

# Package ‘rpostgisLT’

September 6, 2016

**Title** Managing Animal Movement Data with 'PostGIS' and R

**Version** 0.4.0

**Date** 2016-09-06

**Description** The 'rpostgisLT' package develops the integration of R and 'PostGIS' for managing movement trajectories. The focus is on streamlining the work flow 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 formats ('pgtraj' in 'PostGIS', 'ltraj' in R).

**SystemRequirements** PostgreSQL with PostGIS extension

**Depends** R (>= 3.3.0),  
DBI,  
RPostgreSQL,  
rpostgis,  
adehabitatLT (>= 0.3.12)

**License** GPL (>= 3)

**URL** <https://github.com/mablab/rpostgisLT>

**BugReports** <https://github.com/mablab/rpostgisLT/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 5.0.1

**Imports** sp,  
testthat

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

## R topics documented:

as_pgtraj . . . . .	2
ltraj2pgtraj . . . . .	4

pgtraj2ltraj . . . . .	5
pgTrajDrop . . . . .	5
pgTrajSchema . . . . .	6
pgTrajVacuum . . . . .	7
rpostgisLT . . . . .	7
<b>Index</b>	<b>9</b>

---

as_pgtraj	<i>Imports location data from a database table into a 'traj' schema.</i>
-----------	--

---

## 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 of the user.

## Usage

```
as_pgtraj(conn, schema = "traj", relocations_table, pgtrajs = "pgtraj",
  animals = "animal", bursts = NULL, relocations, 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 schema and table that stores the relocations, e.g. c("schema","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.
srid	Integer. Optional SRID (spatial reference ID) of (x,y) coordinates provided for relocations. Ignored if relocations is a geometry type.
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 = c("example_data", "relocations_plus"),
  pgtrajs = "id",
  animals = "animal",
  bursts = "burst",
  relocations = "geom",
  timestamp = "time",
  rid = "gid")

## End(Not run)

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

## End(Not run)
```

---

ltraj2pgtraj

Export ltraj object from R into database.

---

### 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.

### Usage

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

### Arguments

conn	A connection object.
ltraj	An object of class ltraj.
schema	Character. Name of the schema that stores or will store the pgtraj data model.
pgtraj	Character. Name of the new pgtraj. Defaults to the name of the provided ltraj.
note	Character. A note that will be stored with the pgtraj in the database.
overwrite	Logical. Use if a pgtraj with the same name as the provided ltraj already exists in the database: If TRUE, the existing 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.

### Value

TRUE on success.

### Author(s)

Balázs Dukai <balazs.dukai@gmail.com>

### See Also

[as\\_pgtraj](#) to create a pgtraj with data already stored in the database.

### Examples

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

## End(Not run)
```

---

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")
```

---

pgTrajDrop	<i>Delete one or more pgtrajs from a traj schema.</i>
------------	---

---

**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")
pgTrajDrop(conn, schema="traj", pgtraj=c("ibex", "puehcirc"))

## End(Not run)
```

---

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 pgtrajs 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 its 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 VACUUM on a pgtraj schema.***Description**

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 on success.

**Author(s)**

Balázs Dukai <balazs.dukai@gmail.com>

**Examples**

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

## End(Not run)
```

rpostgisLT

*Integration of ltraj (adehabitatLT) and pgtraj (PostGIS).***Description**

rpostgisLT

**Details**

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")`

**Author(s)**

Balázs Dukai <balazs.dukai@gmail.com>



# Index

as\_pgtraj, [2](#), [4](#)

ltraj2pgtraj, [4](#)

pgtraj2ltraj, [5](#)

pgTrajDrop, [5](#)

pgTrajSchema, [6](#)

pgTrajVacuum, [7](#)

rpostgisLT, [7](#)

rpostgisLT-package (rpostgisLT), [7](#)