# Package 'aakmisc'

April 18, 2013

**Type** Package

**Title** Miscellany for the King lab

**Version** 0.5-2

**Date** 2013-04-18

**Author** Aaron A. King

**Maintainer** Aaron A. King <kingaa@umich.edu>

**Author@R**
c(person(given=c(''Aaron'',''A.''),family=''King'',role=c(''aut'',''cre''),email=''kingaa@umich.edu''))

**Description** Miscellaneous tools.

**Depends** R(>= 3.0.0), plyr, reshape2, ggplot2, scales, RPostgreSQL, DBI

**Imports** knitr

**VignetteBuilder** knitr

**License** Private

**LazyLoad** true

**LazyData** true

**BuildVignettes** no

**Collate** aaa.R version.R db.R tunnel.R random.R

## R topics documented:

---

**Description**

Interface with project databases.

**Usage**

```
getMLEs(host = getOption("aakmisc.dbhost"),
        dbname = getOption("aakmisc.dbname"),
        ...)
recMLEs(mle, host = getOption("aakmisc.dbhost"),
        dbname = getOption("aakmisc.dbname"),
        ...)
listScripts(host = getOption("aakmisc.dbhost"),
            dbname = getOption("aakmisc.dbname"),
            ...)
recScript(files, host = getOption("aakmisc.dbhost"),
          dbname = getOption("aakmisc.dbname"),
          ...)
dropScript(script, host = getOption("aakmisc.dbhost"),
           dbname = getOption("aakmisc.dbname"),
           ...)
catScript(script, file = "", host = getOption("aakmisc.dbhost"),
          dbname = getOption("aakmisc.dbname"),
          ...)
getQuery(statement,
         host = getOption("aakmisc.dbhost"),
         dbname = getOption("aakmisc.dbname"),
         ...)
writeDBTable(name, value, overwrite = FALSE, append = FALSE,
             row.names = FALSE,
             host = getOption("aakmisc.dbhost"),
             dbname = getOption("aakmisc.dbname"), ...)
startTunnel(port, remotehost = getOption("aakmisc.remotehost"), sleep = 5)
stopTunnel(info)
```

**Arguments**

| | |
|---|---|
| host | Hostname on which to connect to the PostgreSQL server. |
| dbname | Name of PostgreSQL database. |
| mle | A data-frame of MLEs to be recorded. |
| files | Files containing R scripts to be recorded. |
| script | Name of script. |

| | |
|---|---|
| file | File to which the script will be written. See [cat](). |
| statement | SQL statement passed to [dbGetQuery](). |
| name, value | Name and contents of table to create. |
| overwrite, append, row.names | |
| | See [dbWriteTable](). |
| port | Local port to use for ssh tunnel. If unspecified, a random port number will be used. |
| remotehost | Hostname of PostgreSQL server. An ssh tunnel to this host will be created. |
| sleep | Time in seconds to sleep after initiating the ssh tunnel. |
| info | Information about the ssh tunnel returned by startTunnel. |
| ... | Additional arguments will be passed to [dbConnect](). |

## Author(s)

Aaron A. King <kingaa at umich dot edu>

## Examples

```
   ## Not run:
startTunnel() -> tun
listScripts(port=tun$port)
stopTunnel(tun)

## End(Not run)
```

---

| random | *Functions for generating truly random integers.* |
|---|---|

---

## Description

Functions for generating truly random integers.

## Usage

```
   random.org(n = 10, rnd = "new")
urandom(n = 10)
```

## Arguments

| | |
|---|---|
| n | Number of integers required. |
| rnd | random.org parameter |

## Author(s)

Aaron A. King <kingaa at umich dot edu>

## References

<http://www.random.org>

## Examples

```
## Not run:
random.org(n=5)

## End(Not run)
  seed <- urandom(n=1)
  set.seed(seed)
  runif(5)
```

# Index