Package ‘geneplast’

May 17, 2024

Type Package
Title Evolutionary and plasticity analysis of orthologous groups
Version 1.30.0
Author Rodrigo Dalmolin, Mauro Castro
Maintainer Mauro Castro <mauro.a.castro@gmail.com>
Depends R (>= 4.0), methods
Imports igraph, snow, ape, grDevices, graphics, stats, utils, data.table
Suggests RTN, RUnit, BiocGenerics, BiocStyle, knitr, rmarkdown, Fletcher2013b, geneplast.data, geneplast.data.string.v91, ggplot2, ggpubr, plyr
Description Geneplast is designed for evolutionary and plasticity analysis based on orthologous groups distribution in a given species tree. It uses Shannon information theory and orthologs abundance to estimate the Evolutionary Plasticity Index. Additionally, it implements the Bridge algorithm to determine the evolutionary root of a given gene based on its orthologs distribution.
License GPL (>= 2)
biocViews Genetics, GeneRegulation, SystemsBiology
VignetteBuilder knitr
LazyLoad yes
NeedsCompilation no
git_url https://git.bioconductor.org/packages/geneplast
git_branch RELEASE_3_19
git_last_commit 6445f1f
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-17
geneplast-package

Geneplast: an R package for evolutionary rooting and plasticity inference based on orthologous groups distribution.

Description

Geneplast is designed for evolutionary and plasticity analysis based on orthologous groups distribution in a given species tree. It uses Shannon information theory and orthologs abundance to estimate the Evolutionary Plasticity Index. Additionally, it implements the Bridge algorithm to determine the evolutionary root of a given gene based on its orthologs distribution.

Details

Package: geneplast
Type: Package
Version: 1.0
Date: 2016-06-10
License: GPL (>= 2)

R package for gene plasticity inference based on orthologous groups distribution.

Author(s)

Rodrigo JS Dalmolin, Mauro AA Castro.

References

Description

A dataset used to demonstrate geneplast main functions.

Usage

data(gpdata.gs)

Format

gpdata.gs A data frame containing orthology annotation.

Details

The dataset consists of 4 R objects to be used for demonstration purposes only in the geneplast vignette.

cogdata A data frame with three columns listing orthology annotation retrieved from the STRING database (http://string-db.org/), release 9.1. Column 1 = Ensembl protein ID; column 2 = NCBI species ID; column 3 = OG ID. Note: This dataset is to be used for demonstration purposes only as it represents a subset of the STRING database; in order to reduce the dataset size, orthology annotation was mapped to genome stability genes (Castro et al.).

sspids A data frame containing the species listed in STRING database (http://string-db.org/), release 9.1. Column 1 = NCBI species ID; column 2 = NCBI species name; column 3 = species domain (eukaryotes).

cogids A one-column data.frame listing orthologous groups (OGs) available in the `cogdata` object.

phyloTree An object of class "phylo" for the eukaryotes listed in the STRING database, release 9.1.

Value

a dataset.

References


Examples

data(gpdata.gs)
gplast  Evolutionary plasticity inference.

Description

Function to calculate abundance, diversity, and evolutionary plasticity of an orthologous group (OG).

Usage

gplast(object, verbose=TRUE)

Arguments

- object: this argument is an object of class 'OG' (OG-class).
- verbose: a single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).

Details

This method computes the abundance and diversity of an OG, and derives the evolutionary plasticity as described in Castro et al. (2008) and Dalmolin et al. (2011). The OG diversity corresponds to the normalized Shannon’s diversity index and estimates the distribution of orthologous proteins across the species listed in the input dataset. The OG abundance is simply the ratio between the number of orthologs of a given OG and the number of organisms listed in the group. Evolutionary Plasticity Index is calculated as described in Dalmolin et al (2011).

Value

A processed object of class 'OGP', including COG’s abundance, diversity, and plasticity results.

