union (FALSE) of all genes from species1 and species2.
bg	User supplied background list that will be returned to the user after removing duplicate genes.
gene_map	User-supplied gene_map data table from <a href="#">map_orthologs</a> or <a href="#">map_genes</a> .
method	R package to use for gene mapping: <ul style="list-style-type: none"> <li>• "gprofiler" : Slower but more species and genes.</li> <li>• "homologene" : Faster but fewer species and genes.</li> <li>• "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.</li> </ul>

non121\_strategy

How to handle genes that don't have 1:1 mappings between input\_species:output\_species. Options include:

- "drop\_both\_species" or "dbs" or 1 :  
Drop genes that have duplicate mappings in either the input\_species or output\_species  
(*DEFAULT*).
- "drop\_input\_species" or "dis" or 2 :  
Only drop genes that have duplicate mappings in the input\_species.
- "drop\_output\_species" or "dos" or 3 :  
Only drop genes that have duplicate mappings in the output\_species.
- "keep\_both\_species" or "kbs" or 4 :  
Keep all genes regardless of whether they have duplicate mappings in either species.
- "keep\_popular" or "kp" or 5 :  
Return only the most "popular" interspecies ortholog mappings. This procedure tends to yield a greater number of returned genes but at the cost of many of them not being true biological 1:1 orthologs.
- "sum", "mean", "median", "min" or "max" :  
When gene\_df is a matrix and gene\_output="rownames", these options will aggregate many-to-one gene mappings (input\_species-to-output\_species) after dropping any duplicate genes in the output\_species.

verbose

Print messages.

## Value

Background gene list.

## Examples

```
bg <- orthogene::create_background(species1 = "mouse",
                                  species2 = "rat",
                                  output_species = "human")
```

---

exp\_mouse

*Gene expression data: mouse*

---

## Description

Mean pseudobulk single-cell RNA-seq gene expression matrix.

Data originally comes from Zeisel et al., 2018 (Cell).

## Usage

```
data("exp_mouse")
```

## Format

sparse matrix

**Source**

```
Publication ctd <- ewceData::ctd() exp_mouse <- as(ctd[[1]]$mean_exp, "sparseMatrix")
usethis::use_data(exp_mouse, overwrite = TRUE)
```

---

exp_mouse_enst	<i>Transcript expression data: mouse</i>
----------------	--

---

**Description**

Mean pseudobulk single-cell RNA-seq Transcript expression matrix.  
Data originally comes from Zeisel et al., 2018 (Cell).

**Usage**

```
data("exp_mouse_enst")
```

**Format**

sparse matrix

**Source**

```
Publication data("exp_mouse") mapped_genes <- map_genes(genes = rownames(exp_mouse)[seq(1,100)],
target = "ENST", species = "mouse", drop_na = FALSE) exp_mouse_enst <- exp_mouse[mapped_genes$input,]
rownames(exp_mouse_enst) <- mapped_genes$target all_nas <- orthogene::find_all_nas(rownames(exp_m
exp_mouse_enst <- exp_mouse_enst[!all_nas,] exp_mouse_enst <- phenomix::add_noise(exp_mouse_enst)
usethis::use_data(exp_mouse_enst, overwrite = TRUE)
```

---

gprofiler_namespace	<i><a href="#">gconvert</a> namespaces</i>
---------------------	--

---

**Description**

Available namespaces used by link[gprofiler2]gconvert.

**Format**

data.frame

**Source**

```
gProfiler site
#### Manually-prepared CSV #### path <- "inst/extdata/gprofiler_namespace.csv.gz" gprofiler_namespa
<- data.table::fread(path)
```

---

gprofiler_orgs	<i>Reference organisms</i>
----------------	----------------------------

---

**Description**

Organism for which gene references are available via **gProfiler API**. Used as a backup if API is not available.

