Package: smoothclust (via r-universe)

November 25, 2026

Version 1.10.0

Title smoothclust

Description Tools for spatial domain analysis in spatial transcriptomics data. Includes spatial smoothing of gene expression profiles for downstream clustering and a boundary density (BD) metric for summarizing spatial domain organization. The smoothing function returns processed expression values for subsequent clustering. BD requires only spatial coordinates and categorical labels, and provides local and sample-level summaries of label transitions.

URL https://github.com/lmweber/smoothclust

BugReports https://github.com/lmweber/smoothclust/issues

License MIT + file LICENSE

Encoding UTF-8

biocViews Spatial, SingleCell, Transcriptomics, GeneExpression, Clustering

Depends R (>= 4.4.0)

Imports SpatialExperiment, SummarizedExperiment, S4Vectors, BiocNeighbors, Matrix, methods, utils

VignetteBuilder knitr

Suggests BiocStyle, knitr, STexampleData, scran, scater, ggspavis, testthat

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libmagick++-dev gsfonts libicu-dev libssl-dev zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:52:14 UTC

RemoteUrl https://github.com/bioc/smoothclust

RemoteRef RELEASE_3_24

RemoteSha 2f06abc9f17b7757c17dd3817c4b3527f776a515

Contents

boundary_density 2
smoothclust 3

Index **7**

`boundary_density` *Boundary density metric*

Description

Function for boundary density metric

Usage

```
boundary_density(spatial_coords, labels, k = 6, n_threads = 1, adjust = FALSE)
```

Arguments

`spatial_coords`	Numeric matrix containing spatial coordinates of points, formatted as nrow = number of points, ncol = 2 (assuming x and y dimensions). For example, 'spatial_coords = spatialCoords(spe)' if using a `SpatialExperiment` object.
`labels`	Atomic vector or factor containing cluster labels for each point. Labels must correspond to coordinate rows in the same order. Missing values are not allowed. For example, 'labels <- colData(spe)$label' if using a `SpatialExperiment` object.
`k`	Number of nearest neighbors to use in calculation, excluding the focal location. This count is independent of the k argument used for smoothing in `smoothclust`. Default = 6 (e.g. for hexagonal arrangement in 10x Genomics Visium platform).
`n_threads`	Number of threads to use for nearest-neighbor searches. Default = 1.
`adjust`	Whether to calculate the composition adjustment. With `FALSE`, only the raw boundary density is returned. With `TRUE`, the analytic composition adjustment returns the expectation under random relabeling preserving observed label counts and the ratio of raw BD to this expectation (relative BD). Default = `FALSE`.

Details

Function to calculate the raw boundary density metric, defined as the average fraction of nearest neighbors per point that are from a different cluster. Raw BD summarizes observed label-transition frequency on the nearest-neighbor graph.

An optional analytic composition adjustment provides the sample-level relative BD. Relative BD is a composition-adjusted summary, calculated by dividing raw BD by its analytic expectation under random relabeling with the observed label counts. The expectation is calculated directly from label counts. Values below, equal to, or above 1 indicate fewer, the same, or more transitions than expected under this fixed-count random-label reference.

Value

Returns a list containing values at each point (i.e. the number of nearest neighbors that are from a different cluster, the number of nearest neighbors, and the local raw boundary density) as well as the mean discordant neighbor count and the sample-level raw boundary density. If `adjust = TRUE`, the list also contains the expected boundary density and sample-level relative boundary density.

Examples

```
spatial_coords <- cbind(c(0, 1, 3, 6), 0)
labels <- c("A", "B", "A", "A")

# calculate raw boundary density metric
res <- boundary_density(spatial_coords, labels, k = 2)
str(res)
res$n_discordant
res$mean_discordant
res$boundary_density

# calculate analytic composition adjustment
res_adj <- boundary_density(spatial_coords, labels, k = 2,
                            adjust = TRUE)
res_adj$expected_boundary_density
res_adj$relative_boundary_density
```

`smoothclust` *smoothclust*

Description

Smooths expression profiles across spatial neighbors for downstream spatial domain analysis.

Usage

```
smoothclust(
  input,
  assay_name = "counts",
  spatial_coords = NULL,
  method = c("uniform", "kernel", "knn"),
  bandwidth = 0.05,
  k = 18,
  truncate = 0.05,
  n_threads = 1,
  normalize = TRUE,
  log = TRUE,
  size_factors = NULL,
  pseudo_count = 1,
  output_assay_name = NULL
)
```

