# Package 'Andromeda'

April 17, 2020

**Type** Package

**Title** AsynchroNous Disk-based Representation of MassivE DAta

**Version** 0.0.1

**Date** 2020-04-10

**Maintainer** Martijn Schuemie <schuemie@ohdsi.org>

**Description** AsynchroNous Disk-based Representation of MassivE DAta (AN-
DROMEDA): An R package for storing large data objects. Andromeda allow storing data ob-
jects on a local drive, while still making it possible to manipulate the data in an efficient manner.

**License** Apache License 2.0

**VignetteBuilder** knitr

**URL** https://github.com/OHDSI/Andromeda

**BugReports** https://github.com/OHDSI/Andromeda/issues

**Depends** dplyr

**Imports** RSQLite,
DBI,
zip,
methods,
dbplyr,
cli,
pillar

**Suggests** testthat,
knitr,
rmarkdown,
rlang,
tibble

**LazyData** false

**RoxygenNote** 7.1.0

**Encoding** UTF-8

## R topics documented:

---

andromeda                    *Create an Andromeda object*

---

#### Description

By default the Andromeda object is created in the systems temporary file location. You can override this by specifying a folder using `options(andromedaTempFolder = "c:/andromedaTemp")`, where "c:/andromedaTemp" is the folder to create the Andromeda objects in.

#### Usage

```
andromeda(...)
```

#### Arguments

| | |
|---|---|
| ... | Named objects. See details for what objects are valid. If no objects are provided, an empty Andromeda is returned. |

#### Details

Valid objects are data frames, Andromeda tables, or any other dply table.

#### Examples

```
andr <- andromeda(cars = cars, iris = iris)

names(andr)
# [1] 'cars' 'iris'

andr$cars %>% filter(speed > 10) %>% collect()
# # A tibble: 41 x 2
# speed dist
# <dbl> <dbl>
# 1 11 17
# ...

close(andr)
```

---

appendToTable            *Append to an Andromeda table*

---

### Description

Append a data frame, Andromeda table, or result of a query on an Andromeda table to an existing Andromeda table. If data from another Andromeda is appended, a batch-wise copy process is used, which will be slower than when appending data from within the same Andromeda object.

### Usage

```
appendToTable(tbl, data)
```

### Arguments

tbl            An Andromeda table. This must be a base table (i.e. it cannot be a query result).

data           The data to append. This can be either a data.frame or another Andromeda table.

### Examples

```
andr <- andromeda(cars = cars)
nrow(andr$cars)
# [1] 50

appendToTable(andr$cars, cars)
nrow(andr$cars)
# [1] 100

appendToTable(andr$cars, andr$cars %>% filter(speed > 10))
nrow(andr$cars)
# [1] 182

close(andr)
```

---

batchApply            *Apply a function to batches of data in an Andromeda table*

---

### Description

Apply a function to batches of data in an Andromeda table

### Usage

```
batchApply(tbl, fun, ..., batchSize = 10000, safe = FALSE)
```

**Arguments**

| | |
|---|---|
| `tbl` | An Andromeda table (or any other DBI table). |
| `fun` | A function where the first argument is a data frame. |
| `...` | Additional parameters passed to fun. |
| `batchSize` | Number of rows to fetch at a time. |
| `safe` | Create a copy of tbl first? Allows writing to the same Andromeda as being read from. |

**Details**

This function is similar to the `lapply` function, in that it applies a function to sets of data. In this case, the data is batches of data from an Andromeda table. Each batch will be presented to the function as a data frame.

**Examples**

```
andr <- andromeda(cars = cars)

fun <- function(x) {
  return(nrow(x))
}

result <- batchApply(andr$cars, fun, batchSize = 25)

result
# [[1]]
# [1] 25
#
# [[2]]
# [1] 25

close(andr)
```

---

copyAndromeda                    *Copy Andromeda*

---

**Description**

Creates a complete copy of an Andromeda object. Object attributes are not copied.

**Usage**

```
copyAndromeda(andromeda)
```

**Arguments**

| | |
|---|---|
| `andromeda` | The andromeda object to copy. |

**Value**

The copied andromeda.

