Package 'gibble'

May 8, 2026

Title Geometry Decomposition

Version 0.4.0

Description Build a map of path-based geometry, this is a simple description of the number of parts in an object and their basic structure. Translation and restructuring operations for planar shapes and other hierarchical types require a data model with a record of the underlying relationships between elements. The gibble() function creates a geometry map, a simple record of the underlying structure in path-based hierarchical types. There are methods for the planar shape types in the 'sf' and 'sp' packages and for types in the 'trip' and 'silicate' packages.

Depends R (>= 3.3.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports dplyr, methods, tibble, rlang

Suggests covr, testthat, knitr, rmarkdown

URL https://github.com/mdsumner/gibble

BugReports https://github.com/mdsumner/gibble/issues

VignetteBuilder knitr

NeedsCompilation no

Author Michael Sumner [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2471-7511>)

Maintainer Michael Sumner <mdsumner@gmail.com>

Repository CRAN

Date/Publication 2020-05-09 05:40:03 UTC

Contents

`gibble-package` *gibble package*

Description

Path-based geometry decomposition to data frame.

Details

The goal of gibble is to provide a straightforward map of path-based data structures. Paths are connected linear sequences of coordinates, otherwise known as *lines*. Structured spatial data includes *objects* composed of *paths*, which are islands, holes within island, lines, and points. Polygons require an extra grouping level *subobject* to ensure holes are nested within their island, but this is ignored for other types.

Sequential inear structures are not optimal for many operations when working with spatial data, and gibble provides a simple bridge between existing paths and tools to decompose them to primitives.

`gibble` *Path-based geometry decomposition*

Description

A gibble is a geometry map, a summary of the structure of each path within a simple planar shape. The generic function `gibble()` will summarize all paths within a complex type with a data frame row for each. Each row of the gibble summary represents a component element of the object's geometry and records the number of rows (i.e. how many coordinates) and the number of columns (i.e. the number of axes in the geometric space). Another column `type` records the class of the object, and `object` records an identifier for the entire object. A special case column `subobject` identifies the component POLYGON part within a MULTIPOLYGON. I.e. `subobject` is 1 unless that part is a hole. All identifiers are sequential within the higher level groupings.

Usage

```
## S3 method for class 'POINT'
gibble(x, ...)

## S3 method for class 'MULTIPOINT'
gibble(x, ...)

## S3 method for class 'LINESTRING'
```

```
gibble(x, ...)

## S3 method for class 'MULTILINESTRING'
gibble(x, ...)

## S3 method for class 'POLYGON'
gibble(x, ...)

## S3 method for class 'MULTIPOLYGON'
gibble(x, ...)

## S3 method for class 'list'
gibble(x, ...)

## S3 method for class 'sfc'
gibble(x, ...)

## S3 method for class 'sf'
gibble(x, ...)

## S3 method for class 'PATH0'
gibble(x, ...)

## S3 method for class 'PATH'
gibble(x, ...)

## S3 method for class 'trip'
gibble(x, ...)

## S3 method for class 'Polygon'
gibble(x, ...)

## S3 method for class 'Polygons'
gibble(x, ...)

## S3 method for class 'SpatialPolygons'
gibble(x, ...)

## S3 method for class 'Line'
gibble(x, ...)

## S3 method for class 'Lines'
gibble(x, ...)

## S3 method for class 'SpatialLines'
gibble(x, ...)

## S3 method for class 'SpatialMultiPoints'
```

```
gibble(x, ...)

## S3 method for class 'SpatialPoints'
gibble(x, ...)

## S3 method for class 'Spatial'
gibble(x, ...)

gibble(x, ...)

## Default S3 method:
gibble(x, ...)
```

Arguments

x	geometry model
...	arguments reserved for methods, none currently

Details

Methods are provided for the sf classes. There is an internal version that is designed to work fast on sets of structures, by delaying conversion to list or data frame for as late as possible.

A geometry map is not so helpful on its own, and so does not have a formal class. It is designed for use within other workflows such as updating the coordinates of model object or translating between superficially different formats.

Gibble is a distillation of the sc_path encoding of package silicate, which came from the map_table decompositions of spbabel, as an improvement on the single-table fortify model used in ggplot2.

Value

data frame summarizing the geometry map, see Details

Examples

```
gibble(minimal_mesh)
```

---

hsh_sf *hsh, home sweet home*

Description

A testing data set, a list of poly, line, points. The relationships between the objects cover many interesting cases. There are multi-island objects with holes and other objects that fall inside those holes. All paths share vertices with a neighbour. There is topological intersection along a diagonal line where two objects meet and are joined by shared vertices that were inserted deliberately.

Details

`hsh_sf` is the holey home in sf format.

`minimal_mesh` *minimal mesh*

Description

A testing data set, `minimal_mesh` composed of two MULTIPOLYGONs sharing one edge.

`mpoly` *mpoly*

Description

A testing data set, composed of two MULTIPOLYGONs, one is the manually constructed example from sf (and silicate::sfzoo$multipolygon), the second is South Australia, with many lake-islands in the main polygon, and one artifical hole in the Kangaroo Island polygon.

Index