Package 'Banksy'

November 5, 2025

Title Spatial transcriptomic clustering

Version 1.7.0

Description Banksy is an R package that incorporates spatial information to cluster cells in a feature space (e.g. gene expression). To incorporate spatial information, BANKSY computes the mean neighborhood expression and azimuthal Gabor filters that capture gene expression gradients. These features are combined with the cell's own expression to embed cells in a neighbor-augmented product space which can then be clustered, allowing for accurate and spatially-aware cell typing and tissue domain segmentation.

Depends R (>= 4.4.0)

Imports aricode, BiocParallel, data.table, dbscan, SpatialExperiment, SingleCellExperiment, SummarizedExperiment, S4Vectors, stats, Matrix, MatrixGenerics, mclust, igraph, irlba, leidenAlg (>= 1.1.0), utils, uwot, RcppHungarian, GenomeInfoDb

License file LICENSE

Encoding UTF-8

URL https://github.com/prabhakarlab/Banksy

BugReports https://github.com/prabhakarlab/Banksy/issues

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, pals, scuttle, scater, scran, cowplot, ggplot2, testthat (>= 3.0.0), harmony, Seurat, ExperimentHub, spatialLIBD, BiocStyle

VignetteBuilder knitr

Config/testthat/edition 3

biocViews Clustering, Spatial, SingleCell, GeneExpression, DimensionReduction

git_url https://git.bioconductor.org/packages/Banksy

git_branch devel

git_last_commit 4bdc545

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-11-05

Author Vipul Singhal [aut],
Joseph Lee [aut, cre] (ORCID: <https://orcid.org/0000-0002-4983-4714>)

Maintainer Joseph Lee <joseph.lee@u.nus.edu>

Contents

`Banksy-package` *The Banksy package*

Description

Banksy is a library and R package for network analysis.

Description

Banksy is an R package that incorporates spatial information to cluster cells in a feature space (e.g. gene expression). To incorporate spatial information, BANKSY computes the mean neighborhood expression and azimuthal Gabor filters that capture gene expression gradients. These features are combined with the cell's own expression to embed cells in a neighbor-augmented product space which can then be clustered, allowing for accurate and spatially-aware cell typing and tissue domain segmentation.

Details

For a quick start to the package, please refer to the GitHub page at https://github.com/prabhakarlab/Banksy. For in-depth guides to package functionality and use cases, refer to the package webpage at https://prabhakarlab.github.io/Banksy.

Author(s)

Maintainer: Joseph Lee <joseph.lee@u.nus.edu> (ORCID)

Authors:

- Vipul Singhal

See Also

Useful links:

- https://github.com/prabhakarlab/Banksy
- Report bugs at https://github.com/prabhakarlab/Banksy/issues

---

`clusterBanksy` *Perform clustering in BANKSY's neighborhood-augmented feature space.*

Description

Perform clustering in BANKSY's neighborhood-augmented feature space.

Usage

```
clusterBanksy(
  se,
  use_agf = FALSE,
  lambda = 0.2,
  use_pcs = TRUE,
  npcs = 20L,
  dimred = NULL,
  ndims = NULL,
  assay_name = NULL,
  group = NULL,
  algo = c("leiden", "louvain", "kmeans", "mclust"),
  k_neighbors = 50,
  resolution = 1,
  leiden.iter = -1,
  kmeans.centers = 5,
  mclust.G = 5,
  M = NULL,
  seed = NULL,
  ...
)
```

