

Package ‘spatpersist’

September 5, 2026

Title Create Persistent Identifiers for Longitudinal Spatial Data

Version 0.1.0

Description Creates dataset-local persistent identifiers for polygon units observed across time. Configurable overlap metrics, thresholds, and one-to-one matching rules identify continuity while separate identifiers track boundary versions and broader lineages through splits, mergers, and replacements. Candidate-link diagnostics, validation checks, registry reconciliation, and lineage summaries keep the resulting decisions inspectable and reproducible.

License MIT + file LICENSE

URL <https://github.com/emre-cebeci/spatpersist>

BugReports <https://github.com/emre-cebeci/spatpersist/issues>

Encoding UTF-8

Imports sf

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Emre Cebeci [aut, cre]

Maintainer Emre Cebeci <cebeciemre1@gmail.com>

Repository CRAN

Date/Publication 2026-09-05 13:50:02 UTC

Contents

example_units	2
id_lineages	2
id_transitions	3
persist_ids	3
plot_lineage	5
validate_ids	6

Index**8**

example_units	Create an example longitudinal spatial dataset
---------------	--

Description

Generates synthetic polygon data showing common spatial continuity problems: stable units, boundary changes, renames, and splits.

Usage

```
example_units()
```

Value

An sf object containing synthetic polygons.

Examples

```
data <- example_units()
plot(data["name"])
```

id_lineages	Summarize persistent spatial identities and lineages
-------------	--

Description

Creates one row per spatial_id with its lineage, parent, observed time span, number of observations and geometry versions, registry recovery count, ambiguity count, and minimum non-missing match confidence.

Usage

```
id_lineages(data, time)
```

Arguments

data	A result returned by persist_ids() .
time	A single character string naming the time column in data.

Value

A data frame with one row per persistent spatial identity.

Examples

```
result <- persist_ids(example_units(), time = "year")
id_lineages(result, time = "year")
```

id_transitions	<i>Inspect spatial transition diagnostics</i>
----------------	---

Description

Retrieves the candidate-link diagnostics created by `persist_ids()`. The table contains every positive-area overlap considered across successive observed periods, including its overlap statistics, mutual-best and ambiguity flags, candidate status, selection, and confidence.

Usage

```
id_transitions(x)
```

Arguments

`x` An sf result returned by `persist_ids()`.

Value

A data frame of transition diagnostics.

Examples

```
result <- persist_ids(example_units(), time = "year")
id_transitions(result)
```

persist_ids	<i>Create persistent identifiers for longitudinal polygons</i>
-------------	--

Description

Assigns persistent identifiers by comparing polygons in each pair of successive observed time periods. A polygon continues an earlier identity when its selected overlap statistic meets threshold. Candidate links are selected one-to-one, so a persistent identifier cannot occur more than once in the same time period.

Usage

```
persist_ids(
  data,
  time,
  threshold = 0.75,
  metric = c("share_old", "share_new", "iou"),
  match_rule = c("greedy", "mutual_best"),
  ambiguity_tolerance = 0.05,
```

```

ambiguity_action = c("flag", "new", "error"),
event_threshold = 0.1,
lineage_threshold = 0.6,
version_threshold = 1,
geometry_action = c("error", "repair"),
geometry_precision = NULL,
max_time_gap = Inf,
registry = NULL,
registry_threshold = 0.999999
)

