

Package ‘fibermargin’

September 18, 2026

Type Package

Title Categorical Mask and Spatial Label Refinement

Version 0.1.0

Description Implements 'FiberMargin', a deterministic training-free operator for repairing categorical masks and spatial labels from coordinates and labels alone. Its primary multiclass operator uses rotated space-filling-curve charts and two-sided class enclosure at one fixed geometric transport range. A class-balanced, isolation-protected chart-disagreement rule provides pointwise repair decisions and audit scores. An auxiliary nearest-neighbour ballot handles binary masks. The 'C++' engine supports two- and three-dimensional coordinates, removes constant axes independently within each specimen, and reuses one deterministic CPU worker budget without nested process pools. Reproducible mask corruptions, planar and volumetric simulators, damage-aware evaluation, and compact licensed human dorsolateral prefrontal cortex and colorectal cancer benchmarks support assessment.

License MIT + file LICENSE

URL <https://tkcaccia.github.io/fibermargin/>,
<https://github.com/tkcaccia/fibermargin>

BugReports <https://github.com/tkcaccia/fibermargin/issues>

Encoding UTF-8

Depends R (>= 4.1.0)

Imports Rcpp, parallel

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), ggplot2, dplyr, tidyr, knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Stefano Cacciatore [aut, cre, cph]

Maintainer Stefano Cacciatore <tkcaccia@gmail.com>
Repository CRAN
Date/Publication 2026-09-18 11:50:18 UTC

Contents

available_spatial_benchmarks	2
benchmark_spatial_refiners	3
clean_categorical_mask	4
colorectal_benchmark	4
corrupt_categorical_mask	5
dlpfc_benchmark	6
evaluate_mask_cleaning	7
evaluate_spatial_refinement	7
load_spatial_benchmark	9
refine_spatial_labels	10
simulate_complex_spatial_domains	11
simulate_gradient_regions	12
simulate_spatial_clusters	13
simulate_spatial_domains	14
simulate_volumetric_domains	15
spatial_benchmark	16
Index	18

available_spatial_benchmarks
<i>List real spatial benchmark datasets</i>

Description

Reports which real datasets can be redistributed with the package and records the status of the other biological studies used by the publication scripts.

Usage

available_spatial_benchmarks()

Value

A data frame with source, license, inclusion, and benchmark details.

Examples

available_spatial_benchmarks()

benchmark_spatial_refiners

Benchmark spatial label refinement methods

Description

Runs one or more refinement functions on identical benchmark inputs, records elapsed time, and evaluates every output with [evaluate_spatial_refinement()]. The same 'xy', 'labels', and optional masks are used for each method so results are directly comparable.

Usage

```
benchmark_spatial_refiners(
  data,
  methods = list(FiberMargin = refine_spatial_labels),
  include_initial = TRUE,
  seed = 1L,
  on_error = c("stop", "record")
)
```

Arguments

data	A benchmark object returned by a fibermargin simulator or [spatial_benchmark()], or a named list of such objects.
methods	Named list of refinement functions. A function must accept 'xy' and 'labels'; 'samples' is supplied when the function declares it or accepts '...'.
include_initial	Include the unrefined assignment as method 'Initial'.
seed	Integer seed reset before each method run.
on_error	Either "stop" or "record". The latter returns an 'error' column and 'NA' performance values for a failed method rather than aborting.

Value

A data frame with one row per dataset and method.

Examples

```
sim <- simulate_spatial_domains(n = 500, pattern = "jagged_stripes", seed = 4)
methods <- list(identity = function(xy, labels) labels)
benchmark_spatial_refiners(sim, methods)
```

`clean_categorical_mask`

Clean a noisy categorical mask

Description

Applies FiberMargin to a 2D pixel mask or 3D voxel mask using only coordinates and categorical assignments.