**Format**

data.frame

**Source**

**gProfiler site**

```
# NOTE!: Must run usethis::use_data for all internal data at once. # otherwise, the prior
internal data will be overwritten. ##### Internal data 1: gprofiler_namespace #####
Manually-prepared CSV ##### path <- "inst/extdata/gprofiler_namespace.csv.gz" gprofiler_namespace
<- data.table::fread(path) ##### Internal data 2: gprofiler_orgs gprofiler_orgs <- orthogene::get_or
##### Save ##### usethis::use_data(gprofiler_orgs,gprofiler_namespace, overwrite = TRUE,
internal=TRUE)
```

---

infer_species	<i>Infer species from gene names</i>
---------------	--------------------------------------

---

**Description**

Infers which species the genes within gene\_df is from. Iteratively test the percentage of gene\_df genes that match with the genes from each test\_species.

**Usage**

```
infer_species(
  gene_df,
  gene_input = "rownames",
  test_species = c("human", "monkey", "rat", "mouse", "zebrafish", "fly"),
  method = c("homologene", "gprofiler", "babelgene"),
  make_plot = TRUE,
  show_plot = TRUE,
  verbose = TRUE
)
```

**Arguments**

gene_df	Data object containing the genes (see gene_input for options on how the genes can be stored within the object). Can be one of the following formats:
---------	---

- `matrix` :  
A sparse or dense matrix.
- `data.frame` :  
A `data.frame`, `data.table`. or `tibble`.
- `codelist` :  
A list or character vector.

Genes, transcripts, proteins, SNPs, or genomic ranges can be provided in any format (HGNC, Ensembl, RefSeq, UniProt, etc.) and will be automatically converted to gene symbols unless specified otherwise with the `...` arguments.

*Note:* If you set `method="homologene"`, you must either supply genes in gene symbol format (e.g. "Sox2") OR set `standardise_genes=TRUE`.

<code>gene_input</code>	Which aspect of <code>gene_df</code> to get gene names from: <ul style="list-style-type: none"> <li>• <code>"rownames"</code> : From row names of <code>data.frame/matrix</code>.</li> <li>• <code>"colnames"</code> : From column names of <code>data.frame/matrix</code>.</li> <li>• <code>&lt;column name&gt;</code> : From a column in <code>gene_df</code>, e.g. <code>"gene_names"</code>.</li> </ul>
<code>test_species</code>	Which species to test for matches with. If set to <code>NULL</code> , will default to a list of humans and 5 common model organisms. If <code>test_species</code> is set to one of the following options, it will automatically pull all species from that respective package and test against each of them: <ul style="list-style-type: none"> <li>• <code>"homologene"</code> : 20+ species (default)</li> <li>• <code>"gprofiler"</code> : 700+ species</li> <li>• <code>"babelgene"</code> : 19 species</li> </ul>
<code>method</code>	R package to use for gene mapping: <ul style="list-style-type: none"> <li>• <code>"gprofiler"</code> : Slower but more species and genes.</li> <li>• <code>"homologene"</code> : Faster but fewer species and genes.</li> <li>• <code>"babelgene"</code> : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.</li> </ul>
<code>make_plot</code>	Make a plot of the results.
<code>show_plot</code>	Print the plot of the results.
<code>verbose</code>	Print messages.

## Value

An ordered dataframe of `test_species` from best to worst matches.

## Examples

```
data("exp_mouse")
matches <- orthogene::infer_species(gene_df = exp_mouse[1:200,])
```

---

map_genes	<i>Map genes</i>
-----------	------------------

---

## Description

Input a list of genes, transcripts, proteins, SNPs, or genomic ranges in any format (HGNC, Ensembl, RefSeq, UniProt, etc.) and return a table with standardised gene symbols (the "names" column).

## Usage

```
map_genes(
  genes,
  species = "hsapiens",
  target = "ENSG",
  mthreshold = Inf,
  drop_na = FALSE,
  numeric_ns = "",
  run_map_species = TRUE,
  verbose = TRUE
)
```

## Arguments

genes	Gene list.
species	Species to map against.
target	target namespace.
mthreshold	maximum number of results per initial alias to show. Shows all by default.
drop_na	Drop all genes without mappings. Sets <code>gprofiler2::gconvert(filter_na=)</code> as well an additional round of more comprehensive NA filtering by <b>orthogene</b> .
numeric_ns	namespace to use for fully numeric IDs ( <a href="#">list of available namespaces</a> ).
run_map_species	Standardise species names with <a href="#">map_species</a> first (Default: TRUE).
verbose	Print messages.

