

# Package ‘RamiGO’

April 4, 2014

**Type** Package

**Title** AmiGO visualize R interface

**Version** 1.8.1

**Date** 2012-09-19

**Author** Markus Schroeder, Daniel Gusenleitner, John Quackenbush, Aedin Culhane, Benjamin Haibe-Kains

**Maintainer** Markus Schroeder <mschroed@jimmy.harvard.edu>

**Description** R interface sending requests to AmiGO visualize, retrieving DAG GO trees, parsing GraphViz DOT format files and exporting GML files for Cytoscape. Also uses RCytoscape to interactively display AmiGO trees in Cytoscape.

**License** Artistic-2.0

**LazyLoad** yes

**Depends** gsubfn,methods

**Imports** igraph,RCurl,png,RCytoscape,graph

**biocViews** GO, NetworkVisualization, GraphsAndNetworks, Classification, ConnectTools

## R topics documented:

adjM2gml . . . . .	2
AmigoDot . . . . .	3
AmigoDot-class . . . . .	4
AmigoDot.to.Cyto . . . . .	5
AmigoDot.to.graphAM . . . . .	6
AmigoDot.to.graphNEL . . . . .	7
c5.go.mapping . . . . .	7
exportCytoGML . . . . .	8
getAmigoTree . . . . .	9
igraph-class . . . . .	10
RamiGO . . . . .	11
readAmigoDot . . . . .	12

**Index****13**


---

adjM2gml	<i>Create GML file from readAmigoDot output.</i>
----------	--

---

**Description**

Takes a part of the output of the readAmigoDot function and creates a GML file by calling export-CytoGML().

**Usage**

```
adjM2gml(adjMatrix, edgecolor, vertexcolor,
         nodelabels, nodedescription, filename)
```

**Arguments**

adjMatrix	adjacency matrix, parents in rows, children in cols.
edgecolor	character vector of the length of existing edges.
vertexcolor	character vector of the length of existing nodes.
nodelabels	character vector of the length of existing nodes.
nodedescription	character vector of the length of existing nodes.
filename	output filenames.

**Author(s)**

Markus Schroeder <mschroed@jimmy.harvard.edu>  
 Benjamin Haibe-Kains <bhaibeka@jimmy.harvard.edu>

**Examples**

```
## set GO IDs and color
goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
color <- c("lightblue", "red", "yellow")

#dd <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example", picType="dot", saveResult=FALSE)
#tt <- readAmigoDot(object=dd)

#adjM2gml(adjMatrix(tt), relations(tt)$color,
# annot(tt)$fillcolor, annot(tt)$GO_ID,
# annot(tt)$description, "example")
```

**Description**

Functions for creating and manipulating AmigoDot-class objects.

**Usage**

```
agraph(object)
agraph(object) <- value
adjMatrix(object)
adjMatrix(object) <- value
annot(object)
annot(object) <- value
relations(object)
relations(object) <- value
leaves(object)
leaves(object) <- value
```

**Arguments**

object	An AmigoDot S4 object
value	An AmigoDot S4 object

**Details**

`agraph(object)` Gets the igraph object.

`agraph(object) <- value` Sets the igraph object.

`adjMatrix(object)` Gets the adjacency matrix.

`adjMatrix(object) <- value` Sets the adjacency matrix.

`annot(object)` Gets the data.frame containing the annotation of the tree with the columns node, GO\_ID, description, color, fillcolor and fontcolor.

`annot(object) <- value` Sets the the data.frame containing the annotation of the tree with the columns node, GO\_ID, description, color, fillcolor and fontcolor.

`relations(object)` Gets the data.frame containing the relations between the node of the tree and also information about the edges. The columns are parent, child, arrowhead, arrowtail, color and style.

`relations(object) <- value` Sets the the data.frame containing the relations between the node of the tree and also information about the edges. The columns are parent, child, arrowhead, arrowtail, color and style.

`leaves(object)` Gets the data.frame containing the leaves of the tree with the columns node, GO\_ID, description, color, fillcolor and fontcolor.

`leaves(object) <- value` Sets the the data.frame containing the leaves of the tree with the columns node, GO\_ID, description, color, fillcolor and fontcolor.

**Value**

AmigoDot	Object of class <a href="#">AmigoDot</a> .
agraph	igraph object.
adjMatrix	Adjacency Matrix.
annot, leaves	Annotation for each node.
relations	Meta information about the edges between nodes.

