

# Package ‘orthGS’

September 17, 2024

**Title** Orthology vs Paralogy Relationships among Glutamine Synthetase  
from Plants

**Version** 0.1.5

**Description** Tools to analyze and infer orthology and paralogy relationships between glutamine synthetase proteins in seed plants.

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Depends** R (>= 2.10)

**LazyData** true

**Imports** ape, bio3d, castor, igraph, phangorn, phytools, seqinr,  
TreeTools

**Suggests** BiocManager, Biostrings, fs, knitr, muscle, rmarkdown,  
testthat

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Elena Aledo [aut, cre, cph],  
Juan-Carlos Aledo [aut] (<<https://orcid.org/0000-0002-3497-9945>>)

**Maintainer** Elena Aledo <[elenaaledoesteban@gmail.com](mailto:elenaaledoesteban@gmail.com)>

**Repository** CRAN

**Date/Publication** 2024-09-17 12:10:02 UTC

## Contents

agf . . . . .	2
AngGym . . . . .	3
A_selected . . . . .	4
coltips . . . . .	4
gapless_msa . . . . .	5
getseqGS . . . . .	6
madRoot . . . . .	6

mapTrees . . . . .	7
mltree . . . . .	8
msa . . . . .	9
orthG . . . . .	10
orthP . . . . .	10
sdf . . . . .	11
selected_tr . . . . .	12
speciesGS . . . . .	12
subsetGS . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

agf	<i>Angiosperms Gymnosperms Ferns</i>
-----	--------------------------------------

---

## Description

Angiosperms Gymnosperms Ferns

## Usage

agf

## Format

A dataframe with 275 rows (GS proteins) and 23 columns:

**n** Reference number

**phylo\_id** Unique identification label of the protein/gen

**species** Species

**taxon** Acrogymnospermae, Angiospermae, Polypodiopsida

**dna** CDS sequence

**prot** Protein sequence

**short** Unique three letter identification of the species

**gs** GS2, GS1a or GS1b\_Ang, GS1b\_Gym

**pI** isoelectric point

**factor** Ferns, GS2, GS1a, GS1b\_Ang, GS1b\_Gym

**size** number of residues

**CSpos** position signal

**prediction** prediction

**Lk\_SP** seq pep

**Lk\_mTP** mit

**Lk\_cTP** chl

**Lk\_Thylak** thy  
**secAa** amino acid at position 2  
**core** core  
**dabase** db  
**acc** acc  
**up\_id** uniprot  
**note** note

### Source

It has been manually curated by the authors

---

AngGym	<i>Angiosperms Gymnosperms</i>
--------	--------------------------------

---

### Description

Angiosperms Gymnosperms

### Usage

AngGym

### Format

A dataframe with 155 rows (GS proteins) and 10 columns:

**n** Reference number  
**phylo\_id** Unique identification label of the protein/gen  
**species** Species  
**taxon** Acrogymnospermae or Angiospermae  
**class** Angiosperms: Amborellopsida, Liliopsida, Magnoliopsida; Gymnosperms: Ginkgoopsida, Cycadopsida, Gnetopsida, Pinopsida  
**dna** CDS sequence  
**prot** Protein sequence  
**short** Unique three letter identification of the species  
**gsLineage** Either GS2, GS1a or GS1b  
**plant\_group** Primitive angiosperms, Modern angiosperms, Ginkgo-Cycadales, Gnetales, Pinacea, Conifer II

### Source

It has been manually curated by the authors

---

A\_selected                      *Adjacency Matrix for Orthology Graph*

---

**Description**

155 x 155 square matrix (155 GS proteins from 45 seed plant species)

**Usage**

A\_selected

**Format**

A matrix with 155 rows and 155 columns

**Source**

It has been generated using the function `orthG::mapTrees()` and the reconciliation output file 'selected'. Verbigracia: `orthG::mapTrees('./inst/extdata/selected')` The reconciliation was carried out using RANGER-DTL with parameters  $D = 1$ ,  $T = 10$  and  $L = 1$ .

---

coltips                              *Colouring Tree Tips*

---

**Description**

Make a color vector for colouring tree tips

**Usage**

`coltips(phy)`

**Arguments**

phy                      tree as a phylo object

**Details**

Each tip is given a color according to the nature of the isoform: green (GS2), blue (GS1a), brown (GS1b Gym), salmon (GS1b Ang), purple (other).

**Value**

a color vector as long as the number of tips

**Examples**

```
coltips(ape::read.tree(text = "((Bdi, Sly), (Pp, Ap));"))
```

---

gapless_msa	<i>Remove Gaps in a MSA</i>
-------------	-----------------------------

---

### Description

Removes gaps in a given msa.