Author(s)

Rodrigo Dalmolin, Mauro Castro

References


See Also

OG-class
Examples

```r
#load datasets used for demonstration
data(gpdata.gs)

#create and object of class 'OGP'
ogp <- gplast.preprocess(cogdata=cogdata, sspids=sspids, cogids=cogids)

## run the gplast function
## this example uses the especies/COGs listed in the gpdata object
ogp <- gplast(ogp)
res <- gplast.get(ogp,what="results")
```

**gplast.get**

*Get information from individual slots in an OGP object.*

Description

Get information from individual slots in an OGP object and any available results from previous analysis.

Usage

```r
gplast.get(object, what="status")
```

Arguments

- `object`: an object of class 'OGP' **OGP-class**.
- `what`: a single character value specifying which information should be retrieved from the slots. Options: "cogids", "sspids", "orthodist", "results", and "status".

Value

slot content from an object of class 'OGP' **OGP-class**.

Author(s)

Rodrigo Dalmolin, Mauro Castro

Examples

```r
#load datasets used for demonstration
data(gpdata.gs)

#create and object of class 'OGP'
ogp <- gplast.preprocess(cogdata=cogdata, sspids=sspids, cogids=cogids)

## run the gplast function
## this example uses the species/COGs listed in the gpdata object
ogp <- gplast(ogp)
res <- gplast.get(ogp,what="results")

---

gplast.preprocess  
*Evolutionary plasticity inference.*

**Description**

Constructor for the 'OGP-class' object.

**Usage**

```r
gplast.preprocess(cogdata, sspids=NULL, cogids=NULL, verbose=TRUE)
```

**Arguments**

- `cogdata`: a data frame or matrix with COG's data.
- `sspids`: an optional data frame with species annotation. Alternatively, it can be a character vector with species IDs.
- `cogids`: an optional data frame with COG's annotation. Alternatively, it can be a character vector with COG IDs.
- `verbose`: a single logical value specifying to display detailed messages (when `verbose=TRUE`) or not (when `verbose=FALSE`).

**Details**

This function creates an OGP-class object and checks the consistency of the input data for the evolutionary plasticity pipeline. Internally, the function counts the number of orthologs for each species in a given OG and computes the orthodist matrix.

**Value**

A preprocessed object of class 'OGP'.

**Author(s)**

Rodrigo Dalmolin, Mauro Castro

**References**


groot

See Also

OGP-class

Examples

#load datasets used for demonstration
data(gpdata.gs)

#create an object of class 'OGP'
ogp <- gplast.preprocess(cogdata=cogdata, sspids=sspids, cogids=cogids)

Description

Function to determine the evolutionary root of a gene based on the distribution of its orthologs.

Usage

groot(object, method="BR", penalty=2, cutoff=0.3, nPermutations=1000, pAdjustMethod="bonferroni", verbose=TRUE)

Arguments

object

this argument is an object of class 'OGP' (OGP-class).

method

a single character value specifying the rooting algorithm. Options: "BR" and "KS" (see details).

penalty

a single numeric value specifying the penalty used in the rooting algorithm (see details).

cutoff

a single numeric value in [0,1] specifying the cutoff used in the BR statistics (see details).

nPermutations

a single integer value specifying the number of permutations used to compute a null distribution for the inferred roots in the species tree.

pAdjustMethod

a single character value specifying the p-value adjustment method to be used (see `p.adjust` for details).

verbose

a single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
**Details**

The 'groot' function addresses the problem of finding the evolutionary root of a feature in an phylogenetic tree. The method infers the probability that such feature was present in the Last Common Ancestor (LCA) of a given lineage. Events like horizontal gene transfer, gene deletion, and de novo gene formation add noise to vertical heritage patterns. The 'groot' function assesses the presence and absence of the orthologs in the extant species of the phylogenetic tree in order to build a probability distribution, which is used to identify vertical heritage patterns.

The 'penalty' argument weighs gene gain and loss; penalty=1 indicates equal probability; penalty > 1 indicates higher probability of gene loss while penalty < 1 indicates higher probability of gene gain (penalty value should be greater than zero; default penalty=2).

After the probability distribution is built for a given lineage, then a rooting algorithm is used to search the LCA that provides the best vertical heritage pattern (default method='BR'). The rooting algorithms finds the optimum point that splits the probability distribution into two components: one enriched with the queried feature (supporting vertical heritage in the lineage) and another with low evidence in favour of the feature’s presence. The cutoff sets the tolerance for the discrimination between the two components (default cutoff=0.3). Based on the optimization settings, then the 'groot' function computes the inconsistency score 'Dscore', which assesses the significance of the inferred root against a null distribution derived by permutation analysis.

**Value**

An processed object of class 'OGR', including results from the rooting algorithm.

**Author(s)**

Rodrigo Dalmolin, Mauro Castro

**References**


**See Also**

OGR-class

**Examples**