```

Arguments

<code>data</code>	An sf object containing polygon geometries observed over time.
<code>time</code>	A single character string naming the time column in data.
<code>threshold</code>	A number from 0 to 1 giving the minimum overlap required for continuity.
<code>metric</code>	The overlap statistic used for matching: "share_old", "share_new", or "iou".
<code>match_rule</code>	Candidate selection rule: "greedy" considers every eligible link, while "mutual_best" considers only links that are best for both their predecessor and successor.
<code>ambiguity_tolerance</code>	Maximum difference between eligible top scores that is treated as a near tie.
<code>ambiguity_action</code>	How near ties should be handled: "flag" selects deterministically and marks the result, "new" rejects ambiguous links, and "error" stops before assigning IDs.
<code>event_threshold</code>	A number from 0 to 1 giving the minimum share of a predecessor or successor used when detecting splits and mergers.
<code>lineage_threshold</code>	A number from 0 to 1 giving the minimum share of the smaller polygon that must overlap to connect two observations in the same lineage. Accepted identity matches are always connected.
<code>version_threshold</code>	A number from 0 to 1 giving the minimum IoU needed for a continuing identity to retain its current geometry version. The default of 1 creates a new version for any boundary change.
<code>geometry_action</code>	How invalid polygon geometries should be handled: "error" or "repair". Repair uses <code>sf::st_make_valid()</code> and returns the repaired geometries.
<code>geometry_precision</code>	Optional positive precision scale passed to <code>sf::st_set_precision()</code> before spatial operations. Coordinates are rounded as <code>round(x * geometry_precision) / geometry_precision</code> ; for example, use 100 for two decimal places. NULL preserves input precision.
<code>max_time_gap</code>	Maximum gap across which identities may continue. Defaults to Inf. Finite values require a numeric, Date, or POSIXt time column; Date gaps use days and POSIXt gaps use seconds.

registry Optional prior sf result from `persist_ids()`. Same-time geometries are used to preserve previously issued spatial and geometry-version identifiers across re-runs. Existing lineage identifiers are also retained unless new spatial evidence connects multiple registry lineages; the connected component then uses the lexicographically smallest existing lineage identifier.

registry_threshold Minimum same-time IoU required to recover an observation from registry.

Details

Names and other non-spatial attributes are not used for matching. Unmatched polygons receive new identifiers. Strong spatial links connect related identities in a shared lineage and are used to describe splits, mergers, replacements, and complex reorganizations.

Candidate-link diagnostics are stored with the result and can be retrieved with `id_transitions()`.

Identifiers are dataset-local. Values such as SID000001 and LID000001 are persistent only within one dataset and its registry chain; they are not globally unique across independent projects. Exact same-period coextensive geometries are rejected because geometry-only matching cannot distinguish their identities reproducibly.

Value

The input sf object with identifier and transition columns, including `spatial_id`, `spatial_version_id`, `lineage_id`, `parent_id`, `transition_type`, `match_score`, `match_confidence`, and `match_ambiguous`. `registry_matched` indicates recovered observations.

Examples

```
data <- example_units()
result <- persist_ids(data, time = "year", threshold = 0.75)
result[, c("year", "name", "spatial_id")]
```

plot_lineage

Plot spatial identity lineages over time

Description

Draws observations on a time-by-identity layout. Solid links are selected identity continuations; dashed links are additional spatial lineage relationships such as split and merger branches.

Usage

```
plot_lineage(
  data,
  time,
  lineage_id = NULL,
  show_legend = TRUE,
```

```
node_cex = 1.2,  
...  
)
```

Arguments

- data A result returned by `persist_ids()`.
- time A single character string naming the time column in data.
- lineage_id Optional character vector of lineage IDs to display. NULL displays every lineage.
- show_legend Logical; display a transition-type legend.
- node_cex Numeric node size.
- ... Additional arguments passed to `graphics::plot.default()`.

Value

Invisibly, a list containing the plotted nodes and edges data.

Examples

```
result <- persist_ids(example_units(), time = "year")  
plot_lineage(result, time = "year", lineage_id = "LID000004")
```

validate_ids	<i>Validate persistent spatial identifiers</i>
--------------	--

Description

Checks a result for structural inconsistencies, including missing output values, duplicate identities within a time period, identities assigned to multiple lineages, invalid geometry-version sequences, invalid transition labels, invalid match scores, and broken parent or continuation links.

Usage

```
validate_ids(data, time)
```

Arguments

- data A data frame or sf object containing spatial ID output.
- time A single character string naming the time column in data.

Value

A data frame describing validation issues. A result with zero rows passed all implemented checks.

Examples

```
result <- persist_ids(example_units(), time = "year")  
validate_ids(result, time = "year")
```

Index

`example_units`, [2](#)

`graphics::plot.default()`, [6](#)

`id_lineages`, [2](#)

`id_transitions`, [3](#)

`id_transitions()`, [5](#)

`persist_ids`, [3](#)

`persist_ids()`, [2](#), [3](#), [5](#), [6](#)

`plot_lineage`, [5](#)

`sf::st_make_valid()`, [4](#)

`sf::st_set_precision()`, [4](#)

`validate_ids`, [6](#)