### Usage

```
gapless_msa(msa, seqtype = 'AA', df = TRUE, sfile = FALSE)
```

### Arguments

msa	input alignment.
seqtype	the nature of the sequences: 'DNA' or 'AA'.
df	logical. When TRUE msa should be a matrix, when FALSE msa should be a string giving the path to a fasta file containing the alignment.
sfile	if different to FALSE, then it should be a string indicating the path to save a fasta alignment file.

### Details

It should be noted that this function does not carry out the alignment itself.

### Value

an alignment without gaps in form of matrix or a file containing such an alignment in fasta format.

### See Also

msa

### Examples

```
gapless_msa(msa(sequences = c("APGW", "AGWC", "CWGA"),ids = c("a", "b", "c"))$ali)
```

---

getseqGS	<i>Get the GS Sequence</i>
----------	----------------------------

---

**Description**

Provides the requested GS sequence

**Usage**

```
getseqGS(phylo_id, molecule = "Prot")
```

**Arguments**

phylo_id	the unique sequence identifier
molecule	either "Prot" or "CDS"

**Details**

The identifier should be one of the 'phylo\_id' from data(agf).

**Value**

The requested sequence as a character string.

**Examples**

```
getseqGS("Pp_GS1b_2")
```

---

madRoot	<i>Find The Root of a Phylogenetic Tree Using MAD Method</i>
---------	--

---

**Description**

Finds the root of an unrooted phylogenetic tree by minimizing the relative deviation from the molecular clock.

**Usage**

```
madRoot(tree, output_mode = 'phylo')
```

**Arguments**

tree	unrooted tree string in newick format or a tree object of class 'phylo'.
output_mode	amount of information to return. If 'phylo' (default) only the rooted tree is returned. If 'stats' also a structure with the ambiguity index, clock cv, the minimum ancestor deviation and the number of roots. If 'full' also an unrooted tree object, the index of the root branch, the branch ancestor deviations and a rooted tree object.

**Details**

This function is a slight modification of the code provided by Tria et al at <https://www.mikrobio.uni-kiel.de/de/ag-dagan/ressourcen>.

**Value**

a rooted tree and supplementary information if required.

**Author(s)**

Tria, F. D. K., Landan, G. and Dagan, T.

**References**

Tria, F. D. K., Landan, G. and Dagan, T. Nat. Ecol. Evol. 1, 0193 (2017).

**Examples**

```
a <- msa(sequences=c("RAPGT", "KMPGT", "ESGGT"), ids = letters[1:3])$ali
rownames(a) <- letters[1:3]
tr <- mltree(a)$tree
rtr <- madRoot(tr)
```

---

mapTrees

*Map Gene Tree into Species Tree*

---

**Description**

Maps a gene/protein tree into a species tree

**Usage**

```
mapTrees(path2rec)
```

**Arguments**

path2rec            path to the file containing the reconciliation output.

**Details**

Mapping gene tree into species tree allow to infer the sequence of events (Duplication, Speciation, Transfer).

**Value**

A list with three elements. The first one is a 'phylo' object where the nodelabels indicate the event: D, duplication or T transfer. If no label is shown is because the event correspond to speciation. The second element is a dataframe (the first column is the label of the internal nodes in the gene tree; the second column is the label of the internal nodes in the species tree, and the third and fourth columns label each internal node according to the inferred event). The third element of the list is an adjacency matrix: 1 when two proteins are orthologous, 0 if they are paralogous.

**Examples**

```
mapTrees(fs::path_package("extdata", "representatives", package = "orthGS"))
```

---

mltree

*Build Up a ML Tree*


---

**Description**

Given an alignment builds an ML tree.

**Usage**

```
mltree(msa, df = TRUE, gap1 = TRUE, model = "WAG")
```

**Arguments**

msa	input alignment.
df	logical. When TRUE msa should be a dataframe, when FALSE msa should be a string giving the path to a fasta file containing the alignment.
gap1	logical, when TRUE a gapless alignment is used.
model	allows to choose an amino acid models (see the function phangorn::as.pml)

**Details**

The function makes a NJ tree and then improve it using an optimization procedure based on ML.

**Value**

a ML optimized tree (and parameters)

**See Also**

gapless\_msa

**Examples**

```
a <- msa(sequences=c("RAPGT", "KMPGT", "ESGGT"), ids = letters[1:3])$ali
rownames(a) <- letters[1:3]
tr <- mltree(a)$tree
```

---

**msa** *Multiple Sequence Alignment*

---

**Description**

Aligns multiple protein, DNA or CDS sequences.

