

Package: DiscreteMorseR (via r-universe)

July 17, 2026

Type Package

Title Discrete Morse Theory for 3D Meshes Derived from Point Clouds

Version 1.0.0

Date 2026-06-24

Description Ultra-fast computation of discrete Morse (Marston Morse) gradient vector fields and critical simplices (0-simplices: vertices, 1-simplices: edges, 2-simplices: faces) on 3D triangular meshes from point clouds. Provides C++ backend with parallel processing for large-scale topological analysis, including connected component analysis and visualization tools. Perfect for LiDAR data, computational topology, and geometric analysis applications. The implementation follows Forman (1998) <[doi:10.1007/PL00009307](https://doi.org/10.1007/PL00009307)> for discrete Morse theory, with extensions for 3D mesh processing.

Language en-US

License MIT + file LICENSE

URL <https://github.com/DijoG/DiscreteMorseR>

BugReports <https://github.com/DijoG/DiscreteMorseR/issues>

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.10), dplyr (>= 1.0.0), purrr (>= 0.3.0), stringr (>= 1.4.0), data.table (>= 1.12.0), gtools (>= 3.8.0), readr (>= 1.3.0), future (>= 1.18.0), furrr (>= 0.2.0)

Suggests clustermq (>= 0.8.0), ggplot2 (>= 3.3.0), patchwork (>= 1.1.0), testthat (>= 3.0.0), knitr (>= 1.30), rmarkdown (>= 2.0), tictoc

LinkingTo Rcpp, RcppArmadillo

SystemRequirements C++17

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Config/testthat/edition 3
Config/pak/sysreqs libicu-dev libx11-dev
Repository https://dijog.r-universe.dev
Date/Publication 2026-06-30 13:47:37 UTC
RemoteUrl https://github.com/dijog/discretemorser
RemoteRef HEAD
RemoteSha 719acb2d1a21f2854979e2b9c613bd131fc5f7a0

Contents

compute_lowerSTAR_parallel	2
compute_MORSE_complex	3
get_CCMESH	4
save_MORSE_2d	4
visualize_MORSE_2d	5
visualize_MORSE_2d_panel	6
Index	7

compute_lowerSTAR_parallel
<i>Compute lower star in parallel</i>

Description

Compute lower star in parallel

Usage

```
compute_lowerSTAR_parallel(  
  vertex,  
  edge,  
  face,  
  output_dir = NULL,  
  cores = NULL,  
  batch_size = NULL  
)
```

Arguments

vertex	Vertex data
edge	Edge data
face	Face data
output_dir	Output directory
cores	Number of cores (default: available cores-1)
batch_size	Number of vertices per batch

Value

Combined lower star results

compute_MORSE_complex *Compute Morse complex from mesh*

Description

Compute Morse complex from mesh

Usage

```
compute_MORSE_complex(
  mesh,
  output_dir = NULL,
  parallel = TRUE,
  cores = 4,
  batch_size = NULL
)
```

Arguments

mesh	Mesh object from MeshesOperations
output_dir	Optional directory to save results. If provided, writes: - vertices.txt, edges.txt, faces.txt: Mesh simplices - vector_field.txt: Gradient vector field pairs - critical_simplices.txt: Critical simplices - lowerSTAR.txt: Lower star filtration data
parallel	Whether to use parallel processing (default: TRUE)
cores	Number of cores for parallel processing (default: 4)
batch_size	Number of vertices per batch in parallel processing

Value

List with Morse vector field and critical simplices

Examples

```
# Create a tetrahedron mesh
vertices <- matrix(c(0,0,0, 1,0,0, 0,1,0, 0,0,1), ncol=3, byrow=TRUE)
colnames(vertices) <- c("X", "Y", "Z")
faces <- matrix(c(1,2,3, 1,2,4, 1,3,4, 2,3,4), ncol=3, byrow=TRUE)
colnames(faces) <- c("i1", "i2", "i3")

# Extract unique edges from faces
all_edges <- rbind(faces[,c(1,2)], faces[,c(1,3)], faces[,c(2,3)])
unique_edges <- unique(t(apply(all_edges, 1, sort)))
edges <- data.frame(i1 = unique_edges[,1], i2 = unique_edges[,2])
```

```
# Create mesh object
mesh <- list(vertices = vertices, faces = faces, edges = edges)
attr(mesh, "input_truth") <- 1:nrow(vertices)

# Compute Morse complex (sequential mode for CRAN checks)
result <- compute_MORSE_complex(mesh, parallel = FALSE)

# View results
print(paste("Critical simplices:", length(result$critical)))
print(paste("Gradient pairs:", length(result$vector_field)))
```

get_CCESH

Ultra-fast mesh preparation with connected components

Description

Ultra-fast mesh preparation with connected components

Usage

```
get_CCESH(alphahull, select_largest = TRUE)
```

Arguments

alphahull Alphahull generated by `ahull3D::ahull3d()`

select_largest If TRUE, returns only largest connected component (default: TRUE) If FALSE, returns list of all connected components

Value

Single mesh (if `select_largest=TRUE`) or list of meshes (if `select_largest=FALSE`)

save_MORSE_2d

Save 2D visualization to file

Description

Save 2D visualization to file

Usage

```

save_MORSE_2d(
  morse_complex,
  filename,
  width = 10,
  height = 8,
  dpi = 300,
  panel_2d = FALSE,
  ...
)

```

Arguments

morse_complex	Output from compute_MORSE_complex()
filename	Output file name
width	Plot width in inches
height	Plot height in inches
dpi	Resolution
panel_2d	If TRUE, saves multi-panel plot. If FALSE, saves single projection plot.
...	Additional arguments to visualize_MORSE_2d() or visualize_MORSE_2d_panel()

visualize_MORSE_2d	<i>Fast Morse complex visualization using get_simplexCENTER()</i>
--------------------	---

Description

Fast Morse complex visualization using get_simplexCENTER()

Usage

```

visualize_MORSE_2d(
  morse_complex,
  projection = "XZ",
  point_alpha = 0.6,
  point_size = 1,
  max_points = 30000,
  plot_gradient = TRUE,
  plot_critical = TRUE
)

```

Arguments

morse_complex	Output from compute_MORSE_complex()
projection	Projection plane: "XY", "XZ", or "YZ" (default: "XZ")
point_alpha	Point transparency (default: 0.6)
point_size	Point size (default: 1)
max_points	Maximum points to plot per category (default: 30000)
plot_gradient	Whether to plot gradient arrows (default: TRUE)
plot_critical	Whether to plot critical points (default: TRUE)

Value

ggplot2 object

visualize_MORSE_2d_panel

Create multiple 2D projection plots

Description

Create multiple 2D projection plots

Usage

```
visualize_MORSE_2d_panel(
  morse_complex,
  point_alpha = 0.6,
  point_size = 1,
  max_points = 30000,
  plot_gradient = TRUE,
  plot_critical = TRUE
)
```

Arguments

morse_complex	Output from compute_MORSE_complex()
point_alpha	Point transparency (default: 0.6)
point_size	Point size (default: 1)
max_points	Maximum points to plot per category (default: 30000)
plot_gradient	Whether to plot gradient arrows (default: TRUE)
plot_critical	Whether to plot critical points (default: TRUE)

Value

List of ggplot2 objects

Index

`compute_lowerSTAR_parallel`, [2](#)

`compute_MORSE_complex`, [3](#)

`get_CCMESH`, [4](#)

`save_MORSE_2d`, [4](#)

`visualize_MORSE_2d`, [5](#)

`visualize_MORSE_2d_panel`, [6](#)