Usage

```
clean_categorical_mask(mask, samples = NULL, workers = NULL)
```

Arguments

<code>mask</code>	A categorical matrix or three-dimensional array. Missing cells are treated as void locations.
<code>samples</code>	Optional matrix or array with the same dimensions as <code>mask</code> . Different samples are cleaned independently.
<code>workers</code>	Optional CPU budget used across samples and independent spatial charts. NULL uses up to two physical cores when possible.

Value

A cleaned mask with the representation of `mask`. Pointwise attributes are `candidate`, `margin_score`, `required`, `repair_margin`, `atlas_dispersion`, `isolation`, and `changed`. `margin_score` is a local support contrast on the mask lattice; `repair_margin` is the gap to the local acceptance threshold.

Examples

```
mask <- matrix(c(rep("A", 50), rep("B", 50)), nrow = 10)
mask[5, 5] <- "B"
cleaned <- clean_categorical_mask(mask, workers = 1)
```

`colorectal_benchmark` *Human CRC Visium HD refinement benchmark*

Description

A compact derivative of the 10x Genomics human colorectal cancer (CRC) Visium HD dataset containing 194,541 annotated locations, two-dimensional coordinates, 19 author-derived WSI region labels, and 60 deterministic corruption recipes. Expression counts and tissue imagery are excluded. The derivative records all transformations and is redistributed under CC BY 4.0.

Usage

```
data(colorectal_benchmark)
```

Format

A list with coordinates, integer reference labels, adjacent-region labels, boundary and sparse indicators, corruption design, and attribution metadata.

Source

The **10x Genomics human CRC Visium HD dataset**, analyzed with Space Ranger 4.0.1.

References

10x Genomics (2025). Visium HD Spatial Gene Expression Library, Human Colorectal Cancer (FFPE). Published July 3, 2025. CC BY 4.0.

```
corrupt_categorical_mask
```

Generate a corrupted categorical mask

Description

Creates independent, boundary-concentrated, or coherent-patch errors for mask-cleaning experiments.

Usage

```
corrupt_categorical_mask(
  mask,
  mechanism = c("impulse", "boundary", "patch"),
  rate = 0.15,
  seed = 1L
)
```

Arguments

mask	A reference categorical matrix or three-dimensional array.
mechanism	One of "impulse", "boundary", or "patch".
rate	Fraction of non-missing cells whose labels are replaced.
seed	One integer random seed.

Value

A mask with the same representation as mask and a reproducible set of changed labels. Exactly $\text{round}(\text{rate} * n)$ non-missing cells are changed while at least one correct exemplar of every reference class is retained; a rate that makes this impossible is rejected.

dlpfc_benchmark

Human DLPFC Visium refinement benchmark

Description

A compact redistribution of spatial coordinates and manually annotated cortical layers for 47,329 spots from 12 sections and three donors in the Maynard et al. human DLPFC study. It also contains 45 frozen, anatomically constrained label-corruption scenarios. Expression counts and tissue images are deliberately excluded.

Usage

```
data(dlpfc_benchmark)
```

Format

A list with the following elements:

xy A 47,329 by 2 coordinate matrix.

truth Integer reference-layer codes.

initial A 47,329 by 45 matrix of corrupted label codes.

levels Seven cortical-layer names.

samples Section identifiers.

subject Donor identifiers.

boundary Logical boundary indicator.

sparse Logical sparse-layer indicator.

design Scenario corruption design.

metadata Source, license, and citation information.

Source

The [spatialLIBD Bioconductor data package](#).

References

Maynard KR et al. (2021). Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature Neuroscience* 24, 425-436.

Pardo B et al. (2022). spatialLIBD: an R/Bioconductor package to visualize spatially-resolved transcriptomics data. *BMC Genomics* 23, 434.

`evaluate_mask_cleaning`*Evaluate categorical mask cleaning*

Description

Evaluates a cleaned mask against a fixed reference and noisy input mask. The output includes overlap scores, class-boundary agreement, and correction/damage decomposition.