## Details

Uses [gconvert](#). The exact contents of the output table will depend on target parameter. See `?gprofiler2::gconvert` for more details.

## Value

Table with standardised genes.

## Examples

```
genes <- c(
  "K1f4", "Sox2", "TSPAN12", "NM_173007", "Q8BKT6",
  "ENSMUSG00000012396", "ENSMUSG00000074637"
)
mapped_genes <- map_genes(
```

```

    genes = genes,
    species = "mouse"
)

```

---

map\_orthologs

*Map orthologs*


---

## Description

Map orthologs from one species to another.

## Usage

```

map_orthologs(
  genes,
  standardise_genes = FALSE,
  input_species,
  output_species = "human",
  method = c("gprofiler", "homologene", "babelgene"),
  mthreshold = Inf,
  verbose = TRUE,
  ...
)

```

## Arguments

genes	can be a mixture of any format (HGNC, Ensembl, RefSeq, UniProt, etc.) and will be automatically converted to standardised HGNC symbol format.
standardise_genes	If TRUE AND gene_output="columns", a new column "input_gene_standard" will be added to gene_df containing standardised HGNC symbols identified by <a href="#">gorth</a> .
input_species	Name of the input species (e.g., "mouse", "fly"). Use <a href="#">map_species</a> to return a full list of available species.
output_species	Name of the output species (e.g. "human", "chicken"). Use <a href="#">map_species</a> to return a full list of available species.
method	R package to use for gene mapping: <ul style="list-style-type: none"> <li>• "gprofiler" : Slower but more species and genes.</li> <li>• "homologene" : Faster but fewer species and genes.</li> <li>• "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.</li> </ul>
mthreshold	Maximum number of ortholog names per gene to show. Passed to <a href="#">gorth</a> . Only used when method="gprofiler" ( <i>DEFAULT</i> : Inf).
verbose	Print messages.
...	Additional arguments to be passed to <a href="#">gorth</a> or <a href="#">homologene</a> .

*NOTE:* To return only the most "popular" interspecies ortholog mappings, supply mthreshold=1 here AND set method="gprofiler" above. This procedure tends to yield a greater number of returned genes but at the cost of many of them



not being true biological 1:1 orthologs.

For more details, please see [here](#).

## Details

map\_orthologs() is a core function within convert\_orthologs(), but does not have many of the extra checks, such as non121\_strategy) and drop\_nonorths.

## Value

Ortholog map data.frame with at least the columns "input\_gene" and "ortholog\_gene".

## Examples

```
data("exp_mouse")
gene_map <- map_orthologs(
  genes = rownames(exp_mouse),
  input_species = "mouse"
)
```

---

map_species	<i>Standardise species names</i>
-------------	----------------------------------

---

## Description

Search gprofiler database for species that match the input text string. Then translate to a standardised species ID.

## Usage

```
map_species(
  species = NULL,
  search_cols = c("display_name", "id", "scientific_name", "taxonomy_id"),
  output_format = c("scientific_name", "id", "display_name", "taxonomy_id", "version"),
  method = c("homologene", "gprofiler", "babelgene"),
  use_local = TRUE,
  verbose = TRUE
)
```

## Arguments

species	Species query (e.g. "human", "homo sapiens", "hapiens", or 9606). If given a list, will iterate queries for each item. Set to NULL to return all species.
search_cols	Which columns to search for species substring in metadata <a href="#">API</a> .
output_format	Which column to return.
method	R package to use for gene mapping: <ul style="list-style-type: none"> <li>"gprofiler" : Slower but more species and genes.</li> <li>"homologene" : Faster but fewer species and genes.</li> <li>"babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.</li> </ul>

use_local	If TRUE <i>default</i> , <code>map_species</code> uses a locally stored version of the species meta-data table instead of pulling directly from the gprofiler API. Local version may not be fully up to date, but should suffice for most use cases.
verbose	Print messages.

