Package: coFAST (via r-universe)

October 18, 2026

Type Package

Title Spatially-Aware Cell Clustering Algorithm with Cluster Significant Assessment

Version 0.3.0

Date 2026-06-04

Author Wei Liu [aut, cre], Xiao Zhang [aut], Yi Yang [aut], Peng Xie [aut], Chengqi Lin [aut], Jin Liu [aut]

Maintainer Wei Liu <liuweideng@gmail.com>

Description A spatially-aware cell clustering algorithm is provided with cluster significance assessment. It comprises four key modules: spatially-aware cell-gene co-embedding, cell clustering, signature gene identification, and cluster significant assessment. More details can be referred to Peng Xie, et al. (2025) <doi:10.1016/j.cell.2025.05.035>.

License GPL-3

Depends R (>= 4.0.0),

Imports Rcpp (>= 1.0.10), furrr, purrr, future, ggplot2, irlba, DR.SC, ProFAST, Matrix, ade4, progress, pbapply, dplyr, Seurat, stats, utils

LazyData true

URL https://github.com/feiyoung/coFAST

BugReports https://github.com/feiyoung/coFAST/issues

Suggests knitr, rmarkdown, scater, PRECAST, ggrepel, RANN, grDevices

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2026-06-04 13:20:02 UTC

RemoteUrl https://github.com/cran/coFAST

RemoteRef HEAD

RemoteSha 6e705e672cdf871e9ffabc0c6ade87eea1cf841e

Contents

AddAdj 2
AddCluster 4
Addcoord2embed 5
AggregationScore 6
coembed_plot 7
coembedding_umap 8
coFAST 10
CosMx_subset 11
diagnostic.cor.eigs 11
find.signature.genes 13
get.top.signature.dat 14
NCFM 15
pbmc3k_subset 16
pdistance 16
top5_signatures 17

Index **18**

`AddAdj` *Calculate the adjacency matrix given a spatial coordinate matrix*

Description

Calculate the adjacency matrix given a spatial coordinate matrix with 2-dimension or 3-dimension or more.

Usage

```
AddAdj(
  pos,
  type = "fixed_distance",
  platform = c("Others", "Visium", "ST"),
  neighbors = 6,
  ...
)
```

Arguments

pos	a matrix object, with columns representing the spatial coordinates that can be any diemsion, i.e., 2, 3 and >3.
type	an optional string, specify which type of neighbors' definition. Here we provide two definition: one is "fixed_distance", the other is "fixed_number".
platform	a string, specify the platform of the provided data, default as "Others". There are more platforms to be chosen, including "Visuim", "ST" and "Others" ("Others" represents the other SRT platforms except for 'Visium' and 'ST') The platform helps to calculate the adjacency matrix by defining the neighborhoods when type="fixed_distance" is chosen.
neighbors	an optional postive integer, specify how many neighbors used in calculation, default as 6.
...	Other arguments passed to getAdj_auto.

Details

When the type = "fixed_distance", then the spots within the Euclidean distance cutoffs from one spot are regarded as the neighbors of this spot. When the type = "fixed_number", the K-nearest spots are regarded as the neighbors of each spot.

Value

return a sparse matrix, representing the adjacency matrix.

References

None

See Also

None

Examples

```
data(CosMx_subset)
pos <- as.matrix(CosMx_subset@meta.data[,c("x", "y")])
Adj_sp <- AddAdj(pos)
```

`AddCluster` *Find clusters for SRT data*

Description

Identify clusters of spots by a shared nearest neighbor (SNN) modularity optimization based on coFAST's embeddings.

Usage

```
AddCluster(
  seu,
  reduction = "cofast",
  cluster.name = "cofast.cluster",
  res = 0.8,
  K = NULL,
  res.start = 0.2,
  res.end = 2,
  step = 0.02
)
```

Arguments

`seu`	a Seurat object.
`reduction`	a optional string, dimensional reduction name, 'cofast' by default.
`cluster.name`	an optional string, specify the colname in meta.data for clusters, 'cofast.cluster' by default.
`res`	a positive real, speficy the resolution parameter for Louvain clustering, default as 0.8.
`K`	a positive integer or NULL, specify the number of clusters, default as NULL that indicates not specify the number of clusters.
`res.start`	a positive real, when K is not NULL, starting value of resolution to be searched, default as 0.2.
`res.end`	a positive real, when K is not NULL, ending value of resolution to be searched, default as 2.
`step`	a positive real, when K is not NULL, step size of resolution to be searched, default as 0.02.

Details

None

Value

return a revised Seurat object with a new column in meta.data named cluster.name.

References

None

See Also

None

Examples

```
library(Seurat)
data(pbmc3k_subset)
pbmc3k_subset <- AddCluster(pbmc3k_subset, reduction='ncfm')
head(pbmc3k_subset)
```

`Addcoord2embed` *Add the spatial coordinates to the reduction slot*

Description

Calculate the adjacency matrix given a spatial coordinate matrix with 2-dimension or 3-dimension or more.

Usage

```
Addcoord2embed(seu, coord.name, assay = "RNA")
```

Arguments

`seu`	a SeuratObject with spatial coordinate information in the `meta.data` slot.
`coord.name`	a character vector, specify the names of spatial coordinates in the `meta.data` slot. For example, `c("x", "y")`.
`assay`	a string, specify the assay.

Value

return a revised Seurat object with a slot 'Spatial' in the reductions slot.

References

None

See Also

None

Examples

```
data(CosMx_subset)
library(Seurat)
Addcoord2embed(CosMx_subset, coord.name = c("x", "y"))
```

`AggregationScore` *Calculate the aggregation score for specific clusters*

Description

Calculate the adjacency matrix given a spatial coordinate matrix with 2-dimension or 3-dimension or more.