Arguments

se	A SpatialExperiment, SingleCellExperiment or SummarizedExperiment object with computeBanksy ran.
use_agf	A logical vector specifying whether to use the AGF for clustering.
lambda	A numeric vector in $\in [0, 1]$ specifying a spatial weighting parameter. Larger values (e.g. 0.8) incorporate more spatial neighborhood and find spatial domains, while smaller values (e.g. 0.2) perform spatial cell-typing.
use_pcs	A logical scalar specifying whether to cluster on PCs. If FALSE, runs on the BANKSY matrix.
npcs	An integer scalar specifying the number of principal components to use if use_pcs is TRUE.
dimred	A string scalar specifying the name of an existing dimensionality reduction result to use. Will overwrite use_pcs if supplied.
ndims	An integer scalar specifying the number of dimensions to use if dimred is supplied.
assay_name	A string scalar specifying the name of the assay used in computeBanksy.
group	A string scalar specifying a grouping variable for samples in se. This is used to scale the samples in each group separately.
algo	A string scalar specifying the clustering algorithm to use; one of leiden, louvain, mclust, kmeans.
k_neighbors	An integer vector specifying number of neighbors for constructing sNN (for louvain / leiden).
resolution	A numeric vector specifying resolution used for clustering (louvain / leiden).
leiden.iter	An integer scalar specifying the number of leiden iterations. For running till convergence, set to -1 (leiden).
kmeans.centers	An integer vector specifying the number of kmeans clusters (kmeans).
mclust.G	An integer vector specifying the number of mixture components (Mclust).
M	Advanced usage. An integer vector specifying the highest azimuthal Fourier harmonic to cluster with. If specified, overwrites the use_agf argument.
seed	Random seed for clustering. If not specified, no seed is set.
...	to pass to methods

Details

This function performs clustering on the principal components computed on the BANKSY matrix, i.e., the BANKSY embedding. The PCA corresponding to the parameters use_agf and lambda must have been computed with runBanksyPCA. Clustering may also be performed directly on the BANKSY matrix with use_pcs set to FALSE (this is not recommended).

Four clustering algorithms are implemented.

- leiden: Leiden graph-based clustering. The arguments k_neighbors and resolution should be specified.

- louvain: Louvain graph-based clustering. The arguments `k_neighbors` and `resolution` should be specified.
- kmeans: kmeans clustering. The argument `kmeans.centers` should be specified.
- mclust: Gaussian mixture model-based clustering. The argument `mclust.G` should be specified.

By default, no seed is set for clustering. If a seed is specified, the same seed is used for clustering across the input parameters.

Value

A SpatialExperiment / SingleCellExperiment / SummarizedExperiment object with cluster labels in `colData(se)`.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
spe <- runBanksyPCA(spe, M = 1, lambda = c(0, 0.2), npcs = 20)
spe <- clusterBanksy(spe, M = 1, lambda = c(0, 0.2), resolution = 1)
```

`clusterNames` *Get names of clustering runs.*

Description

Get names of clustering runs.

Usage

```
clusterNames(se)
```

Arguments

`se`	A `SpatialExperiment`, `SingleCellExperiment` or `SummarizedExperiment` object with `clusterBanksy` ran.

Value

A character vector of names of clustering runs.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
spe <- runBanksyPCA(spe, M = 1, lambda = c(0, 0.2), npcs = 20)
spe <- clusterBanksy(spe, M = 1, lambda = c(0, 0.2), resolution = 1)
clusterNames(spe)
```

compareClusters *Compare cluster outputs based on various clustering comparison measures.*

Description

Compare cluster outputs based on various clustering comparison measures.

Usage

```
compareClusters(
  se,
  func = c("ARI", "AMI", "MARI", "MARIraw", "RI", "NID", "NMI", "NVI"),
  digits = 3
)
```

Arguments

se	A SpatialExperiment, SingleCellExperiment or SummarizedExperiment object with cluster labels in colData(se).
func	A string scalar specifying what clustering comparison measure to compute. See ?aricode for more information.
digits	An integer scalar specifying the number of digits to round to.