Usage

```
evaluate_mask_cleaning(  
  reference,  
  initial,  
  cleaned,  
  elapsed = NA_real_,  
  method = NULL  
)
```

Arguments

<code>reference</code>	Reference categorical matrix or three-dimensional array.
<code>initial</code>	Imperfect input mask.
<code>cleaned</code>	Output mask to evaluate.
<code>elapsed</code>	Optional elapsed runtime in seconds.
<code>method</code>	Optional method name included in the returned row.

Value

A one-row data frame containing class-aware overlap, boundary agreement, rare-class, correction, and damage measures.

`evaluate_spatial_refinement`*Evaluate a spatial label refinement*

Description

Computes primary and failure-mode metrics for label repair. Returned measures include recovery (accuracy and adjusted Rand index), class imbalance robustness (macro and worst recall), boundary/sparse reliability, and damage.

Usage

```
evaluate_spatial_refinement(
  truth,
  initial,
  refined,
  boundary = NULL,
  regions = NULL,
  sparse = NULL,
  elapsed = NA_real_,
  method = NULL
)
```

Arguments

<code>truth</code>	Reference assignments.
<code>initial</code>	Initial noisy assignments.
<code>refined</code>	Assignments returned by a refinement method.
<code>boundary</code>	Optional logical vector marking boundary observations.
<code>regions</code>	Optional region identifier. The least and most frequent regions define sparse- and dense-region accuracy when ‘sparse’ is omitted.
<code>sparse</code>	Optional logical vector marking sparse-region observations.
<code>elapsed</code>	Optional elapsed runtime in seconds.
<code>method</code>	Optional method name included in the returned row.

Details

‘correction_recall’ is the fraction of initially incorrect labels that are repaired. ‘damage_rate’ is the fraction of initially correct labels that are flipped to an incorrect label. ‘changed_precision’ is the fraction of modified sites that are correct after refinement. Empty strata are reported as ‘NA’ rather than imputed as zero. Adjusted Rand index is calculated directly from the vectors and does not require an external package.

Value

A one-row data frame containing the evaluation measures.

Examples

```
truth <- factor(c("A", "A", "B", "B"))
initial <- factor(c("A", "B", "B", "A"))
refined <- truth
evaluate_spatial_refinement(truth, initial, refined)
```

load_spatial_benchmark

Load a bundled real spatial benchmark

Description

Loads one stored corruption scenario from the human dorsolateral prefrontal cortex (DLPFC) Visium benchmark or generates one colorectal cancer (CRC) Visium HD corruption from its stored recipe. The package contains coordinates and labels, not expression counts or histology images.

Usage

```
load_spatial_benchmark(
  name = c("dlpfc", "merfish", "crc", "colorectal"),
  scenario = 1L,
  seed = NULL
)
```

Arguments

name	Dataset name. "dlpfc" and "crc" are bundled. The legacy "colorectal" alias is accepted for compatibility.
scenario	Scenario number or scenario identifier. Inspect the 'design' element of the corresponding data object for the complete design.
seed	Optional integer seed used when generating a CRC corruption. The default 'NULL' uses the current R random-number stream without setting a seed. Supplying a value makes the generated CRC labels reproducible. DLPFC scenarios already contain their corrupted labels and ignore 'seed'.

Value

A 'spatial_refinement_benchmark' ready for [benchmark_spatial_refiners()].

Examples

```
dlpfc <- load_spatial_benchmark("dlpfc", scenario = 1)
dim(dlpfc$xy)

crc <- load_spatial_benchmark("crc", "CRC_random_25_r1", seed = 1040001L)
mean(crc$labels != crc$truth)
```

refine_spatial_labels *Refine noisy categorical labels on a spatial domain*

Description

`refine_spatial_labels()` applies FiberMargin to a coordinate-indexed label field. The function is deterministic: no training phase is run and all route constants are fixed internally. For multiclass data, each class receives a two-sided local evidence score from short directional neighborhood sweeps; in the binary case it applies a fixed nearest-neighbour ballot rule.

