Package 'SimplicialComplex'

October 20, 2025

Type Package

Title Topological Data Analysis: Simplicial Complex

Version 0.1.0

Maintainer ChiChien Wang <kennywang2003@gmail.com>

Description Provides an implementation of simplicial complexes for
Topological Data Analysis (TDA). The package includes functions to
compute faces, boundary operators, Betti numbers, Euler characteristic,
and to construct simplicial complexes. It also implements persistent
homology, from building filtrations to computing persistence diagrams,
with the aim of helping readers understand the core concepts of
computational topology.
Methods are based on standard references in persistent homology such as
Zomorodian and Carlsson (2005) <doi:10.1007/s00454-004-1146-y> and
Chazal and Michel (2021) <doi:10.3389/frai.2021.667963>.

Imports Matrix, gtools, igraph, ggplot2

License MIT + file LICENSE

URL https://github.com/TDA-R/SimplicialComplex

BugReports https://github.com/TDA-R/SimplicialComplex/issues

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author ChiChien Wang [aut, cre, trl]

Repository CRAN

Date/Publication 2025-10-20 19:30:15 UTC

Contents

---

abstract_simplicial_complex

Compute Euler characteristic for an abstract simplicial complex

---

Description

Compute Euler characteristic for an abstract simplicial complex

Usage

```
abstract_simplicial_complex(simplices, dimension, tol = NULL)
```

Arguments

simplices	A list of simplices (each a numeric vector).
dimension	Optional max dimension to compute up to.
tol	Optional numerical tolerance to pass to rankMatrix().

Value

The Euler characteristic χ.

Examples

```
simplices <- list(c(1, 2), c(3, 4), c(2, 1, 3), c(4, 2))
abstract_simplicial_complex(simplices, 2)
```

---

boundary *Compute the boundary operator for a simplicial complex*

Description

Compute the boundary operator for a simplicial complex

Usage

```
boundary(simplices, bound_dim)
```

Arguments

simplices	A list of simplices (each a numeric vector).
bound_dim	The dimension k of the boundary operator ∂_k.

Details

$$\partial_k \sigma = \sum_i (-1)^i [v_0 v_1 \ldots \hat{v}_i \ldots v_k]$$

Value

A sparse matrix representing ∂_k.

Examples

```
simplices <- list(c(1, 2), c(3, 4), c(2, 1, 3), c(4, 2))
boundary(simplices, 0)
```

---

boundary_info *Get the boundary matrix and its reduction information in matrix form*

Description

Get the boundary matrix and its reduction information in matrix form

Usage

```
boundary_info(filist)
```

Arguments

filist	Filtration list, each element includes simplex and time.

Value

A list containing the boundary matrix, the last boundary row, and the pivot owner for persistence extraction.

Examples

```
points <- matrix(c(0, 1, 1, 0, 0, 0, 1, 1), ncol = 2)
filtration <- build_vr_filtration(points, eps_max=1.2)
res <- boundary_info(filtration)
```

---

build_vr_filtration	*Vietoris-Rips Filtration: Get the boundary matrix and its reduction information in matrix form*

Description

Vietoris-Rips Filtration: Get the boundary matrix and its reduction information in matrix form

Usage

```
build_vr_filtration(points, eps_max)
```

Arguments

points   Data point input.

eps_max   Maximum scale (epsilon).

Value

A list of simplices with their information.

Examples

```
points <- matrix(c(0, 1, 1, 0, 0, 0, 1, 1), ncol = 2)
filtration <- build_vr_filtration(points, eps_max=1.2)
```

---

euler_characteristic *Compute the Euler characteristic χ of a simplicial complex*

Description

Compute the Euler characteristic χ of a simplicial complex

Usage

```
euler_characteristic(simplices, tol)
```

Arguments

simplices   A list of simplices (each a numeric vector).

tol   Optional numerical tolerance to pass to rankMatrix().