**Author(s)**

Markus Schroeder [mschroed@jimmy.harvard.edu](mailto:mschroed@jimmy.harvard.edu)

**See Also**

[AmigoDot](#)-class [igraph](#)

**Examples**

```
## set GO IDs and color
goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
color <- c("lightblue", "red", "yellow")

#dd <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example", picType="dot", saveResult=FALSE)
#tt <- readAmigoDot(object=dd)

#agraph(tt)
#adjMatrix(tt)
#annot(tt)
#relations(tt)
#leaves(tt)
```

---

AmigoDot-class

---

*Class "AmigoDot"*


---

**Description**

AmigoDot represents a collection of objects that are extracted from an GraphViz DOT format file.

**Objects from the Class**

Objects can be created by calls to the constructor [AmigoDot](#).

Elements can be accessed with [agraph](#), [adjMatrix](#), [annot](#), [relations](#), and [leaves](#).

**Slots**

**agraph:** Object of class "igraph" containing the graph extracted from the DOT format file.  
**adjMatrix:** Object of class "matrix" containing the adjacency matrix of the tree.  
**annot:** Object of class "data.frame" containing the annotation of the tree.  
**relations:** Object of class "data.frame" containing the relations between the node of the tree.  
**leaves:** Object of class "data.frame" containing the leaves of the tree.

**Methods**

See [AmigoDot](#)

**Author(s)**

Markus Schroeder [mschroed@jimmy.harvard.edu](mailto:mschroed@jimmy.harvard.edu)

**Examples**

```
showClass("AmigoDot")
```

---

AmigoDot.to.Cyto

*Opening the AmigoDot graph in Cytoscape through RCytoscape.*

---

**Description**

Opening the AmigoDot graph in Cytoscape through RCytoscape.

**Usage**

```
AmigoDot.to.Cyto(object)
```

**Arguments**

**object** is a AmigoDot S4 object.

**Details**

See <http://rcytoscape.systemsbiology.net/versions/current/> and <http://www.bioconductor.org/packages/release/bioc/html/RCytoscape.html> for details on how to install and use RCytoscape.

**Author(s)**

Markus Schroeder <[mschroed@jimmy.harvard.edu](mailto:mschroed@jimmy.harvard.edu)>

**Examples**

```
## set GO IDs and color
#goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
#color <- c("lightblue", "red", "yellow")
#
#dd <- getAmigoTree(goIDs=goIDs, color=color,
#  filename="example", picType="dot", saveResult=FALSE)
#tt <- readAmigoDot(object=dd)
#AmigoDot.to.Cyto(tt)
```

---

AmigoDot.to.graphAM	<i>Converts an AmigoDot S4 object to a graphAM object.</i>
---------------------	--

---

**Description**

Converts an AmigoDot S4 object to a graphAM object.

**Usage**

```
AmigoDot.to.graphAM(object)
```

**Arguments**

object                    is a AmigoDot S4 object.

**Value**

gAM                        is a graphAM object.

**Author(s)**

Markus Schroeder <mschroed@jimmy.harvard.edu>

**Examples**

```
## set GO IDs and color
goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
color <- c("lightblue", "red", "yellow")

#dd <- getAmigoTree(goIDs=goIDs, color=color,
#  filename="example", picType="dot", saveResult=FALSE)
#tt <- readAmigoDot(object=dd)
#AmigoDot.to.graphAM(tt)
```

---

AmigoDot.to.graphNEL     *Converts an AmigoDot S4 object to a graphNEL object.*

---

### Description

Converts an AmigoDot S4 object to a graphNEL object.

### Usage

```
AmigoDot.to.graphNEL(object)
```

### Arguments

object                    is a AmigoDot S4 object.

### Value

gNEL                    is a graphNEL object.

### Author(s)

Markus Schroeder <mschroed@jimmy.harvard.edu>

### Examples

```
## set GO IDs and color
goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
color <- c("lightblue", "red", "yellow")

#dd <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example", picType="dot", saveResult=FALSE)
#tt <- readAmigoDot(object=dd)
#AmigoDot.to.graphNEL(tt)
```

---

c5.go.mapping                    *MSigDB C5 GO term to GO ID mapping*

---

### Description

This object provides a mapping for MSigDB GO terms for the bp, mf and cc GO categories to official GO ID's.

### Format

A data frame with 1454 observations on the following 2 variables.

description a character vector of MSigDB GO terms

goid a character vector of official GO ID's

**Source****c5.go.mapping:**<http://www.broadinstitute.org/gsea/msigdb/collections.jsp#C5>**References**

Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* 2005 Oct;**102**(43):15545 -15550.