Value

A matrix of cluster comparison measures.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
spe <- runBanksyPCA(spe, M = 1, lambda = 0.2, npcs = 20)
spe <- clusterBanksy(spe, M = 1, lambda = 0.2, resolution = c(0.1, 1))
spe <- connectClusters(spe)
compareClusters(spe)
```

computeBanksy *Compute the component neighborhood matrices for the BANKSY matrix.*

Description

Compute the component neighborhood matrices for the BANKSY matrix.

Usage

```
computeBanksy(
  se,
  assay_name,
  coord_names = NULL,
  compute_agf = FALSE,
  k_geom = 15,
 spatial_mode = c("kNN_median", "kNN_r", "kNN_rn", "kNN_rank", "kNN_unif", "rNN_gauss"),
  n = 2,
  sigma = 1.5,
  alpha = 0.05,
  k_spatial = 100L,
  M = NULL,
  sample_size = NULL,
  sample_renorm = TRUE,
  seed = NULL,
  dimensions = "all",
  center = TRUE,
  chunk_size = NULL,
  parallel = FALSE,
  num_cores = NULL,
  row_limit_factor = 0.75,
  verbose = TRUE
)
```

Arguments

se	A SpatialExperiment, SingleCellExperiment or SummarizedExperiment object. If not a SpatialExperiment object, argument coord_names must be provided.
assay_name	A string scalar specifying the name of the assay to use.
coord_names	A string vector specifying the names in colData corresponding to spatial coordinates.
compute_agf	A logical scalar specifying whether to compute the AGF.
k_geom	An integer scalar specifying the number of neighbors to use. Values $\in [15, 30]$ work well.

`spatial_mode` A string scalar specifying the kernel for neighborhood computation (default: kNN_median).

- kNN_median: k-nearest neighbors with median-scaled Gaussian kernel
- kNN_r: k-nearest neighbors with $1/r$ kernel
- kNN_rn: k-nearest neighbors with $1/r^n$ kernel
- kNN_rank: k-nearest neighbors with rank Gaussian kernel
- kNN_unif: k-nearest neighbors wth uniform kernel
- rNN_gauss: radial nearest neighbors with Gaussian kernel

`n` A numeric scalar specifying the exponent of radius (for kNN_rn).

`sigma` A numeric scalar specifying the std. dev. of Gaussian kernel (for rNN_gauss).

`alpha` A numeric scalar specifying the radius used: larger alphas give smaller radii (for rNN_gauss).

`k_spatial` An integer scalar specifying the initial number of neighbors to use (for rNN_gauss)

`M` Advanced usage. A integer scalar specifying the highest azimuthal Fourier harmonic to compute. If specified, overwrites the `use_agf` argument.

`sample_size` An integer scalar number of neighbors to sample from the neighborhood.

`sample_renorm` A logical scalar specifying whether to renormalize the neighbor weights to 1.

`seed` An integer scalar specifying seed for sampling the neighborhood.

`dimensions` A character vector specifying the dimensions to use when computing neighborhood.

- subset of colnames of cell.locs
- allUses all colnames of spatialCoords to compute (default)

`center` A logical scalar specifying whether to center higher order harmonics in local neighborhoods.

`chunk_size` A integer scalar specifying the number of rows / genes of the neighborhood cell matrix to compute. Must be strictly less than floor of 2e31-1 / number of cells, though this may still give rise to negative length vector errors. For safety, set this to less than 2e31-1 * row_limit_factor / number of cells. This is automatically computed based on the latter, but can be specified.

`parallel` A logical scalar specifying whether to compute chunks in parallel using bplapply. Not implemented for Windows.

`num_cores` A integer scalar specifying the number of cores to use if parallel is TRUE.

`row_limit_factor` A numeric scalar specifying the safety factor applied to the maximum vector length (2^31-1) when computing chunk sizes. Accounts for data.table overhead during grouping operations. Default is 0.75.

`verbose` A logical scalar specifying verbosity.

Details

Given an expression matrix (as specified by `assay_name`), this function computes the mean neighborhood matrix (`H0`) and optionally, the azimuthal Gabor filter (AGF) matrix (`H1`). The number of neighbors used to define the spatial neighborhood is given by `k_geom`. Different kernels may be used to compute the neighborhood features, specified by `spatial_mode`.

Value

A SpatialExperiment / SingleCellExperiment / SummarizedExperiment object with neighborhood matrices added.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
```

`connectClusters` *Relabel cluster labels across parameter runs to maximise their similarity.*

Description

Relabel cluster labels across parameter runs to maximise their similarity.