## Value

Species ID of type `output_format`

## Examples

```
ids <- map_species(species = c(
  "human", 9606, "mus musculus",
  "fly", "C elegans"
))
```

---

plot_orthotree	<i>Create a phylogenetic tree of shared orthologs</i>
----------------	---

---

## Description

Automatically creates a phylogenetic tree plot annotated with metadata describing how many orthologous genes each species shares with the `reference_species` ("human" by default).

## Usage

```
plot_orthotree(
  tree = NULL,
  orth_report = NULL,
  species = NULL,
  method = c("homologene", "gprofiler", "babelgene"),
  tree_source = "timetree",
  non121_strategy = "drop_both_species",
  reference_species = "human",
  clades = list(Primates = c("Homo sapiens", "Macaca mulatta"), Eutherians =
    c("Homo sapiens", "Mus musculus", "Bos taurus"), Mammals = c("Homo sapiens",
    "Mus musculus", "Bos taurus", "Ornithorhynchus anatinus", "Monodelphis domestica"),
  Tetrapods = c("Homo sapiens", "Mus musculus", "Gallus gallus", "Anolis carolinensis",
    "Xenopus tropicalis"), Vertebrates = c("Homo sapiens", "Mus musculus",
    "Gallus gallus", "Anolis carolinensis", "Xenopus tropicalis", "Danio rerio")),
  scaling_factor = 1,
  show_plot = TRUE,
  save_paths = c(tempfile(fileext = ".ggtree.pdf"), tempfile(fileext = ".ggtree.png")),
  width = 10,
  height = 10,
  mc.cores = 1,
  verbose = TRUE
)
```

**Arguments**

tree	A phylogenetic tree of class <a href="#">phylo</a> . If no tree is provided (NULL) a 100-way multiz tree will be imported from <a href="#">UCSC Genome Browser</a> .
orth_report	An ortholog report from one or more species generated by <a href="#">report_orthologs</a> .
species	Species to include in the final plot. If NULL, then all species from the given database (method) will be included (via <a href="#">map_species</a> ), so long as they also exist in the tree.
method	R package to use for gene mapping: <ul style="list-style-type: none"> <li>• "gprofiler" : Slower but more species and genes.</li> <li>• "homologene" : Faster but fewer species and genes.</li> <li>• "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.</li> </ul>
tree_source	Can be one of the following: <ul style="list-style-type: none"> <li>• "timetree2022": Import and prune the <a href="#">TimeTree &gt;147k species</a> phylogenetic tree. Can also simply type "timetree".</li> <li>• "timetree2015": Import and prune the <a href="#">TimeTree &gt;50k species</a> phylogenetic tree.</li> <li>• "OmaDB": Construct a tree from <a href="#">OMA</a> (Orthologous Matrix browser) via the <a href="#">getTaxonomy</a> function. <i>NOTE:</i> Does not contain branch lengths, and therefore may have limited utility.</li> <li>• "UCSC": Import and prune the <a href="#">UCSC 100-way alignment</a> phylogenetic tree (hg38 version).</li> <li>• "&lt;path&gt;": Read a tree from a newick text file from a local or remote URL using <a href="#">read.tree</a>.</li> </ul>
non121_strategy	How to handle genes that don't have 1:1 mappings between input_species:output_species. Options include: <ul style="list-style-type: none"> <li>• "drop_both_species" or "dbs" or 1 : Drop genes that have duplicate mappings in either the input_species or output_species (<i>DEFAULT</i>).</li> <li>• "drop_input_species" or "dis" or 2 : Only drop genes that have duplicate mappings in the input_species.</li> <li>• "drop_output_species" or "dos" or 3 : Only drop genes that have duplicate mappings in the output_species.</li> <li>• "keep_both_species" or "kbs" or 4 : Keep all genes regardless of whether they have duplicate mappings in either species.</li> <li>• "keep_popular" or "kp" or 5 : Return only the most "popular" interspecies ortholog mappings. This procedure tends to yield a greater number of returned genes but at the cost of many of them not being true biological 1:1 orthologs.</li> </ul>



---

prepare_tree	<i>Prepare a phylogenetic tree</i>
--------------	------------------------------------

---

## Description

Import a phylogenetic tree and then conduct a series of optional standardisation steps. Optionally, if `output_format` is not `NULL`, species names from both the tree and the `species` argument will first be standardised using [map\\_species](#).