Details

The Euler characteristic is computed as:

$$\chi = \sum_{k=0}^{k_{\max}} (-1)^k \beta_k$$

where β_k is the kth Betti number, and $k_{\max}$ is the highest dimension of any simplex in the complex. Interpretation of values:

- $\chi = 2$: Sphere-like surfaces
- $\chi = 1$: Disk-like spaces
- $\chi = 0$: Torus-like or circle-like spaces
- $\chi < 0$: Surfaces with multiple handles or genus

Value

An integer representing the Euler characteristic χ.

See Also

betti_number

Examples

```
simplices <- list(c(1, 2), c(3, 4), c(2, 1, 3), c(4, 2))
euler_characteristic(simplices, tol=0.1)
```

```
extract_persistence_pairs
```
> *This function extracts the persistence from combining the boundary matrix and its filtration*

Description

This function extracts the persistence from combining the boundary matrix and its filtration

Usage

```
extract_persistence_pairs(filist, last_1, pivot_owner)
```

Arguments

`filist`	Filtration list, each element includes simplex and time.
`last_1`	The last 1 row index for each column in boundary matrix.
`pivot_owner`	The column index owning the pivot row.

Value

A data frame with columns: dimension, birth, and death.

Examples

```
points <- matrix(c(0, 1, 1, 0, 0, 0, 1, 1), ncol = 2)
filtration <- build_vr_filtration(points, eps_max=1.2)
res <- boundary_info(filtration)
pairs <- extract_persistence_pairs(filtration, res$last_1, res$pivot_owner)
```

---

faces	*Generate all unique faces of a given dimension from simplices*

Description

Generate all unique faces of a given dimension from simplices

Usage

```
faces(simplices, target_dim)
```

Arguments

`simplices`	A list of simplices (each a numeric vector).
`target_dim`	The target dimension k for the faces (e.g., `0` for vertices, `1` for edges, etc.).

Details

The function generates all possible subsets (combinations) of each simplex, removes duplicates, and filters them to only include those of length `target_dim` + 1.

For example, a 2-simplex `c(1, 2, 3)` has three 1-dimensional faces (edges): `c(1,2)`, `c(1,3)`, and `c(2,3)`, and three 0-dimensional faces (vertices): 1, 2, and 3.

Value

A list of faces (each a numeric vector) of dimension `target_dim`.

Examples

```
simplices <- list(c(1, 2), c(3, 4), c(2, 1, 3), c(4, 2))
faces(simplices, target_dim=0)
```

---

plot_persistence	*Plot Persistence Diagram*

Description

Plot Persistence Diagram

Usage

```
plot_persistence(df)
```

Arguments

df   Dataframe from plot_persistence.

Value

A ggplot2 object representing the persistence diagram.

Examples

```
points <- matrix(c(0, 1, 1, 0, 0, 0, 1, 1), ncol = 2)
filtration <- build_vr_filtration(points, eps_max=1.2)
res <- boundary_info(filtration)
pairs <- extract_persistence_pairs(filtration, res$last_1, res$pivot_owner)
plot_persistence(pairs)
```

VietorisRipsComplex   *Construct a Vietoris–Rips Complex (1-skeleton + maximal simplices)*

Description

Construct a Vietoris–Rips Complex (1-skeleton + maximal simplices)

Usage

```
VietorisRipsComplex(points, epsilon)
```

Arguments

points   A numeric matrix or data.frame with one point per row (columns are coordi-
                nates).

epsilon   A positive numeric threshold; connect points with distance $< \epsilon$.

Details

The Vietoris–Rips complex at scale ϵ includes a simplex for every finite set of points with pairwise distances $< \epsilon$. This function constructs the 1-skeleton (edges only) and then uses maximal cliques in that graph as the maximal simplices.

Value

A list with:

network An igraph object representing the 1-skeleton.

simplices A list of integer vectors, each the vertex indices of a maximal simplex.

Examples

```
points <- matrix(c(0, 1, 1, 0, 0, 0, 1, 1), ncol = 2)
epsilon <- 1.5
vr_complex <- VietorisRipsComplex(points, epsilon)
```

Index