gaggle

October 25, 2011

`broadcast` *Broadcast Data to the Gaggle*

Description

Broadcast a list of names, a matrix, a network, an associative array (an R environment) or a `cluster` to the `currentTargetGoose`. A `cluster` is simply two lists of names, typically a list of matrix row names, and a list of column names, which together identify a cluster of genes which behave coherently in the specified conditions.

Usage

```
broadcast (x, name='from R')
```

Arguments

`x`	list of names, matrix, network, environment, cluster.
`name`	any string, especially useful for identifying matrices or environments broadcast to other geese.

Value

Returns NULL.

See Also

`geese`. `setTargetGoose`. `getTargetGoose`.

connectToGaggle *Connect to the Gaggle Boss*

Description

Call this function to connect to a gaggle boss that has already been initialized (with gaggleInit()) and has subsequently been disconnected.

Usage

```
connectToGaggle()
```

Value

Returns NULL.

See Also

gaggleInit. disconnectFromGaggle.

disconnectFromGaggle *Disconnect from the Gaggle Boss*

Description

Call this function to disconnect from the Gaggle Boss.

Usage

```
disconnectFromGaggle()
```

Value

Returns NULL.

See Also

connectToGaggle.

gaggleInit — *Initialize the R Goose, registering with the Gaggle*

Description

Call this function to register the R Goose with a Gaggle Boss. If you do not specify an argument, the function will automatically start the Gaggle Boss for you if it is not already running. If you do specify an argument (a remote machine on which a boss is running), the boss must be running on that machine or this function will fail. By default, this function looks for, and registers with, a boss on your current computer ('localhost'). Alternatively, you may supply the name of another computer where a boss is running. This is useful if you wish to run R on a remote compute server (perhaps with extra memory and speed), and connect it to a gaggle running on your desktop.

Usage

```
gaggleInit (bossHost = 'localhost')
```

Arguments

bossHost	the computer ('host') on which your GaggleBoss is running. *The host machine must be accessible through any firewall (port 1099 must be open).*

Value

Returns NULL.

See Also

broadcast. getNameList. getMatrix. getNetwork. getCluster. getTuple.

geese — *What Geese (Programs) are Currently Running in the Gaggle?*

Description

Data can be broadcast to, or recevied from, any program currently running in the Gaggle. You can broadcast to `boss` or to `all`, and let the current Boss settings determine which geese get the data. Or you can use `setTargetGoose` to specify one particular goose – this is the most common approach. The current function supports that approach: it returns the names of all currently running geese.

Usage

```
geese ()
```

Value

Returns the names of all currently running geese as a list of character strings.

See Also

setTargetGoose. getTargetGoose. broadcast.

getCluster *Create an R variable from Most Recently Broadcast Cluster*

Description

After a cluster is broadcast to the R goose, one additional step is required of the R user before it is available as a character list variable in R. (This step may be eliminated in the future.)

Usage

```
getCluster ()
```

Value

Returns a list with three fields: name, rowNames, columnNames.

See Also

getNameList. getMatrix. getNetwork.

getJavaClassName *Get Java class name*

Description

Get the class name of a Java object for debugging purposes. Used with RJava objects.

Usage

```
getJavaClassName(jobj, quiet=FALSE)
```

Arguments

jobj	a Java object
quiet	complain if jobj isn't a java object

Value

Returns a string.

getMatrix *Create an R matrix variable from Most Recently Broadcast Matrix*

Description

After a matrix is broadcast to the R goose, one additional step is required of the R user before it is available as a matrix variable. (This step may be eliminated in the future.)

Usage

```
getMatrix ()
```

Value

Returns a matrix with double precision elements, row names, and column names.

See Also

getNameList. getNetwork.

getNameList *Create an R variable from Most Recently Broadcast NameList*

Description

After a nameList is broadcast to the R goose, one additional step is required of the R user before it is available as a character list variable in R. (This step may be eliminated in the future.)

Usage

```
getNameList ()
```

Value

Returns a list of names.

See Also

getMatrix. getNetwork.

getNetwork *Create an R network variable from a Broadcast Network*

Description

After a network is broadcast to the R goose, one additional step is required of the R user before it is available as a network variable. (This step may be eliminated in the future.)

Usage

```
getNetwork (directed=TRUE)
```

Arguments

directed	Whether the network is directed

Value

Returns a GraphNEL object.

See Also

getNameList. getMatrix.

getSpecies *Specify the Species To Which Subsequent Data Belongs*

Description

A gaggle broadcast always includes the species (the organism) the data is associated with. For example, the KEGG goose must tell the KEGG web server what organism the accomanying gene names are from. The R Goose has a notion of *default species*; this command is a convenient way to learn the current value of that default.

Usage

```
getSpecies ()
```

Value

Returns the name of the current species.

See Also

setSpecies. broadcast.