## Usage

```
prepare_tree(
  tree_source = "timetree",
  species = NULL,
  output_format = "scientific_name",
  run_map_species = c(TRUE, TRUE),
  method = c("homologene", "gprofiler", "babelgene"),
  force_ultrametric = TRUE,
  age_max = NULL,
  show_plot = TRUE,
  verbose = TRUE,
  ...
)
```

## Arguments

<code>tree_source</code>	Can be one of the following: <ul style="list-style-type: none"> <li>"timetree2022": Import and prune the <b>TimeTree &gt;147k species</b> phylogenetic tree. Can also simply type "timetree".</li> <li>"timetree2015": Import and prune the <b>TimeTree &gt;50k species</b> phylogenetic tree.</li> <li>"OmaDB": Construct a tree from <b>OMA</b> (Orthologous Matrix browser) via the <a href="#">getTaxonomy</a> function. <i>NOTE:</i> Does not contain branch lengths, and therefore may have limited utility.</li> <li>"UCSC": Import and prune the <b>UCSC 100-way alignment</b> phylogenetic tree (hg38 version).</li> <li>"&lt;path&gt;": Read a tree from a newick text file from a local or remote URL using <a href="#">read.tree</a>.</li> </ul>
<code>species</code>	Species names to subset the tree by (after <code>standardise_species</code> step).
<code>output_format</code>	Which column to return.
<code>run_map_species</code>	Whether to first standardise species names with <a href="#">map_species</a> .
<code>method</code>	R package to use for gene mapping: <ul style="list-style-type: none"> <li>"gprofiler": Slower but more species and genes.</li> </ul>

- "homologene" : Faster but fewer species and genes.
- "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.

force_ultrametric	Whether to force the tree to be ultrametric (i.e. make all tips the same date) using <a href="#">force.ultrametric</a> .
age_max	Rescale the edges of the tree into units of millions of years (MY) instead than evolutionary rates (e.g. dN/dS ratios). Only used if age_max, the max number , is numeric. Times are computed using <a href="#">makeChronosCalib</a> and <a href="#">chronos</a> .
show_plot	Show a basic plot of the resulting tree.
verbose	Print messages.
...	Additional arguments passed to <a href="#">makeChronosCalib</a> .

Value

A filtered tree of class "phylo" (with standardised species names).

Source

TimeTree 5: An Expanded Resource for Species Divergence Times

Examples

```
species <- c("human","chimp","mouse")
tr <- orthogene::prepare_tree(species = species)
```

---

report_orthologs	<i>Report orthologs</i>
------------------	-------------------------

---

Description

Identify the number of orthologous genes between two species.

Usage

```
report_orthologs(
  target_species = "mouse",
  reference_species = "human",
  standardise_genes = FALSE,
  method_all_genes = c("homologene", "gprofiler", "babelgene"),
  method_convert_orthologs = method_all_genes,
  drop_nonorths = TRUE,
  non121_strategy = "drop_both_species",
  round_digits = 2,
  return_report = TRUE,
  mc.cores = 1,
  verbose = TRUE,
  ...
)
```