Usage

```
refine_spatial_labels(xy, labels, samples = NULL, workers = NULL)
```

Arguments

<code>xy</code>	Numeric matrix with two or three spatial coordinates per row.
<code>labels</code>	Initial categorical assignment for every row of <code>'xy'</code> .
<code>samples</code>	Optional specimen identifier. Different specimens are refined independently.
<code>workers</code>	Optional total CPU budget shared by the native computation. <code>'NULL'</code> uses up to two physical cores. The budget is never multiplied by the number of specimens, and results are independent of the worker count.

Details

The method can process multiple specimens at once. `'samples'` defines independent coordinate systems: coordinates and labels are never shared across specimens, even when their numeric coordinate ranges overlap. Constant coordinate axes are removed separately within each specimen. Consequently, a three-column specimen with constant `'z'` is processed identically to its two-column `'(x, y)'` representation, while genuinely three-dimensional specimens retain all three axes.

Value

A factor with the levels and names of `'labels'`; when `'labels'` is unnamed, row names from `'xy'` are used as names. Attributes `'candidate'`, `'margin_score'`, `'required'`, `'repair_margin'`, `'atlas_dispersion'`, `'isolation'`, and `'changed'` contain pointwise diagnostics. `'margin_score'` is a local support contrast on the internal coordinate lattice, `'repair_margin'` is the difference between `'margin_score'` and the adaptive admission threshold, and `'changed'` marks updated sites. `'isolation'` is a deterministic local-gap protection factor in the multiclass route (one for the binary ballot). A candidate is accepted exactly when it differs from the observed label and `'repair_margin'` is nonnegative. Summary attributes report `'workers'`, `'dimensions_used'`, `'labels_changed'`, `'changed_fraction'`, `'classes_before'`, `'classes_after'`, `'removed_classes'`, and `'sample_sizes'`. Except for `'workers'`, these summaries are named by specimen; class summaries are named lists. Class counts are not constrained, so `'removed_classes'` explicitly records observed classes that disappear after refinement.

Examples

```

sim <- simulate_gradient_regions(n = 2000, samples = 2)
refined <- refine_spatial_labels(sim$xy, sim$labels, sim$samples)
mean(refined == sim$truth)

# A constant z axis is removed within this specimen.
xyz <- cbind(sim$xy, z = 0)
stopifnot(identical(
  refine_spatial_labels(xyz, sim$labels, sim$samples, workers = 1),
  refine_spatial_labels(sim$xy, sim$labels, sim$samples, workers = 1)
))

```

simulate_complex_spatial_domains

Simulate held-out complex spatial domain geometries

Description

Generates two- or three-dimensional geometries designed to stress junctions, narrow channels, non-convex boundaries, and severe regional concentration. These shapes are separate from those in [simulate_spatial_domains()].

Usage

```

simulate_complex_spatial_domains(
  n = 50000L,
  shape = c("voronoi_mosaic", "radial_sectors", "checkerboard_junctions",
    "tubular_network", "braided_channels", "shells_3d"),
  density_profile = c("uniform", "extreme"),
  noise_type = c("random", "boundary", "patch"),
  noise = 0.25,
  samples = 1L,
  k = 5L,
  seed = 1L
)

```

Arguments

n	Number of observations.
shape	Domain geometry.
density_profile	Either "uniform" or "extreme".
noise_type	Either "random", "boundary", or "patch".
noise	Fraction of labels to corrupt. Each true class present in a specimen retains one correct exemplar; a rate that makes this impossible is rejected.
samples	Number of independent tissues.
k	Number of domains.
seed	Random seed.

Value

A 'spatial_refinement_benchmark'.

