Package 'rpostgisLT'

August 21, 2016

Title Managing and visualizing animal movement data with PostGIS and R

Version 0.2.1

Description The `rpostgisLT` package develops the integration of R and PostGIS for managing movement trajectories. The focus is on streamlining the workflow for biologists to store and process animal trajectories in PostGIS and analyze them in R, thus utilizing the strengths of both software. The package relies on `ltraj` objects from the R package `adehabitatLT`, and provides the analogous `pgtraj` data structure in PostGIS, with all functions to create and manage `pgtraj` data, and convert from and to both format (`pgtraj` in PostGIS, `ltraj` in R). For a list of documented functions, use `library(help = ``rpostgisLT'')`

SystemRequirements PostgreSQL with PostGIS extension

Depends R (>= 3.3.0),
rpostgis,
adehabitatLT

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

Author David Bucklin [aut, cre],
Mathieu Basille [aut],
Balázs Dukai [aut],
Clément Calenge [ctb]

Imports testthat

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

as_pgtraj *Imports location data from a database table into a 'traj' schema.*

Description

`as_pgtraj` populates a `traj` schema from the data provided in `relocations_table`. If the provided schema doesn't exist, it will be created. On successful data input, `as_pgtraj` creates three database views for each pgtraj. These views are named <pgtraj_name>_parameters, <pgtraj_name>_step_geometry and <pgtraj_name>_summary and described more in detail in the package vignette.

The time zone of the pgtraj is set to the local time zone.

Usage

```
as_pgtraj(conn, schema = "traj", relocations_table = NULL,
  pgtrajs = "pgtraj", animals = "animal", bursts = NULL,
  relocations = NULL, timestamps = NULL, rids = "rid", srid = NULL,
  note = NULL)
```

Arguments

`conn`	Connection object created with RPostgreSQL
`schema`	String. Name of the schema that stores or will store the pgtraj data model.
`relocations_table`	String. Name of the table that stores the relocations, e.g. "public.relocations"
`pgtrajs`	String. Name of the pgtraj or name of the field that stores the pgtraj names.
`animals`	String. Name of the animal or name of the field that stores the animal names.
`bursts`	String. Name of the burst or name of the field that stores the burst names.
`relocations`	String. Name of the field that contains the relocations in relocations_table. Relocations can be provided either as X,Y coordinates or PostGIS geometry. In both cases all relocations in the 'relocations_table' have to have the same projection.
`timestamps`	String. Name of the field in relocations_table that contains the timestamps. If NULL, Type I trajectory is assumed.
`rids`	String. Name of the field in relocations_table that contains the numeric IDs of relocations.
`note`	String. Comment on the pgtraj. The comment is only used in the database and not transferred into an ltraj.

Details

Opening and closing connections have to be done manually by the user. However, the function checks if the provided connection is still valid. Not tested with capital letters for PostgreSQL field names.

Value

TRUE on success

Author(s)

Balázs Dukai <balazs.dukai@gmail.com>

References

https://cran.r-project.org/web/packages/adehabitatLT/vignettes/adehabitatLT.pdf

See Also

Section on traj data model in the package vignette.

Examples

```
## Not run:
as_pgtraj(conn,
        schema = "traj_t4",
        relocations_table = "example_data.relocations_plus",
        pgtrajs = "id",
        animals = "animal",
        bursts = "burst",
        relocations = "geom",
        timestamp = "time",
        rid = "gid")

## End(Not run)

#' \dontrun{
as_pgtraj(conn,
        schema = "traj_t4",
        relocations_table = "example_data.relocations_plus",
        pgtrajs = "id",
        animals = "animal",
        bursts = "burst",
        relocations = c("x","y"),
        timestamp = "time",
        rid = "gid")
}
```

ltraj2pgtraj *Export an ltraj object from R into a traj database schema.*

Description

ltraj2pgtraj creates a new traj schema or uses an existing one and exports an ltraj to the database. The time zone and projection information stored in the ltraj is transferred to the database. Uses as_pgtraj to insert the values into the traj schema.