**Examples**

```
## load the mapping
data(c5.go.mapping)

## look at the object
str(c5.go.mapping)

## create go term vector
terms <- c("CHROMATIN_REMODELING_COMPLEX",
"RNA_POLYMERASE_COMPLEX",
"CYTOKINESIS", "CELL_RECOGNITION")
id <- sapply(terms, function(x) c5.go.mapping[
  match(x, c5.go.mapping[, 1]), 2])
id
```

---

exportCytoGML

---

*Writes out an igraph graph to a Cytoscape readable GML file.*


---

**Description**

Takes the igraph object edited in adjM2gml() and writes it to a GML file that is readable by Cytoscape.

**Usage**

```
exportCytoGML(graph, filename)
```

**Arguments**

graph	igraph graph (for example from adjM2gml()).
filename	output filename.

**Author(s)**

Markus Schroeder <mschroed@jimmy.harvard.edu>

Benjamin Haibe-Kains <bhaibeka@jimmy.harvard.edu>



## Examples

```
## set GO IDs and color
goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
color <- c("lightblue", "red", "yellow")

## getAmigoTree
dd <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example", picType="dot", saveResult=FALSE)
tt <- readAmigoDot(object=dd)

## exportCytoGML is called inside adjM2gml
adjM2gml(adjMatrix(tt), relations(tt)$color,
# annot(tt)$fillcolor, annot(tt)$GO_ID,
# annot(tt)$description, "example")
```

---

getAmigoTree

*Getting the DAG GO tree for a set of GO ID's.*


---

## Description

Getting the DAG GO tree for a set of GO ID's from AmiGO. Saves it as a png, svg or dot file. Returns the webserver response as a string.

## Usage

```
getAmigoTree(goIDs, color, pvalues, pcolors = c("white", "tomato"),
psplit = c(1, 0.25, 0.1, 0.05, 0.001), filename, picType = "png",
modeType = "advanced", webserver, saveResult = TRUE)
```

## Arguments

goIDs	is a vector of GO ID's.
color	is a vector of colors(). Either of length 1 or length(goIDs).
pvalues	is a vector of pvalues.
pcolors	colors for gradient.
psplit	is a vector of pvalues that sets different color gradients ranges.
filename	is the output filename. If the file ending is missing or not matching the picType, the file ending is automatically fixed (replaced or created).
picType	supports "png", "svg" and "dot". default is "png".
modeType	is the AmiGO query mode, only "advanced" supported.
webserver	is by default the AmiGO webserver. You can specify another one.
saveResult	saving the results as a png, svg or dot file. Default is TRUE.

## Details

The edge colors in the tree represent the relationship between two nodes. In particular: green=positively regulates, red=negatively regulates, black=regulates, blue=is a, light blue=part of. The tree represents the set of input GO ID's and all parents of those GO ID's.

**Value**

res                      string with the webserver response.

**Author(s)**

Markus Schroeder <mschroed@jimmy.harvard.edu>

**Examples**

```
## set GO IDs and color
goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
color <- c("lightblue", "red", "yellow")

## get results
#pp <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example")
#ss <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example", picType="svg", saveResult=FALSE)
#dd <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example", picType="dot", saveResult=FALSE)

#ppNoColor <- getAmigoTree(goIDs=goIDs,
# filename="example-no-color", saveResult=FALSE)

#goIDs <- c("GO:0051130", "GO:0050789", "GO:0019912", "GO:0016301", "GO:0003824", "GO:0005783", "GO:0043229")
#pvalues <- c(0.001, 0.5, 0.0001, 0.16, 0.47, 0.00006, 0.002)

#ptest <- getAmigoTree(goIDs=goIDs, pvalues=pvalues, filename="pvaluetest")
```

---

igraph-class

Class "igraph"

---

**Description**

This is an igraph object.

**Objects from the Class**

A virtual Class: No objects may be created from it.

**Slots**

.S3Class: Object of class "character"

**Extends**

Class "[oldClass](#)", directly.

**Methods**

See [igraph](#) for functions.

**References**

See [igraph](#)

---

RamiGO

*AmiGO visualize R interface*

---

**Description**

R interface sending requests to AmiGO visualize, retrieving DAG GO trees, parsing GraphViz DOT format files and exporting GML files for Cytoscape.