`getTargetGoose` *Which Goose will Receive Broadcasts?*

Description

A gaggle broadcast usually (though not necessarily) goes to a specific goose; this function will tell you which one. Note that if the current target goose is *'all'* or *'boss'* then the actual distribution of your broadcast will be controlled by setting you have made in the graphical user interface to the Gaggle Boss.

Usage

```
getTargetGoose ()
```

Value

Returns a goose name, or *'all'* or *'boss'*

See Also

`setTargetGoose`. `broadcast`.

`getTuple` *Create an R variable from Most Recently Broadcast Tuple*

Description

After a tuple is broadcast to the R goose, this method makes it available as an R environment object.

Usage

```
getTuple ()
```

Value

Returns an environment with a title, and then the usual keys and values.

See Also

`newTuple`. `getNameList`. `getCluster`. `getMatrix`. `getNetwork`.

hideGoose *Hide a Goose from View*

Description

If a goose has a graphical user interface, then it can usually be hidden from view, that is, iconified. This command accomplishes that.

Usage

```
hideGoose (target=NULL)
```

Arguments

target	If missing, or NULL, then the currentTargetGoose is hidden. If called with the name of one of the current geese, that goose will be hidden.

Value

Returns NULL.

See Also

geese. getTargetGoose. setTargetGoose. showGoose.

newTuple *Manipulate the Gaggle Tuple data type*

Description

Gaggle Tuples are lists of key/value pairs which can be nested. These functions support creation of tuple objects for broadcast to other Gaggle-connected software or conversion of tuples to R list objects.

The function *getTupleAsList* receives a Gaggle Tuple broadcast by converting the received value to a list. This is an alternative to the older *getTuple* which returns an environment object.

Usage

```
newTuple(name, list=NULL)
newSingle(name, value)
addSingle(tuple, name=NULL, value=NULL, single=NULL)
tupleToList(tuple)
getTupleAsList()
```

Arguments

name	name to be assigned to the tuple or individual value
list	a list of items to be converted to Singles and added to the tuple
value	the value of a key/value pair
tuple	a java tuple object
single	a java single object

See Also

getTuple.broadcast.

Examples

```
t <- newTuple('sequence', list(name='chromosome', length=2014239))
s <- newSingle('topology', 'circular')
addSingle(t, single=s)
addSingle(t, name='color', 'blue')
tupleToList(t)
# expected output:
#  $name
#    [1] "chromosome"
#  $length
#    [1] 2014239
#  $topology
#   [1] "circular"
#  $color
#    [1] "blue"
```

---

setSpecies *Specify the Species To Which Subsequent Data Belongs*

Description

A gaggle broadcast always includes the species (the organism) the data is associated with. For example, the KEGG goose must tell the KEGG web server what organism the accomanying gene names are from. The R Goose has a notion of *default species*; this command is a convenient way to set that default.

Usage

```
setSpecies (newValue)
```

Arguments

newValue	typically the Linnean name, e.g., 'Homo sapiens'

See Also

getSpecies.broadcast.

setTargetGoose *Specify Which Goose will Receive Broadcasts*

Description

A gaggle broadcast usually (though not necessarily) goes to a specific goose; this function allows you to specify which one. Note that if the current target goose is *'all'* or *'boss'* then the actual distribution of your broadcast will be controlled by setting you have made in the graphical user interface to the Gaggle Boss.

Usage

```
setTargetGoose (gooseName)
```

Arguments

gooseName	any current goose name, or *'all'* or *'boss'*

Value

Returns NULL

See Also

geese. getTargetGoose. broadcast.

showGoose *Show a Goose On the Screen*

Description

If a goose has a graphical user interface, then it can usually be hidden from view, that is, iconified. This command *reverses* that action, making sure that is visible..

Usage

```
showGoose (target=NULL)
```

Arguments

target	If missing, or NULL, then the currentTargetGoose is displayed. If called with the name of one of the current geese, that goose will be displayed.

Value

Returns NULL.

See Also

geese. getTargetGoose. setTargetGoose. hideGoose.

`testGagglePackage` *Run unit tests for the Gaggle Package*

Description

There are five data types supported in the Gaggle:

1. nameLists
2. matrices
3. networks
4. tuples (environments in R)
5. clusters (typically, row names and associated column names, selecting a submatix)

A good test of an installation, therfore, is to send each of these data types from R to an 'echo goose', and to make sure that the returned data is identical to that which was sent out. This function loads the `RUnit` package, assumes that a boss and echo goose are running and properly configured, then calls one function for each data type.

Procedure:

1. Browse to http://gaggle.systemsbiology.net/projects/rValidation
2. start the Gaggle Boss
3. start the echo goose
4. (back in R): testGagglePackage ()

For more details, please examine the source: unitTests/gaggleTest.R.

Usage

```
testGagglePackage ()
```

Value

the status of the last test

Index