

# guildifyR

February 9, 2018

---

<code>get.species.info</code>	<i>Get info on available species and tissues</i>
-------------------------------	--

---

## Description

Get info on available species and tissues

## Usage

```
get.species.info()
```

## Value

result List containing available species tax ids and names for tissue-specific networks

## Examples

```
result = get.species.info()
```

---

<code>query</code>	<i>Get protein/gene info associated with the query keywords</i>
--------------------	---

---

## Description

Get protein/gene info associated with the query keywords

## Usage

```
query(keywords, species = "9606", tissue = "All", quote.keywords = T)
```

## Arguments

<code>keywords</code>	Text containing description of a phenotype or list of genes (seperated by ";")
<code>species</code>	Species tax identifier (9606: human, 10090: mouse, etc., see <code>get.species.info</code> method)
<code>quote.keywords</code>	Quotes the keywords to treat whitespaces as ANDs (e.g., "Muscular Dystrophy")
<code>tissues</code>	Tissue identifier (All, brain, liver, etc., see <code>get.species.info</code> method)

**Value**

result.table Data frame containing list of matching proteins/genes and their description

**Examples**

```
result.table = query("alzheimer", species="10090", tissue="All")
```

---

retrieve.job	<i>Retrieve results using job id</i>
--------------	--------------------------------------

---

**Description**

Retrieve results using job id

**Usage**

```
retrieve.job(job.id, n.top = NULL, fetch.files = F, output.dir = NULL)
```

**Arguments**

job.id	Job id
n.top	Number of top proteins to retrieve. If NULL top functionally enriched proteins are provided (upto 500 proteins)
fetch.files	Flag to fetch result files from server and save them locally in output.dir (If NULL files for top functionally enriched proteins are provided, otherwise files for top-ranking 1% are provided)
output.dir	Directory to save the ranking, function, subnetwork and drug info files fetched from the server. If NULL, a folder with the same as name as the job.id is created in the current working directory

**Value**

result List containing scores of top-ranking proteins, functions enriched among top-ranking proteins, drugs targeting top-ranking proteins, top-ranking subnetwork cutoff

**Examples**

```
result = retrieve.job(job.id)
getSlots(class(result))
#Scores
head(gScores(result))
#Functions
head(gFunctions(result))
#Drugs
head(gDrugs(result))
```

---

retrieve.overlap	<i>Retrieve overlap between two results using two job ids</i>
------------------	---

---

### Description

Retrieve overlap between two results using two job ids

### Usage

```
retrieve.overlap(job.id1, job.id2, top.validated = T, fetch.files = F,
  output.dir = NULL)
```

### Arguments

job.id1	Job id 1
job.id2	Job id 2
top.validated	Flag to get overlap between either top-ranking functionally validated proteins (default) or top top-ranking 1%
fetch.files	Flag to fetch result files (for top-ranking 1%) from server and save them locally in output.dir
output.dir	Directory to save the ranking, function, subnetwork and drug info files fetched from the server

### Value

result List containing scores of common top-ranking proteins, common functions enriched among top-ranking proteins, drugs targeting common top-ranking proteins (Note that the number of top-ranking proteins and common functions are limited to 500)

### Examples

```
result = retrieve.overlap(job.id1, job.id2)
getSlots(class(result))
#Scores
head(gScores(result))
#Functions
head(gFunctions(result))
#Drugs
head(gDrugs(result))
```

---

submit.job	<i>Submit job using the guildifyR query result table</i>
------------	--

---

### Description

Submit job using the guildifyR query result table

**Usage**

```
submit.job(result.table, species, tissue, scoring.options = list(netcombo =  
  T))
```

**Arguments**

scoring.options	Prioritization method options (default NetCombo)
query.table	Data frame containing information from guildifyR query (user entity info, network info, source etc...)

**Value**

job id

**Examples**

```
species="10090"  
tissue="All"  
result.table = query("alzheimer", species)  
#Diamond  
scoring.options = list(diamond=T)  
#NetScore  
scoring.options = list(netscore=T, repetitionSelector=3, iterationSelector=2)  
#NetZcore  
scoring.options = list(netzcore=T, repetitionZelector=3)  
#NetShort  
scoring.options = list(netshort=T)  
#NetCombo  
scoring.options = list(netcombo=T)  
job.id = submit.job(result.table, species, tissue, scoring.options)
```

# Index

`get.species.info`, [1](#)

`query`, [1](#)

`retrieve.job`, [2](#)

`retrieve.overlap`, [3](#)

`submit.job`, [3](#)