Arguments

input	Input data, which can be provided as either a SpatialExperiment object or a numeric matrix. All expression values must be finite. With the default preprocessing settings, the selected assay or matrix is assumed to contain non-negative raw expression counts. Pre-normalized or log-transformed values may also be supplied if normalize and log are set appropriately. Spatial coordinates are obtained from the spatialCoords slot for a SpatialExperiment, or must be supplied separately using spatial_coords for a numeric matrix.
assay_name	For a SpatialExperiment input object, this argument specifies the name of the assay containing the expression values to be processed and smoothed. With the default preprocessing settings, this should contain raw expression counts. If an assay containing normalized or log-transformed values is used, the normalize and log arguments should be set to avoid applying either step twice. This argument is only used if the input is a SpatialExperiment object. Default = counts.
spatial_coords	Numeric matrix of spatial coordinates, assumed to contain x coordinates in first column and y coordinates in second column. Rows must correspond to expression columns in the same order. This argument is only used if the input is a numeric matrix.
method	Method used for smoothing. Options are uniform, kernel, and knn. All methods include the focal location. The uniform method calculates unweighted averages across spatial locations within a circular window with radius bandwidth at each spatial location, which smooths out spatial variability as well as sparsity due to sampling variability. The kernel method calculates a weighted average using a truncated exponential kernel applied to Euclidean distances. Nearby locations receive greater weight, with the length scale controlled by bandwidth. This method may be affected by sparsity due to sampling variability (especially sparsity at the focal location) and is computationally slower. The knn method calculates an unweighted average across the focal location and its k nearest other locations, and is the fastest method. Default = uniform.
bandwidth	Bandwidth parameter for smoothing, expressed as proportion of width or height (whichever is greater) of tissue area. Only used for method = "uniform" or method = "kernel". For method = "uniform", the bandwidth represents the radius of a circle, and unweighted averages are calculated across neighboring points within this circle. For method = "kernel", the averaging is weighted by distances scaled using a truncated exponential kernel applied to Euclidean distances. For example, a bandwidth of 0.05 will smooth values across neighbors weighted by distances scaled using a truncated exponential kernel with length scale equal to 5 area. Weights for method = "kernel" are truncated at small values for computational efficiency. Default = 0.05.
k	Number of nearest other locations for method = "knn". Unweighted averages are calculated across the focal location and its k nearest neighbors, for k + 1 locations in total. This smoothing-neighbor count is independent of the k argument in boundary_density. Default = 18 (based on two layers in honeycomb pattern for 10x Genomics Visium platform).
truncate	Truncation threshold parameter if method = "kernel". Kernel weights below

this value are set to zero for computational efficiency. Only used for `method = "kernel"`. Default = 0.05.

`n_threads` Number of threads to use for nearest-neighbor searches. Default = 1.

`normalize` Whether to perform library size normalization before spatial smoothing. Expression values in each column are divided by a size factor rescaled to have mean one. Default = `TRUE`.

`log` Whether to apply a log2 transformation after spatial smoothing. The transformation is `log2(x + pseudo_count)`. Default = `TRUE`.

`size_factors` Optional positive numeric vector with one value per spatial location, used for normalization when `normalize = TRUE`. Supplied values are rescaled to have mean one. If the vector is named, its names must match the column names of `input`, and values are reordered to match the input columns. If `NULL`, size factors are calculated from column sums of the selected, unsmoothed input assay or matrix. To calculate size factors from a broader set of genes than those supplied to `smoothclust`, calculate them before gene filtering and provide them here. When valid size factors are supplied, columns with zero sums in the selected input are permitted; when `NULL`, every input column must have a positive sum. Default = `NULL`.

`pseudo_count` Positive numeric scalar added before log2 transformation when `log = TRUE`. The default value of 1 preserves sparse zeros. Non-default values may produce a dense output matrix. Default = 1.

`output_assay_name`
Optional name for the output assay when `input` is a `SpatialExperiment`. If `NULL`, the name is `logcounts_smooth` when `log = TRUE`, `normcounts_smooth` when `normalize = TRUE` and `log = FALSE`, and `<assay_name>_smooth` otherwise. An error is raised if the output assay already exists. This argument is ignored for a numeric matrix input. Default = `NULL`.

Details

The function returns processed expression values for downstream feature selection, dimensionality reduction, and clustering.

When `normalize = TRUE`, library size normalization is performed before spatial smoothing. This ensures that neighboring locations contribute according to the spatial weights rather than their sequencing depths. When `log = TRUE`, log2 transformation is also performed after smoothing. Size factors should be calculated from the desired reference set of genes before filtering and supplied using `size_factors`.

Value

Returns processed and spatially smoothed expression values for use in downstream analyses. Results are returned either as a `SpatialExperiment` object containing a new assay, or as a numeric matrix-like object, depending on the input type.

Examples

```
set.seed(123)
```

```
input <- matrix(rpois(60, lambda = 10), nrow = 10)
spatial_coords <- cbind(rep(1:3, each = 2), rep(1:2, times = 3))

# normalize, smooth, and log-transform a numeric expression matrix
out <- smoothclust(input, spatial_coords = spatial_coords, bandwidth = 0.6)
dim(out)

# supply size factors calculated before any gene filtering
size_factors <- colSums(input)
out_precomputed <- smoothclust(input, spatial_coords = spatial_coords,
                               bandwidth = 0.6,
                               size_factors = size_factors)
dim(out_precomputed)
```

Index

boundary_density, 2

smoothclust, 3