Usage

```
connectClusters(se, map_to = NULL, verbose = TRUE)
```

Arguments

`se`	A `SpatialExperiment`, `SingleCellExperiment` or `SummarizedExperiment` object with cluster labels in `colData(se)`.
`map_to`	A string scalar specify a cluster to map to.
`verbose`	A logical scalar specifying verbosity.

Value

A SpatialExperiment / SingleCellExperiment / SummarizedExperiment object with 'connected' cluster labels in `colData(se)`.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
spe <- runBanksyPCA(spe, M = 1, lambda = c(0, 0.2), npcs = 20)
spe <- clusterBanksy(spe, M = 1, lambda = c(0, 0.2), resolution = 1)
spe <- connectClusters(spe)
```

`getBanksyMatrix` *Builds the BANKSY matrix from neighborhood matrices.*

Description

Builds the BANKSY matrix from neighborhood matrices.

Usage

```
getBanksyMatrix(
  se,
  M,
  lambda,
  assay_name = NULL,
  scale = FALSE,
  group = NULL,
  verbose = TRUE
)
```

Arguments

`se`	A `SpatialExperiment`, `SingleCellExperiment` or `SummarizedExperiment` object with `computeBanksy` ran.
`M`	A integer scalar specifying the highest azimuthal Fourier harmonic to compute.
`lambda`	A numeric vector in $\in [0, 1]$ specifying a spatial weighting parameter. Larger values (e.g. 0.8) incorporate more spatial neighborhood and find spatial domains, while smaller values (e.g. 0.2) perform spatial cell-typing.
`assay_name`	A string scalar specifying the name of the assay used in `computeBanksy`.
`scale`	A logical scalar specifying whether to scale the features to zero mean and unit standard deviation. This is performed before multiplying the assays by their corresponding lambda weighting factors.
`group`	A string scalar specifying a grouping variable for samples in `se`. This is used to scale the samples in each group separately.
`verbose`	A logical scalar specifying verbosity.

Details

After computation of the neighborhood matrices (see computeBanksy), this function builds the BANKSY matrix by concatenating the original expression matrix with the neighborhood matrices, and scales each matrix by an appropriate weight as determined by `lambda`. The weights of the own expression matrix, mean neighborhood matrix and azimuthal Gabor filter are given by $\sqrt{1-\lambda}$, $\sqrt{\lambda/\mu}$ and $\sqrt{\lambda/2\mu}$ respectively, where $\mu = 1.5$. In the case where the AGF is not computed, the weights for the own and mean neighborhood expression matrix simplify to $\sqrt{1-\lambda}$ and $\sqrt{\lambda}$ respectively.

Value

BANKSY matrix.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
banksyMatrix <- getBanksyMatrix(spe, M = 1, lambda = 0.2)
```

---

hippocampus *Mouse Hippocampus VeraFISH data*

Description

This dataset comprises VeraFISH profiling of cells in the mouse hippocampus. Gene expression and cell centroids for 10,944 cells and 129 genes in 2 spatial dimensions are provided. For details on how this dataset was generated, refer to Supplementary Information section 2.2 of our preprint.

Usage

```
data(hippocampus)
```

Format

A list with 2 entries:

expression (matrix) gene expression matrix

locations (data.frame) cell centroids in 2D

Value

List with expression and locations

---

rings *An unrealistic simulation of spatially-resolved omics data.*

Description

This dataset comprises gene expression and spatial coordinates for 50 genes and 308 cells from 4 clusters (`rings$clusters`). See `system.file('scripts/rings.R', package='Banksy')` on how this dataset was generated.

