

# *RamiGO*: an R interface for amiGO

Markus Schröder<sup>1,2</sup>, Aedín Culhane<sup>1</sup>, Benjamin Haibe-Kains<sup>1</sup>, and John Quackenbush<sup>1</sup>

<sup>1</sup>*Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Harvard School of Public Health, Boston, USA*

<sup>2</sup>*Computational Genomics, Center for Biotechnology, Bielefeld University, Germany*

June 1, 2011

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Getting started</b>	<b>1</b>
<b>3</b>	<b>Example</b>	<b>2</b>
<b>4</b>	<b>Session Info</b>	<b>5</b>

## 1 Introduction

The *RamiGO* package is providing functions to interact with amiGO and retrieving GO (Gene Ontology) trees in various formats. The most common requests would be as png or svg. RamiGO also provides a parser for the GraphViz DOT format that returns a graph object and meta data.

## 2 Getting started

The DOT format parser uses the `strapply` function from the *gsubfn* package to extract the information from the DOT format file. The *gsubfn* package has to be installed on your system.

```
> library(RamiGO)
> library(gsubfn)
```

`strapply` enables perl-like regular expression in R, as do `grep`, `sub` or `gsub`. In particular, it enables the use of the perl variables `$1`, `$2`, ... for extracting information from within a regular expression. The code below shows an example of the use of `strapply`. The string within brackets (...) is returned in a list by `strapply`.

```
> strapply(c("node25 -> node30"), "node([\\d]+) -> node([\\d]+)",
+         c, backref = -2)
```

```
[[1]]
[1] "25" "30"
```

### 3 Example

The RamiGO package currently provides two functions that enable the user to retrieve directed acyclic trees from amiGO and parse the GraphViz DOT format. An example on how to use the functions is given below.

To retrieve a tree from amiGO, the user has to provide a vector of GO ID's. For example GO:0051130, GO:0019912, GO:0005783, GO:0043229 and GO:0050789. These GO ID's represent entries from the three GO categories: Biological Process, Molecular Function and Cellular Component. The given GO ID's can be highlighted with different colors within the tree, therefore the user has to provide a vector of colors for each GO ID. A request could look like this:

```
> getAmigoTree(picType = "png", goIDs = c("GO:0051130", "GO:0019912",
+     "GO:0005783", "GO:0043229", "GO:0050789"), color = c("lightblue",
+     "red", "yellow", "green", "pink"), filename = "example")
```

The GO tree representing the given GO ID's is downloaded to the file "example.png" (see Figure 1); the file extension is created automatically according to picType. The request for a svg file is similar:

```
> getAmigoTree(picType = "svg", goIDs = c("GO:0051130", "GO:0019912",
+     "GO:0005783", "GO:0043229", "GO:0050789"), color = c("lightblue",
+     "red", "yellow", "green", "pink"), filename = "example")
```

In order to further analyze the tree, amiGO provides the possibility to retrieve the tree in the GraphViz DOT format. The function readAmigoTree parses these DOT format files and returns multiple objects. A graph object, an adjacency matrix representing the graph, a data.frame with the annotation for each node, the relations (edges) between the nodes and a data.frame with the leaves of the tree and their annotation. An example could look like this:

```
> getAmigoTree(picType = "dot", goIDs = c("GO:0051130", "GO:0019912",
+     "GO:0005783", "GO:0043229", "GO:0050789"), color = c("lightblue",
+     "red", "yellow", "green", "pink"), filename = "example")
> tt <- readAmigoDot("example.dot")
> str(tt)
```

```
List of 5
 $ graph      :List of 9
```