## Arguments

- target\_species** Target species.
- reference\_species**  
Reference species.
- standardise\_genes**  
If TRUE AND `gene_output="columns"`, a new column "input\_gene\_standard" will be added to `gene_df` containing standardised HGNC symbols identified by [gorth](#).
- method\_all\_genes**  
R package to to use in [all\\_genes](#) step:
- "gprofiler" : Slower but more species and genes.
  - "homologene" : Faster but fewer species and genes.
  - "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.
- method\_convert\_orthologs**  
R package to to use in [convert\\_orthologs](#) step:
- "gprofiler" : Slower but more species and genes.
  - "homologene" : Faster but fewer species and genes.
  - "babelgene" : Faster but fewer species and genes. Also gives consensus scores for each gene mapping based on a several different data sources.
- drop\_nonorths** Drop genes that don't have an ortholog in the output\_species.
- non121\_strategy**  
How to handle genes that don't have 1:1 mappings between input\_species:output\_species. Options include:
- "drop\_both\_species" or "dbs" or 1 :  
Drop genes that have duplicate mappings in either the input\_species or output\_species  
(*DEFAULT*).
  - "drop\_input\_species" or "dis" or 2 :  
Only drop genes that have duplicate mappings in the input\_species.
  - "drop\_output\_species" or "dos" or 3 :  
Only drop genes that have duplicate mappings in the output\_species.
  - "keep\_both\_species" or "kbs" or 4 :  
Keep all genes regardless of whether they have duplicate mappings in either species.
  - "keep\_popular" or "kp" or 5 :  
Return only the most "popular" interspecies ortholog mappings. This procedure tends to yield a greater number of returned genes but at the cost of many of them not being true biological 1:1 orthologs.
  - "sum", "mean", "median", "min" or "max" :  
When `gene_df` is a matrix and `gene_output="rownames"`, these options will aggregate many-to-one gene mappings (input\_species-to-output\_species) after dropping any duplicate genes in the output\_species.
- round\_digits** Number of digits to round to when printing percentages.
- return\_report** Return just the ortholog mapping between two species (FALSE) or return both the ortholog mapping as well a data.frame of the report statistics (TRUE).

mc.cores	Number of cores to parallelise each target_species with.
verbose	Print messages.
...	Additional arguments to be passed to <a href="#">gorth</a> or <a href="#">homologene</a> .

*NOTE:* To return only the most "popular" interspecies ortholog mappings, supply `mthreshold=1` here AND set `method="gprofiler"` above. This procedure tends to yield a greater number of returned genes but at the cost of many of them not being true biological 1:1 orthologs.

For more details, please see [here](#).

### Value

A list containing:

- `map` : A table of inter-species gene mappings.
- `report` : A list of aggregate orthology report statistics.

If >1 target\_species are provided, then a table of aggregated report statistics concatenated across species will be returned instead.

### Examples

```
orth_fly <- orthogene::report_orthologs(
  target_species = "fly",
  reference_species = "human"
)
```



# Index

## \* datasets

- exp\_mouse, [11](#)
- exp\_mouse\_enst, [12](#)

aggregate\_mapped\_genes, [3](#), [9](#)

aggregate\_rows, [5](#)

all\_genes, [5](#), [23](#)

chronos, [22](#)

convert\_orthologs, [7](#), [23](#)

create\_background, [10](#)

data.frame, [3](#), [4](#)

DelayedArray, [5](#), [9](#)

exp\_mouse, [11](#)

exp\_mouse\_enst, [12](#)

floor, [5](#)

force.ultrametric, [22](#)

gconvert, [6](#), [12](#), [15](#)

getTaxonomy, [19](#), [21](#)

gorth, [6](#), [8](#), [9](#), [16](#), [23](#), [24](#)

gprofiler\_namespace, [12](#)

gprofiler\_orgs, [13](#)

homologene, [6](#), [9](#), [16](#), [24](#)

homologeneData, [6](#)

infer\_species, [13](#)

makeChronosCalib, [22](#)

many2many\_rows, [3](#), [5](#)

map\_genes, [4](#), [10](#), [15](#)

map\_orthologs, [4](#), [10](#), [16](#)

map\_species, [4](#), [6–8](#), [15](#), [16](#), [17](#), [18](#), [19](#), [21](#)

MRCA, [20](#)

orthogene (orthogene-package), [3](#)

orthogene-package, [3](#)

phylo, [19](#)

plot\_orthotree, [18](#)

prepare\_tree, [21](#)

read.tree, [19](#), [21](#)

report\_orthologs, [19](#), [22](#)