## Examples

```
andr <- andromeda(cars = cars, iris = iris)

andr2 <- copyAndromeda(andr)

names(andr2)
# [1] 'cars' 'iris'

close(andr)
close(andr2)
```

---

| groupApply | *Apply a function to groups of data in an Andromeda table* |

---

## Description

Apply a function to groups of data in an Andromeda table

## Usage

```
groupApply(tbl, groupVariable, fun, ..., batchSize = 10000, safe = FALSE)
```

## Arguments

| | |
|---|---|
| tbl | An Andromeda table (or any other DBI table). |
| groupVariable | The variable to group by |
| fun | A function where the first argument is a data frame. |
| ... | Additional parameters passed to fun. |
| batchSize | Number of rows fetched from the table at a time. This is not the number of rows to which the function will be applied. Included mostly for testing purposes. |
| safe | Create a copy of tbl first? Allows writing to the same Andromeda as being read from. |

## Details

This function applies a function to groups of data. The groups are identified by unique values of the groupVariable, which must be a variable in the table.

## Value

Invisibly returns a list of objects, where each object is the output of the user supplied function applied to a group.

## Examples

```
andr <- andromeda(cars = cars)

fun <- function(x) {
  return(tibble::tibble(speed = x$speed[1], meanDist = mean(x$dist)))
}

result <- groupApply(andr$cars, "speed", fun)
result <- bind_rows(result)
result
# # A tibble: 19 x 2
# speed meanDist
# <dbl> <dbl>
# 1 4 6
# 2 7 13
# 3 8 16
# ...

close(andr)
```

---

isAndomeda                    *Check whether an object is an Andromeda object*

---

## Description

Check whether an object is an Andromeda object

## Usage

```
isAndomeda(x)
```

## Arguments

x                    The object to check.

## Details

Checks whether an object is an Andromeda object.

## Value

A logical value.

| isValidAndromeda | *Check whether an Andromda object is still valid* |
|---|---|

### Description

Check whether an Andromda object is still valid

### Usage

```
isValidAndromeda(x)
```

### Arguments

x                    The Andromeda object to check.

### Details

Checks whether an Andromeda object is still valid, or whether it has been closed.

### Value

A logical value.

### Examples

```
andr <- andromeda(cars = cars, iris = iris)

isValidAndromeda(andr)
# TRUE

close(andr)

isValidAndromeda(andr)
# FALSE
```

| loadAndromeda | *Load Andromeda from file* |
|---|---|

### Description

Load Andromeda from file

### Usage

```
loadAndromeda(fileName)
```

### Arguments

fileName          The path where the object was saved using [saveAndromeda](saveAndromeda).

**See Also**

[saveAndromeda](saveAndromeda)

**Examples**

```
## Not run:
andr <- loadAndromeda("c:/temp/andromeda.zip")
names(andr)
# [1] 'cars'

close(andr)

## End(Not run)
```

---

names,Andromeda-method

*names*

---

**Description**

Show the names of the tables in an Andromeda object.

**Usage**

```
## S4 method for signature 'Andromeda'
names(x)
```

**Arguments**

x                    An Andromeda object.

**Examples**

```
andr <- andromeda(cars = cars, iris = iris)

names(andr)
# [1] 'cars' 'iris'

close(andr)
```

---

saveAndromeda                    *Save Andromeda to file*

---

### Description

Saves the Andromeda object in a zipped file. Note that by default the Andromeda object is automatically closed by saving it to disk. This is due to a limitation of the underlying technology (RSQLite). To keep the connection open, use maintainConnection = TRUE. This will first create a temporary copy of the Andromeda object. Note that this can be substantially slower.

### Usage

```
saveAndromeda(
  andromeda,
  fileName,
  maintainConnection = FALSE,
  overwrite = TRUE
)
```

### Arguments

| | |
|---|---|
| andromeda | An object of class Andromeda. |
| fileName | The path where the object will be written. |
| maintainConnection | |
| | Should the connection be maintained after saving? If FALSE, the Andromeda object will be invalid after this operation, but saving will be faster. |
| overwrite | If the file exists, should it be overwritten? If FALSE and the file exists, an error will be thrown. |

### See Also

[loadAndromeda](#)

### Examples

```
## Not run:
andr <- andromeda(cars = cars)
saveAndromeda(cars, "c:/temp/andromeda.zip")

## End(Not run)
```

# Index