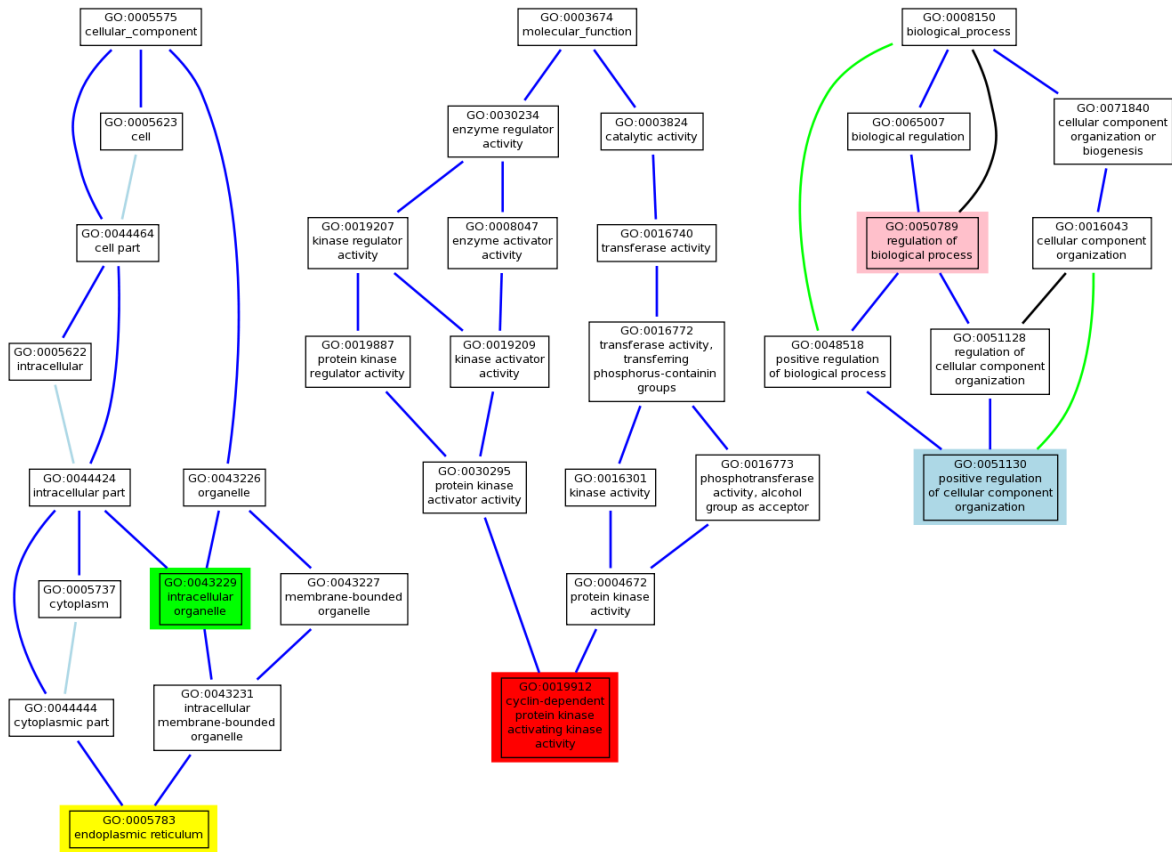


Figure 1: Example PNG returned from amiGO.

```

..$ : num 34
..$ : logi TRUE
..$ : num [1:46] 0 0 1 2 3 4 4 5 6 6 ...
..$ : num [1:46] 1 6 6 3 10 5 19 26 7 15 ...
..$ : num [1:46] 0 1 2 3 4 5 6 7 8 9 ...
..$ : num [1:46] 28 31 0 23 3 38 18 5 1 2 ...
..$ : num [1:35] 0 2 3 4 5 7 8 11 12 13 ...
..$ : num [1:35] 0 2 3 4 6 7 8 10 12 14 ...
..$ :List of 4
.. ..$ : num [1:2] 1 0
.. ..$ : Named list()
.. ..$ :List of 1
.. .. ..$ name: chr [1:34] "node1" "node2" "node3" "node4" ...
.. ..$ : list()
..- attr(*, "class")= chr "igraph"
$ adjMatrix: num [1:34, 1:34] 0 0 0 0 0 0 0 0 0 0 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:34] "node1" "node2" "node3" "node4" ...

```

```

.. ..$ : chr [1:34] "node1" "node2" "node3" "node4" ...
$ annot      : 'data.frame':      34 obs. of  6 variables:
..$ node      : chr [1:34] "node1" "node2" "node3" "node4" ...
..$ GO_ID      : chr [1:34] "GO:0044464" "GO:0005622" "GO:0016301" "GO:0004672" ...
..$ description: chr [1:34] "cell part" "intracellular" "kinase activity" "protein kinase activity" ...
..$ color      : chr [1:34] "#000000" "#000000" "#000000" "#000000" ...
..$ fillcolor  : chr [1:34] "#ffffff" "#ffffff" "#ffffff" "#ffffff" ...
..$ fontcolor  : chr [1:34] "#000000" "#000000" "#000000" "#000000" ...
$ relations: 'data.frame':      46 obs. of  6 variables:
..$ parent     : chr [1:46] "node32" "node32" "node26" "node4" ...
..$ child      : chr [1:46] "node26" "node14" "node17" "node11" ...
..$ arrowhead  : chr [1:46] "none" "none" "none" "none" ...
..$ arrowtail  : chr [1:46] "normal" "normal" "normal" "normal" ...
..$ color      : chr [1:46] "blue" "blue" "blue" "blue" ...
..$ style      : chr [1:46] "bold" "bold" "bold" "bold" ...
$ leaves      : 'data.frame':      3 obs. of  6 variables:
..$ node       : chr [1:3] "node10" "node11" "node34"
..$ GO_ID       : chr [1:3] "GO:0005783" "GO:0019912" "GO:0051130"
..$ description: chr [1:3] "endoplasmic reticulum" "cyclin-dependent protein kinase activating kinase activity" "positive regulation of cellular component organization"
..$ color       : chr [1:3] "#000000" "#000000" "#000000"
..$ fillcolor   : chr [1:3] "yellow" "red" "lightblue"
..$ fontcolor   : chr [1:3] "#000000" "#000000" "#000000"

```

The leaves of the tree are:

```
> tt$leaves[, c("node", "GO_ID", "description")]
```

	node	GO_ID	description
10	node10	GO:0005783	endoplasmic reticulum
11	node11	GO:0019912	cyclin-dependent protein kinase activating kinase activity
34	node34	GO:0051130	positive regulation of cellular component organization

## 4 Session Info

- R version 2.13.0 (2011-04-13), i686-pc-linux-gnu
- Locale: LC\_CTYPE=en\_US.utf8, LC\_NUMERIC=C, LC\_TIME=en\_US.utf8, LC\_COLLATE=en\_US.utf8, LC\_MONETARY=C, LC\_MESSAGES=en\_US.utf8, LC\_PAPER=en\_US.utf8, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US.utf8, LC\_IDENTIFICATION=C
- Base packages: base, datasets, graphics, grDevices, methods, stats, tcltk, tools, utils
- Other packages: ade4 1.4-17, cacheSweave 0.4-5, codetools 0.2-8, filehash 2.1-1, formatR 0.2-2, getopt 1.16, GSA 1.03, gsubfn 0.5-5, highlight 0.2-5, optparse 0.9.1, parser 0.0-13, pgfSweave 1.2.1, proto 0.3-9.2, RamiGO 0.1, Rcpp 0.9.4, stashR 0.3-3, tikzDevice 0.6.1
- Loaded via a namespace (and not attached): digest 0.4.2, igraph 0.5.5-2