**Details**

Package:	RamiGO
Type:	Package
Version:	1.8.1
Date:	2012-09-19
License:	Artistic-2.0
LazyLoad:	yes

**Author(s)****Markus Schroeder**

- Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Harvard School of Public Health, Boston, MA, USA

<http://compbio.dfci.harvard.edu/>

- Computational Genomics, Center for Biotechnology (CeBiTec), Bielefeld University, Germany

<http://www.cebitec.uni-bielefeld.de/cebitec/computational-genomics/home.html>

**Maintainer:**

Markus Schroeder

<mschroed@jimmy.harvard.edu>

**References**

Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium (2000) Nature Genet. 25: 25-29

<http://amigo.geneontology.org/cgi-bin/amigo/go.cgi>

---

`readAmigoDot`*Parser for the GraphViz DOT format.*

---

**Description**

Reads DOT file or takes the output of `getAmigoTree` (with `picType="dot"`) and returns an `AmigoDot` S4 object with a graph, adjacency matrix, edges and leaves and also the annotation for the nodes.

**Usage**

```
readAmigoDot(object, filename)
```

**Arguments**

<code>object</code>	A newline separated string in DOT format (as returned by <code>getAmigoTree</code> ).
<code>filename</code>	A .dot file in DOT format.

**Value**

An `AmigoDot` S4 object.

**Author(s)**

Markus Schroeder <mschroed@jimmy.harvard.edu>

**Examples**

```
## set GO IDs and color
goIDs <- c("GO:0051130", "GO:0019912", "GO:0005783")
color <- c("lightblue", "red", "yellow")

## get dot file / object
#dd <- getAmigoTree(goIDs=goIDs, color=color,
# filename="example", picType="dot", saveResult=TRUE)

## parse file or object
##tt1 <- readAmigoDot(filename="example.dot")
##tt2 <- readAmigoDot(object=dd)

## look at results
##show(tt1)
#show(tt2)
```

# Index

- \*Topic **Cytoscape**
  - adjM2gml, [2](#)
  - exportCytoGML, [8](#)
- \*Topic **classes**
  - AmigoDot, [3](#)
  - AmigoDot-class, [4](#)
  - igraph-class, [10](#)
- \*Topic **datasets**
  - c5.go.mapping, [7](#)
- \*Topic **graph**
  - adjM2gml, [2](#)
  - exportCytoGML, [8](#)
- \*Topic **igraph**
  - adjM2gml, [2](#)
  - exportCytoGML, [8](#)
- \*Topic **network**
  - adjM2gml, [2](#)
  - exportCytoGML, [8](#)

adjM2gml, [2](#)  
adjMatrix, [4](#)  
adjMatrix (AmigoDot), [3](#)  
adjMatrix, AmigoDot-method (AmigoDot), [3](#)  
adjMatrix<- (AmigoDot), [3](#)  
adjMatrix<-, AmigoDot-method (AmigoDot),  
[3](#)  
agraph, [4](#)  
agraph (AmigoDot), [3](#)  
agraph, AmigoDot-method (AmigoDot), [3](#)  
agraph<- (AmigoDot), [3](#)  
agraph<-, AmigoDot-method (AmigoDot), [3](#)  
AmigoDot, [3](#), [4](#), [5](#)  
AmigoDot-class, [4](#)  
AmigoDot.to.Cyto, [5](#)  
AmigoDot.to.graphAM, [6](#)  
AmigoDot.to.graphNEL, [7](#)  
annot, [4](#)  
annot (AmigoDot), [3](#)  
annot, AmigoDot-method (AmigoDot), [3](#)  
annot<- (AmigoDot), [3](#)  
annot<-, AmigoDot-method (AmigoDot), [3](#)  
c5.go.mapping, [7](#)  
exportCytoGML, [8](#)  
getAmigoTree, [9](#)  
igraph, [4](#), [11](#)  
igraph-class, [10](#)  
leaves, [4](#)  
leaves (AmigoDot), [3](#)  
leaves, AmigoDot-method (AmigoDot), [3](#)  
leaves<- (AmigoDot), [3](#)  
leaves<-, AmigoDot-method (AmigoDot), [3](#)  
oldClass, [10](#)  
RamiGO, [11](#)  
readAmigoDot, [12](#)  
relations, [4](#)  
relations (AmigoDot), [3](#)  
relations, AmigoDot-method (AmigoDot), [3](#)  
relations<- (AmigoDot), [3](#)  
relations<-, AmigoDot-method (AmigoDot),  
[3](#)