**Usage**

```
msa(sequences, ids = names(sequences), seqtype = "prot", sfile = FALSE, inhouse = FALSE)
```

**Arguments**

sequences	vector containing the sequences as strings.
ids	character vector containing the sequences' ids.
seqtype	it should be either "prot" of "dna" or "cds" (see details).
sfile	if different to FALSE, then it should be a string indicating the path to save a fasta alignment file.
inhouse	logical, if TRUE the in-house MUSCLE software is used. It must be installed on your system and in the search path for executables.

**Details**

If seqtype is set to "cds" the sequences must not contain stop codons and they will be translated using the standard code. Afterward, the amino acid alignment will be used to lead the codon alignment.

**Value**

Returns a list of four elements. The first one (`$seq`) provides the sequences analyzed, the second element (`$id`) returns the identifiers, the third element (`$aln`) provides the alignment in fasta format and the fourth element (`$ali`) gives the alignment in matrix format.

**Examples**

```
msa(sequences = c("APGW", "AGWC", "CWGA"),ids = c("a", "b", "c"))
```

---

orthG *Infer GS OrthoGroups Within a Set of Species*

---

**Description**

Infers GS orthogroups using tree reconciliation

**Usage**

```
orthG(set = "all")
```

**Arguments**

set set of species of interest provided as a character vector either with the binomial or short code of the species (see data(sdf)).

**Details**

When set = "all", all the species in the database will be included.

**Value**

A list with two elements. The first one is the adjacency matrix (1 for orthologous, 0 for paralogous). The second element is an orthogroup graph.

**Examples**

```
orthG(set = c("Pp", "Psy", "Psm", "Ap"))
```

---

orthP *Search Orthologous of a Given Protein*

---

**Description**

Searchs orthologous of a given protein within a set of selected species

**Usage**

```
orthP(phylo_id, set = "all")
```

**Arguments**

phylo\_id phylo\_id of the query protein  
 set set of species of interest provided as a character vector, either with the binomial or short code of the species (see details).

**Details**

When set = "all", the search will be carry out against all the species in the database.

**Value**

A list with thee elements: 1. subtree of the relevant proteins; 2. vector color; 3. phylo\_ids of the orthologous found.

**Examples**

```
orthP(phylo_id = "Pp_GS1a", set = c("Pp", "Psy", "Psm", "Ap"))
```

---

sdf

*Seed Plants and Ferns GS*


---

**Description**

155 GS proteins from 25 seed plants species and 41 GS proteins from 11 fern species

**Usage**

sdf

**Format**

A dataframe with 196 rows (GS proteins) and 7 columns:

**n** Reference number

**Sec.Name\_** Unique identification label of the protein

**species** Species

**taxon** Acrogymnospermae, Angiospermae or Polypodiopsida

**short** Unique three letter identification of the species

**gs** Either GS2, GS1a, GS1b\_Gym or GS1b\_Ang. Here the ferns proteins have been forced to be either GS1a or GS2

**tax\_group** Taxonomic group

**Source**

It has been handly curated by the authors

---

selected_tr	<i>Ultrametric Rooted Seed Plants Tree</i>
-------------	--

---

**Description**

155 GS proteins from 45 seed plants species Rooted using MAD (Minimal Ancestor Deviation)

**Usage**

```
selected_tr
```

**Format**

An phylo object

**Source**

It has been manually curated by the authors

---

speciesGS	<i>Map Species Names</i>
-----------	--------------------------

---

**Description**

Map binomial species name to short code species name and vice versa

**Usage**

```
speciesGS(sp)
```

**Arguments**

sp                    set of species of interest (either binomial or short code name)

**Details**

The species set should be given as a character vector (see example)

**Value**

A dataframe containing the information for the requested species.

**Examples**

```
speciesGS(c("Pinus pinaster", "Ath"))
```

---

`subsetGS`*GS Proteins Report*

---

**Description**

Assembles a report regarding the GS proteins found in the indicated subset of species

**Usage**

```
subsetGS(sp)
```

**Arguments**

`sp` set of species of interest (either binomial or short code name)

**Details**

This function returns the protein and DNA sequences of the different isoforms found in each species, along with other relevant data.

**Value**

A dataframe with the information for the requested species.

**Examples**

```
subsetGS(c("Pinus pinaster", "Ath"))
```

# Index

## \* datasets

- A\_selected, 4
- agf, 2
- AngGym, 3
- sdf, 11
- selected\_tr, 12

A\_selected, 4

agf, 2

AngGym, 3

coltips, 4

gapless\_msa, 5

getseqGS, 6

madRoot, 6

mapTrees, 7

mltree, 8

msa, 9

orthG, 10

orthP, 10

sdf, 11

selected\_tr, 12

speciesGS, 12

subsetGS, 13