Examples

```
sim <- simulate_complex_spatial_domains(
  n = 1000, shape = "tubular_network", density_profile = "extreme"
)
table(sim$truth)
```

simulate_gradient_regions

Simulate layered gradient-mixture tissue regions

Description

Generates four ordered tissue areas whose reference classes are 'A', 'B', 'B', and 'C'. Within the areas, a configurable minority fraction is labeled 'B', 'A', 'C', and 'B', respectively. Independent tissues are generated with separate curved boundaries and coordinate offsets.

Usage

```
simulate_gradient_regions(
  n = 50000L,
  minority = 0.05,
  dimensions = 2L,
  samples = 1L,
  seed = 1L,
  curvature = 0.04,
  density_profile = c("uniform", "moderate", "strong", "extreme")
)
```

Arguments

n	Total number of observations across all tissues.
minority	Fraction of minority labels in every area.
dimensions	Either 2 or 3 spatial dimensions.
samples	Number of independent tissues.
seed	Random seed.
curvature	Boundary waviness on the unit coordinate scale.
density_profile	Relative observation concentration across the four tissue areas. Use "uniform", "moderate", "strong", "extreme", or a positive numeric vector of length four.

Value

A 'spatial_refinement_benchmark' containing 'xy', mixed 'labels', reference 'truth', 'samples', 'area', boundary indicators, and sparse-region indicators.

Examples

```
sim <- simulate_gradient_regions(n = 4000, minority = 0.05, samples = 2)
table(sim$area, sim$labels)
```

```
simulate_spatial_clusters
```

Simulate labelled spatial clusters

Description

Simulate labelled spatial clusters

Usage

```
simulate_spatial_clusters(
  n = 50000L,
  dimensions = 2L,
  k = 5L,
  samples = 1L,
  noise = 0.08,
  seed = 1L
)
```

Arguments

n	Number of observations.
dimensions	Spatial dimensions, either 2 or 3.
k	Number of clusters.
samples	Number of independent tissues or sections.
noise	Fraction of labels to corrupt at random. Each true class present in a specimen retains one correct exemplar; a rate that makes this impossible is rejected.
seed	Random seed.

Value

A 'spatial_refinement_benchmark' containing coordinates, noisy labels, truth, sample identifiers, region, boundary, and sparse-region indicators.

Examples

```
sim <- simulate_spatial_clusters(n = 500, dimensions = 3, k = 4)
```

simulate_spatial_domains

Simulate spatial tissue domains with challenging geometry

Description

Generates irregular 2D tissue sections or 3D tissue volumes with known domain labels and corrupted initial cluster assignments.

Usage

```
simulate_spatial_domains(
  n = 50000L,
  pattern = c("jagged_stripes", "wavy_layers", "rings", "spiral", "branching", "lobes",
    "islands", "disconnected", "thin_layers", "intermixed", "layers3d"),
  k = 5L,
  noise = 0.2,
  dimensions = if (identical(pattern[1L], "layers3d")) 3L else 2L,
  samples = 1L,
  noise_type = c("random", "boundary", "patch", "region"),
  feature_scale = 1,
  density_profile = c("uniform", "moderate", "strong", "extreme", "hotspot"),
  seed = 1L
)
```

Arguments

n	Number of observations.
pattern	Domain geometry. See Details.
k	Number of tissue domains.
noise	Fraction of initial labels to corrupt. Each true class present in a specimen retains one correct exemplar; a rate that makes this impossible is rejected.
dimensions	Either 2 or 3. Pattern ‘layers3d’ requires 3.
samples	Number of independent slides or samples.
noise_type	One of “random”, “boundary”, “patch”, or “region”.
feature_scale	Relative width of islands and thin layers. Values below one create more difficult sub-neighborhood structures.
density_profile	Relative observation concentration across tissue regions. Use “uniform”, “moderate”, “strong”, “extreme”, “hotspot”, or a positive numeric vector of length ‘k’. Character profiles permute region weights reproducibly to avoid tying density to a particular class identifier.
seed	Random seed.