Usage

```
ltraj2pgtraj(conn, ltraj, schema = "traj", pgtraj = NULL, note = NULL,
  overwrite = FALSE)
```

Arguments

conn Connection object created with RPostgreSQL

ltraj An ltraj object.

schema String. Name of the schema that stores or will store the pgtraj data model.

pgtraj String. Name of the new pgtraj. Defaults to the name of the object that stores the ltraj.

note String. A comment that will be stored with the pgtraj in the database.

overwrite Boolean. If TRUE, and a pgtraj with the same name as the provided ltraj exists in the database, the pgtraj is deleted and the provided ltraj is inserted. If FALSE, the function exits. Note that overwrite requires an exact match among the pgtraj names otherwise it is ignored.

create Logical. If no matching SRID is found, should a new SRID be created? User must have write access on spatial_ref_sys table.

new.srid Integer. Optional SRID to give to a newly created SRID. If left NULL (default), the next open value of ‘srid‘ in ‘spatial_ref_sys‘ between 880000 and 890000 will be used.

Value

TRUE on success

Author(s)

Balázs Dukai <balazs.dukai@gmail.com>

See Also

as_pgtraj

Examples

```
## Not run: ltraj2pgtraj(conn, ibex, "traj_t2")
```

`pgtraj2ltraj` *Import a pgtraj into an ltraj.*

Description

`pgtraj2ltraj` imports a single pgtraj from a database into an ltraj object.

Usage

```
pgtraj2ltraj(conn, schema = "traj", pgtraj)
```

Arguments

`conn`	Connection object created with RPostgreSQL
`schema`	String. Name of the schema that stores or will store the pgtraj data model.
`pgtraj`	String. Name of the pgtraj.

Value

an ltraj object

Author(s)

Balázs Dukai <`balazs.dukai@gmail.com`>

Examples

```
## Not run: pgtraj2ltraj(conn, "traj_t2", "ibex")
```

`pgTrajDrop` *Delete one or more pgtrajs from a traj schema.*

Description

`pgTrajDrop` deletes one or more pgtrajs from a traj schema.

Usage

```
pgTrajDrop(conn, schema = "traj", pgtraj)
```

Arguments

`conn`	Connection object created with RPostgreSQL
`schema`	String. Name of the schema that stores the traj data model.
`pgtraj`	String. A vector containing the names of the pgtrajs.

Value

TRUE on success

Author(s)

Balázs Dukai <`balazs.dukai@gmail.com`>

Examples

```
## Not run: pgTrajDrop(conn, "traj", "ibex")

## Not run: pgTrajDrop(conn, schema="traj", pgtraj=c("ibex", "puechcirc")
```

`pgTrajSchema` *Check pgtraj schema.*

Description

Checks if the provided schema is a valid pgtraj schema, and creates one if it does not exist.

Usage

```
pgTrajSchema(conn, name = "traj")
```

Arguments

`conn` Connection object created with RPostgreSQL.

`name` Character string. Name of the schema that stores or will store the pgtraj data model.

Details

Creates a schema to store `pgtraj`s in the database by calling a SQL script from `./sql/traj_schema.sql`. The schema name defaults to `traj`. If a schema with the provided name already exists in the database, it checks if it contains all the required tables. The function does not attempt to repair the schema if all pgtraj tables are not present (e.g. because some were manually deleted). In this case, a new pgtraj schema needs to be created, or the existing schema needs to be deleted and recreated.

The function has it's standalone transaction control.

Value

`TRUE` if the schema exists (whether it was already available or was successfully created).

Author(s)

Balázs Dukai <`balazs.dukai@gmail.com`>

Examples

```
## Not run:
pgTrajSchema(conn, name = "traj_1")

## End(Not run)
```

`pgTrajVacuum` *pgTrajVacuum Performs a VACUUM (garbage-collect and optionally analyze) on all the tables of a `traj` schema.*

Description

pgTrajVacuum Performs a VACUUM (garbage-collect and optionally analyze) on all the tables of a `traj` schema.

Usage

```
pgTrajVacuum(conn, schema, full = FALSE, verbose = FALSE, analyze = TRUE)
```

Arguments

conn	Connection object created with RPostgreSQL
schema	String. Name of the schema that stores or will store the pgtraj data model.
full	Logical. Whether to perform a "full" vacuum, which can reclaim more space, but takes much longer and exclusively locks the table.
verbose	Logical. Whether to print a detailed vacuum activity report for each table.
analyze	Logical. Whether to update statistics used by the planner to determine the most efficient way to execute a query (default to `TRUE`).

Value

`TRUE`

Author(s)

Balázs Dukai <balazs.dukai@gmail.com>

Examples

```
## Not run: pgTrajVacuum(conn, "traj_1")
```

Index