```r
#load datasets used for demonstration
data(gpdata.gs)

#create and object of class 'OGR' for H. sapiens
ogr <- groot.preprocess(cogdata=cogdata, phyloTree=phyloTree, spid="9606", cogids=cogids)

## run the groot function
## this example uses the orthologous groups listed in the gpdata object
ogr <- groot(ogr, nPermutations=100)
res <- groot.get(ogr, what="results")
```
## Not run:

# Option: parallel version with SNOW package!
library(snow)
options(cluster=makeCluster(2, "SOCK"))
g <- groot(g, nPermutations=100)
stopCluster(getOption("cluster"))

## End(Not run)

groot.get

Get information from individual slots in an OGR object.

**Description**

Get information from individual slots in an OGR object and any available results from a previous analysis.

**Usage**

groot.get(object, what="status")

**Arguments**

- **object**: an object of class `OGR` [OGR-class].
- **what**: a single character value specifying which information should be retrieved from the slots. Options: "cogids", "spbranches", "orthoroot", "results" and "status".

**Value**

slot content from an object of class `OGR` [OGR-class].

**Author(s)**

Rodrigo Dalmolin, Mauro Castro

**Examples**

```r
#load datasets used for demonstration
data(gpdata.gs)

#create and object of class 'OGR' for H. sapiens
ogr <- groot.preprocess(cogdata=cogdata, phyloTree=phyloTree, spid="9606", cogids=cogids)