Usage

```
data(rings)
```

Format

A SpatialExperiment object.

Value

A SpatialExperiment object

`runBanksyPCA` *Run PCA on a BANKSY matrix.*

Description

Run PCA on a BANKSY matrix.

Usage

```
runBanksyPCA(
  se,
  use_agf = FALSE,
  lambda = 0.2,
  npcs = 20L,
  assay_name = NULL,
  scale = TRUE,
  group = NULL,
  M = NULL,
  seed = NULL
)
```

Arguments

`se`	A `SpatialExperiment`, `SingleCellExperiment` or `SummarizedExperiment` object with `computeBanksy` ran.
`use_agf`	A logical vector specifying whether to use the AGF for computing principal components.
`lambda`	A numeric vector in $\in [0, 1]$ specifying a spatial weighting parameter. Larger values (e.g. 0.8) incorporate more spatial neighborhood and find spatial domains, while smaller values (e.g. 0.2) perform spatial cell-typing.
`npcs`	An integer scalar specifying the number of principal components to compute.
`assay_name`	A string scalar specifying the name of the assay used in `computeBanksy`.
`scale`	A logical scalar specifying whether to scale features before PCA. Defaults to TRUE.
`group`	A string scalar specifying a grouping variable for samples in `se`. This is used to scale the samples in each group separately.
`M`	Advanced usage. An integer vector specifying the highest azimuthal Fourier harmonic to use. If specified, overwrites the `use_agf` argument.
`seed`	Seed for PCA. If not specified, no seed is set.

Details

This function runs PCA on the BANKSY matrix (see getBanksyMatrix) with features scaled to zero mean and unit standard deviation.

Value

A SpatialExperiment / SingleCellExperiment / SummarizedExperiment object with PC coordinates in `reducedDims(se)`.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
spe <- runBanksyPCA(spe, M = 1, lambda = 0.2, npcs = 20)
```

---

runBanksyUMAP *Run UMAP on a BANKSY embedding.*

Description

Run UMAP on a BANKSY embedding.

Usage

```
runBanksyUMAP(
  se,
  use_agf = FALSE,
  lambda = 0.2,
  use_pcs = TRUE,
  npcs = 20L,
  dimred = NULL,
  ndims = NULL,
  assay_name = NULL,
  scale = TRUE,
  group = NULL,
  n_neighbors = 30L,
  spread = 3,
  min_dist = 0.1,
  n_epochs = 300L,
  M = NULL,
  seed = NULL,
  ...
)
```

Arguments

se	A SpatialExperiment, SingleCellExperiment or SummarizedExperiment object with computeBanksy ran.
use_agf	A logical vector specifying whether to use the AGF for computing UMAP.
lambda	A numeric vector in $\in [0, 1]$ specifying a spatial weighting parameter. Larger values (e.g. 0.8) incorporate more spatial neighborhood and find spatial domains, while smaller values (e.g. 0.2) perform spatial cell-typing.
use_pcs	A logical scalar specifying whether to run UMAP on PCs. If FALSE, runs on the BANKSY matrix.
npcs	An integer scalar specifying the number of principal components to use if use_pcs is TRUE.
dimred	A string scalar specifying the name of an existing dimensionality reduction result to use. Will overwrite use_pcs if supplied.
ndims	An integer scalar specifying the number of dimensions to use if dimred is supplied.
assay_name	A string scalar specifying the name of the assay used in computeBanksy.
scale	A logical scalar specifying whether to scale features before UMAP. Only used when use_pcs is FALSE. Defaults to TRUE.
group	A string scalar specifying a grouping variable for samples in se. This is used to scale the samples in each group separately.
n_neighbors	An integer scalar specifying the number of neighbors to use for UMAP.
spread	A numeric scalar specifying the effective scale of embedded points.
min_dist	A numeric scalar specifying the effective min. dist. between embedded points.
n_epochs	An integer scalar specifying the number of epochs to run UMAP optimization.
M	Advanced usage. An integer vector specifying the highest azimuthal Fourier harmonic to use. If specified, overwrites the use_agf argument.
seed	Seed for UMAP. If not specified, no seed is set.
...	parameters to pass to uwot::umap

Details

This function runs UMAP on the principal components computed on the BANKSY matrix.

Value

A SpatialExperiment / SingleCellExperiment / SummarizedExperiment object with UMAP coordinates in reducedDims(se).

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
spe <- runBanksyPCA(spe, M = 1, lambda = 0.2, npcs = 20)
spe <- runBanksyUMAP(spe, M = 1, lambda = 0.2)
```

