

Package ‘rpostgisLT’

July 28, 2016

Title Managing and visualizing animal movement data with PostGIS and R

Version 0.1.1

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

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

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

Imports testthat

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

as_pgtraj 2

ltraj2pgtraj	3
pgtraj2ltraj	4
pgTrajSchema	5

Index	6
--------------	----------

as_pgtraj	<i>Converts relocation data into the pgtraj data model.</i>
-----------	---

Description

This is the core function of the rpostgisLT package and it is also used by [ltraj2pgtraj](#) to import trajectory data into a pgtraj data model. as_pgtraj copies the trajectory data which is stored in a database to a traj schema. If the provided schema doesn't exist, it is created on demand. On successful data input, as_pgtraj creates a view for each pgtraj, with the views named as <pgtraj>_params. The view contains the same step parameters as an ltraj object (e.g. R2n, rel.angle, dt...). If the input geometries are projected, their projection is used to create the steps in the schema, otherwise either no projection is used or the fuction exits.

Usage

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

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.
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.
db	Boolean. If TRUE, the relocations are stored in a database table, if FALSE relocations are stored in an R object. It is meant to be used by other functions internally. If you want to import an ltraj from R, use ltraj2pgtraj().

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, but it probably won't work. Its a bad practice anyway to force uppercase in PostgreSQL so use lowercase.

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 pgtraj 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)
```

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. Uses as_pgtraj to insert the values into the traj schema.

Usage

```
ltraj2pgtraj(conn, ltraj, schema = "traj", pgtraj = NULL, comment = NULL)
```

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 variable that stores the ltraj.
comment	String. A comment that will be stored with the pgtraj in the database.
epsg	Numeric. The EPSG code of the Coordinate Reference System of the relocation coordinates in the ltraj. Defaults to 0.

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	<i>Import a pgtraj into an ltraj.</i>
--------------	---------------------------------------

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")
```

pgTrajSchema

Check 'traj' schema.

Description

Checks if the provided schema exists in the database, and creates if it doesn't.

Usage

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

Arguments

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

Details

Creates a 'traj' schema in the database by calling a SQL script from (./inst/sql/). The schema name defaults to 'traj'.

Value

TRUE on success

Author(s)

Balázs Dukai <balazs.dukai@gmail.com>

Examples

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

Index

as_pgtraj, [2](#), [4](#)

ltraj2pgtraj, [2](#), [3](#)

pgtraj2ltraj, [4](#)

pgTrajSchema, [5](#)