## run the groot function
## this example uses the orthologous groups listed in the gpdata object
ogr <- groot(ogr, nPermutations=100)
res <- groot.get(ogr, what="results")
```
groot.plot

Plot the inferred evolutionary root of a given OG or the map of LCAs of a given species.

Description
Plot the inferred evolutionary root of a given OG onto the species tree or the map of LCAs of a given species.

Usage
```r
groot.plot(ogr, whichOG, fname="gproot", width=4.5, height=6.5, cex.lab=0.3, cex.nodes=0.6, adj.tips=c(1, 0.5), lab.offset=1.5, col.tips=c("green2","grey"), col.edges=c("black","grey"), col.root="red", plot.sspnames=TRUE, plot.subtree=FALSE, plot.lcas=FALSE)
```

Arguments
- `ogr` this argument is an object of class `OGR` evaluated by the groot `groot` method.
- `whichOG` a single character value indicating the OG to be plotted.
- `fname` a character string naming a file.
- `width` a single numeric value specifying the width of the graphics region in inches.
- `height` a single numeric value specifying the height of the graphics region in inches.
- `cex.lab` numeric character expansion factor for tip labels.
- `cex.nodes` numeric expansion factor for node symbols.
- `adj.tips` two numeric values specifying the adjustment of the labels.
- `lab.offset` a single numeric value specifying the offset of the labels.
- `col.tips` a character vector of length=2 specifying the colors of the tips.
- `col.edges` a character vector of length=2 specifying the colors of the edges.
- `col.root` a character value specifying the color of the inferred root.
- `plot.sspnames` a single logical value specifying whether ssp names should be used to generate the plot.
- `plot.subtree` a single logical value specifying whether a sub-species tree should be used to generate the plot.
- `plot.lcas` a single logical value specifying whether a species tree should be generated mapping the positions of all possible roots.

Value
a pdf file.

Author(s)
Rodrigo Dalmolin, Mauro Castro
groot.preprocess

References


See Also

groot

Examples

#load datasets used for demonstration
data(gpdata.gs)

#create and object of class 'OGR' for H. sapiens
ogr <- groot.preprocess(cogdata=cogdata, phyloTree=phyloTree, spid="9606", cogids=cogids)

## run the groot function
ogr <- groot(ogr, nPermutations=100)

## this example plots NOG40170 in the phyloTree
groot.plot(ogr, whichOG="NOG40170")

---

**groot.preprocess**   *Evolutionary rooting inference.*

**Description**

Constructor for the 'OGR-class' object.

**Usage**

groot.preprocess(cogdata, phyloTree, spid, cogids=NULL, verbose=TRUE)

**Arguments**

- **cogdata**
  a data frame with COG's data.
- **phyloTree**
  an object of class "phylo".
- **spid**
  a single character or integer value specifying the reference species to be used in the rooting algorithm. This species should be listed in the 'phyloTree'.
- **cogids**
  an optional data frame with COG's annotation. Alternatively, it can be a character vector with COG IDs.
- **verbose**
  a single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
Details

This function creates an OGR-class object and checks the consistency of the input data for the evolutionary root pipeline. Internally, the function access the presence and absence of orthologs for each species in a given OG and computes the orthoct data.frame.

Value

A preprocessed object of class 'OGR'.

Author(s)

Rodrigo Dalmolin, Mauro Castro

References


See Also

OGR-class

Examples

#load datasets used for demonstration
data(gpdata.gs)

#create and object of class 'OGR' for H. sapiens
ogr <- groot.preprocess(cogdata=cogdata, phyloTree=phyloTree, spid="9606", cogids=cogids)
**OGR-class**

**Slots**

- **sspids**: Object of class "data.frame", a data frame with species annotation.
- **cogids**: Object of class "data.frame", a data frame with COG's data.
- **orthodist**: Object of class "matrix", a matrix with COG's information (see return values in the OGP methods).
- **abundance**: Object of class "numeric", a numeric vector with results from the gplast function (see return values in the OGP methods).
- **diversity**: Object of class "numeric", a numeric vector with results from the gplast function (see return values in the OGP methods).
- **plasticity**: Object of class "numeric", a numeric vector with results from the gplast function (see return values in the OGP methods).
- **status**: Object of class "character", a character value specifying the status of the OGP object based on the available methods.

**Methods**

- `gplast signature(object = "OGP")`: see `gplast`
- `gplast.get signature(object = "OGP")`: see `gplast.get`

**Author(s)**

Rodrigo Dalmolin, Mauro Castro

**See Also**

`gplast.preprocess`

---

**OGR-class**

*Class "OGR": an S4 class for rooting analysis.*

**Description**

This S4 class includes methods to do inferential analysis of evolutionary roots in a given species tree.

**Objects from the Class**

Objects can be created by calls to the "groot.preprocess" constructor.
Slots

cogids: Object of class "data.frame", a data frame with COG's data.
tree: Object of class "phylo", a given species tree.
spbranches: Object of class "data.frame", a data frame listing branches of a given species tree.
orthoroot: Object of class "data.frame", a data.frame with results from the 'groot' function (see return values in the OGR methods).
orthoct: Object of class "data.frame", a data.frame with results from the 'groot.preprocess' function (see return values in the OGR methods).
status: Object of class "character", a character value specifying the status of the OGR object based on the available methods.

Methods

groot signature(object = "OGR"): see groot
groot.get signature(object = "OGR"): see groot.get

Author(s)
Rodrigo Dalmolin, Mauro Castro

See Also

groot.preprocess

Description
This function adds evolutionary rooting information to an 'igraph' class object.

Usage

ogr2igraph(ogr, cogdata, g, idkey = "ENTREZ")

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ogr</td>
<td>a processed object of class 'OGR' evaluated by the groot method.</td>
</tr>
<tr>
<td>cogdata</td>
<td>a data.frame with COG to protein mapping, with at least one gene annotation type listed in the 'rtni' object (e.g. ENTREZ gene ID).</td>
</tr>
<tr>
<td>g</td>
<td>a 'igraph' object.</td>
</tr>
<tr>
<td>idkey</td>
<td>a single character value specifying a vertex attribute name used to map 'cogdata' annotation to 'g'.</td>
</tr>
</tbody>
</table>
ogr2tni

Value
Return an updated `igraph` object with evolutionary root information.

Author(s)
Rodrigo Dalmolin, Mauro Castro, Sheyla Trefflich

References

See Also
OGR-class

Examples
```r
## Not run:
#This example requires the geneplast.data.string.v91 package! (currently available under request)
library(geneplast.data.string.v91)
data(gpdata_string_v91)
## End(Not run)
```

Description
This function adds evolutionary rooting information to regulons of a TNI class object.

Usage
ogr2tni(ogr, cogdata, tni)

Arguments
- **ogr**: a processed object of class `'OGR'` evaluated by the `groot` method.
- **cogdata**: a data.frame with COG to protein mapping, with at least one gene annotation type listed in the `rtni` object (e.g. ENTREZ gene ID).
- **tni**: a processed object of class `'TNI'` `TNI-class` evaluated by the methods `tni.permutation`, `tni.bootstrap` and `tni.dpi.filter`.

Value
Return an updated TNI object with evolutionary root information.
**Author(s)**

Rodrigo Dalmolin, Mauro Castro, Sheyla Trefflich

**References**


**See Also**

OGR-class

**Examples**