`simulateDataset` *Simulate an unrealistic spatial omics dataset.*

Description

Simulate an unrealistic spatial omics dataset.

Usage

```
simulateDataset(n_cells = 300, n_genes = 30, n_rings = 3, rate = 10)
```

Arguments

`n_cells`	An integer scalar specifying the approximate number of cells.
`n_genes`	An integer scalar specifying the number of genes.
`n_rings`	An integer scalar specifying the number of spatial rings.
`rate`	A numeric scalar specifying the Poisson rate parameter for simulating counts.

Details

This function generates an unrealistic spatial omics dataset based on a user-specified number of cells and genes. The number of clusters is defined by `n_rings`, while counts follow a Poisson distribution with a user-specified rate `rate`. The simulation is set up such that the number of cells in each cluster is uniformly distributed; as such, the final number of cells is approximately equal to the user-specified number of cells.

Value

A SpatialExperiment object.

Examples

```
set.seed(2023)
rings <- simulateDataset(n_cells = 5e3, n_genes = 50, n_rings = 8)
rings
table(rings$cluster)
df <- cbind.data.frame(
    SummarizedExperiment::colData(rings),
    SpatialExperiment::spatialCoords(rings))
library(ggplot2)
ggplot(df, aes(x=x, y=y, col=cluster)) + geom_point() + theme_classic()
```

smoothLabels *k-Nearest neighbor cluster label smoothing.*

Description

k-Nearest neighbor cluster label smoothing.

Usage

```
smoothLabels(
  se,
  cluster_names = NULL,
  coord_names = NULL,
  k = 15L,
  prop_thres = 0.5,
  max_iter = 10,
  verbose = TRUE
)
```

Arguments

se	A SpatialExperiment, SingleCellExperiment or SummarizedExperiment object with cluster labels in colData(se).
cluster_names	A string vector of label names to smooth. If NULL, smooths labels in colData(se) matching /^clust/
coord_names	A string vector specifying the names in colData corresponding to spatial coordinates.
k	An integer scalar specifying number of neighbors for smooething.
prop_thres	A numeric scalar $\in [0, 1]$ specifying a label proportions threshold If the fraction of neighbors with a certain label exceeds this proportion, change the label of the current sample (default: 0.5).
max_iter	An integer scalar specifying the max number of smoothing iterations. Set to -1 for smoothing to convergence.
verbose	A logical scalar specifying verbosity.

Details

As described in SpiceMix (https://doi.org/10.1038/s41588-022-01256-z). Implemented for labels that can be coerced to numeric only.

Value

A SpatialExperiment / SingleCellExperiment / SummarizedExperiment object with smoothed cluster labels in colData(se) suffixed with '_smooth'.

Examples

```
data(rings)
spe <- computeBanksy(rings, assay_name = "counts", M = 1, k_geom = c(15, 30))
spe <- runBanksyPCA(spe, M = 1, lambda = 0.2, npcs = 20)
spe <- clusterBanksy(spe, M = 1, lambda = 0.2, resolution = 1)
spe <- smoothLabels(spe, cluster_names = "clust_M1_lam0.2_k50_res1")
```

Index