Details

Available geometries include jagged and wavy layers, concentric rings, spiral arms, branching sectors, lobes, rare islands, disconnected domains, thin layers, interleaved microdomains, and curved 3D layers.

Value

A ‘spatial_refinement_benchmark’ with coordinates, noisy labels, truth, samples, geometry meta-data, and boundary and sparse-region indicators.

```
simulate_volumetric_domains
```

Simulate complex three-dimensional tissue domains

Description

Generates volumetric spatial-label benchmarks with curved interfaces, disconnected components, thin structures, class imbalance, irregular z-plane acquisition, and spatially structured label corruption.

Usage

```
simulate_volumetric_domains(
  n = 50000L,
  shape = c("concentric_shells", "warped_ellipsoids", "toroidal_compartments",
    "folded_layers", "thin_folded_sheets", "branching_tubes", "disconnected_volumes",
    "helical_channels"),
  acquisition = c("uniform", "class_imbalanced", "irregular_z"),
  noise_type = c("random", "boundary", "patch", "region"),
  noise = 0.25,
  k = 5L,
  samples = 1L,
  seed = 1L
)
```

Arguments

n	Number of observations.
shape	Three-dimensional domain geometry.
acquisition	Sampling design: "uniform", "class_imbalanced", or "irregular_z".
noise_type	Corruption mechanism: "random", "boundary", "patch", or "region".
noise	Fraction of labels to corrupt within each sample. Each true class present in a specimen retains one correct exemplar; a rate that makes this impossible is rejected.
k	Number of domains.
samples	Number of independent tissue volumes.
seed	Random seed.

Value

A ‘spatial_refinement_benchmark’ with three coordinate columns.

Examples

```
sim <- simulate_volumetric_domains(
  n = 1000, shape = "folded_layers", noise_type = "boundary", seed = 8
)
dim(sim$xy)
```

spatial_benchmark	<i>Construct a spatial refinement benchmark</i>
-------------------	---

Description

Creates a validated benchmark object that is consumed by simulations, refinement runs, and benchmarking utilities. All fields are aligned by spot index, so every function in the package sees the same coordinates, labels, and optional evaluation masks.

Usage

```
spatial_benchmark(
  xy,
  labels,
  truth,
  samples = NULL,
  boundary = NULL,
  regions = NULL,
  sparse = NULL,
  name = NULL,
  metadata = list()
)
```

Arguments

xy	Numeric matrix containing two or three spatial coordinates per observation.
labels	Initial cluster assignments.
truth	Reference assignments used only for evaluation.
samples	Optional tissue or section identifier.
boundary	Optional logical vector marking boundary observations.
regions	Optional region identifier used for defining sparse-versus-dense accuracy when ‘sparse’ is not supplied.
sparse	Optional logical vector marking a user-provided sparse subset.
name	Optional benchmark name.
metadata	Optional named list with provenance or scenario information.

Value

An object of class 'spatial_refinement_benchmark'.

Examples

```
sim <- simulate_spatial_domains(n = 500, pattern = "jagged_stripes")
bench <- spatial_benchmark(
  sim$xy, sim$labels, sim$truth, sim$samples,
  boundary = sim$boundary, regions = sim$region
)
```

Index

* datasets

colorectal_benchmark, [4](#)

dlpfc_benchmark, [6](#)

available_spatial_benchmarks, [2](#)

benchmark_spatial_refiners, [3](#)

clean_categorical_mask, [4](#)

colorectal_benchmark, [4](#)

corrupt_categorical_mask, [5](#)

dlpfc_benchmark, [6](#)

evaluate_mask_cleaning, [7](#)

evaluate_spatial_refinement, [7](#)

load_spatial_benchmark, [9](#)

refine_spatial_labels, [10](#)

simulate_complex_spatial_domains, [11](#)

simulate_gradient_regions, [12](#)

simulate_spatial_clusters, [13](#)

simulate_spatial_domains, [14](#)

simulate_volumetric_domains, [15](#)

spatial_benchmark, [16](#)