```r
## Not run:
#This example requires the geneplast.data.string.v91 package! (currently available under request)

# Evolutionary rooting inference for all human genes
library(geneplast.data.string.v91)
data(gpdata_string_v91)
cogids <- cogdata$cog_id[cogdata$ssp_id=="9606"]
ogr <- groot.preprocess(cogdata=cogdata, phyloTree=phyloTree, spid="9606", cogids=cogids)
ogr <- groot(ogr, nPermutations=100, verbose=TRUE)

# Integrates evolutionary rooting information and regulons
library(RTN)
library(Fletcher2013b)
data("rtni1st")
rtni1st <- ogr2tni(ogr, cogdata, rtni1st)

## End(Not run)
```

---

**ppi**

A pre-processed igraph object for the geneplast package.

**Description**

An igraph object used to demonstrate geneplast main functions.

**Usage**

data(ppi.gs)
**Format**

`ppi.gs` an igraph object.

**Details**

Protein-protein interactions (PPI) mapped to *H. sapiens* apoptosis and genome-stability genes. The PPI information is derived from the STRING database, release 9.1, using combined score >=900 (Franceschini et al., 2013).

**Value**

an igraph object.

**References**


**Examples**

`data(ppi.gs)`
Index

* classes
  OGP-class, 12
  OGR-class, 13
* dataset
  gpdata, 3
  ppi, 16
* methods
  gplast, 4
  gplast.get, 5
  gplast.preprocess, 6
  groot, 7
  groot.get, 9
  groot.plot, 10
  groot.preprocess, 11
  ogr2igraph, 14
  ogr2tni, 15
* package
  geneplast-package, 2

geneplast (geneplast-package), 2
geneplast-package, 2
gpdata, 3
gplast, 4, 13
gplast, OGP-method (OGP-class), 12
gplast.get, 5, 13
gplast.get, OGP-method (OGP-class), 12
gplast.preprocess, 6, 13
groot, 7, 10, 11, 14, 15
groot, OGR-method (OGR-class), 13
groot.get, 9, 14
groot.get, OGR-method (OGR-class), 13
groot.integration, OGR-method (OGR-class), 13
groot.plot, 10
groot.preprocess, 11, 14

OGP-class, 12
OGR-class, 13
ogr2igraph, 14
ogr2tni, 15

ppi, 16
tni.bootstrap, 15
tni.dpi.filter, 15
tni.permutation, 15