Usage

```
AggregationScore(seu, reduction.name = "cofast", random.seed = 1)
```

Arguments

`seu`	a SeuratObject with reductions not `NULL`.
`reduction.name`	an character, specify the reduction name for calculating the aggregation score.
`random.seed`	a positive integer, specify the random seed for reproducibility.

Value

return a data.frame with two columns: the first column is the number of spots in each category (cluster/cell type); the second column is the corresponding aggregation score.

References

None

See Also

None

Examples

```
library(Seurat)
data(CosMx_subset)
CosMx_subset <- Addcoord2embed(CosMx_subset, coord.name = c("x", "y"))
Idents(CosMx_subset) <- 'cell_type'

dat.sp.score <- AggregationScore(CosMx_subset, reduction.name = 'Spatial')
print(dat.sp.score)
```

coembed_plot *Coembedding dimensional reduction plot*

Description

Graph output of a dimensional reduction technique on a 2D scatter plot where each point is a cell or feature and it's positioned based on the coembeddings determined by the reduction technique. By default, cells and their signature features are colored by their identity class (can be changed with the group.by parameter).

Usage

```
coembed_plot(
  seu,
  reduction,
  gene_txtdata = NULL,
  cell_label = NULL,
  xy_name = reduction,
  dims = c(1, 2),
  cols = NULL,
  shape_cg = c(1, 5),
  pt_size = 1,
  pt_text_size = 5,
  base_size = 16,
  base_family = "serif",
  legend.point.size = 5,
  legend.key.size = 1.5,
  alpha = 0.3
)
```

Arguments

seu	a Seurat object with coembedding in the reductions slot wiht component name reduction.
reduction	a string, specify the reduction component that denotes coembedding.
gene_txtdata	a data.frame object with columns indcluding 'gene' and 'label', specify the cell type/spatial domain and signature genes. Default as NULL, all features will be used in comebeddings.
cell_label	an optional character in columns of metadata, specify the group of cells/spots. Default as NULL, use Idents as the group.
xy_name	an optional character, specify the names of x and y-axis, default as the same as reduction.
dims	a postive integer vector with length 2, specify the two components for visualization.
cols	an optional string vector, specify the colors for cell group in visualization.

shape_cg	a positive integers with length 2, specify the shapes of cell/spot and feature in plot.
pt_size	an optional integer, specify the point size, default as 1.
pt_text_size	an optional integer, specify the point size of text, default as 5.
base_size	an optional integer, specify the basic size.
base_family	an optional character, specify the font.
legend.point.size	an optional integer, specify the point size of legend.
legend.key.size	an optional integer, specify the size of legend key.
alpha	an optional positive real, range from 0 to 1, specify the transparancy of points.

Details

None

Value

return a ggplot object

References

None

See Also

coembedding_umap

Examples

```
library(Seurat)
data(pbmc3k_subset)
data(top5_signatures)
coembed_plot(pbmc3k_subset, reduction = "UMAPsig",
 gene_txtdata = top5_signatures,  pt_text_size = 3, alpha=0.3)
```

---

coembedding_umap *Calculate UMAP projections for coembedding of cells and features*

Description

Calculate UMAP projections for coembedding of cells and features

Usage

```
coembedding_umap(
  seu,
  reduction,
  reduction.name,
  gene.set = NULL,
  slot = "data",
  assay = "RNA",
  seed = 1
)
```

Arguments

seu	a Seurat object with coembedding in the reductions slot wiht component name reduction.
reduction	a string, specify the reduction component that denotes coembedding.
reduction.name	a string, specify the reduction name for the obtained UMAP projection.
gene.set	a string vector, specify the features (genes) in calculating the UMAP projection, default as all features.
slot	an optional string, specify the slot in the assay, default as 'data'.
assay	an optional string, specify the assay name in the Seurat object when adding the UMAP projection.
seed	an optional integer, specify the random seed for reproducibility.

Details

None

Value

return a revised Seurat object by adding a new reduction component named 'reduction.name'.

References

None

See Also

None

Examples

```
library(Seurat)
data(pbmc3k_subset)
data(top5_signatures)

pbmc3k_subset <- coembedding_umap(
  pbmc3k_subset, reduction = "ncfm", reduction.name = "UMAPsig",
```

```
  gene.set = top5_signatures$gene
)
```

---

coFAST *Cell-feature coembedding for SRT data*

Description

Run cell-feature coembedding for SRT data based on FAST model.

Usage

```
coFAST(
  object,
  Adj_sp,
  assay = NULL,
  slot = "data",
  nfeatures = 2000,
  q = 10,
  reduction.name = "cofast",
  var.features = NULL,
  ...
)
```

Arguments

object	a Seurat object.
Adj_sp	a sparse matrix, specify the adjacency matrix among spots.
assay	an optional string, the name of assay used.
slot	an optional string, the name of slot used.
nfeatures	an optional postive integer, the number of features to select as top variable features. Default is 2000.
q	an optional positive integer, specify the dimension of low dimensional embeddings to compute and store. Default is 10.
reduction.name	an optional string, dimensional reduction name, ‘cofast‘ by default.
var.features	an optional string vector, specify the variable features, used to calculate cell embedding.
...	Other argument passed to the FAST_run.

Value

return a revised Seurat object with a new reduction slot reduction.name obtained by coFAST coembedding, where default reduction.name is ‘cofast‘.

Examples

```
library(Seurat)
data(CosMx_subset)
pos <- as.matrix(CosMx_subset@meta.data[,c("x", "y")])
Adj_sp <- AddAdj(pos)
# Here, we set maxIter = 3 for cofast computation and demonstration.
CosMx_subset <- coFAST(CosMx_subset, Adj_sp = Adj_sp, maxIter=3)
```

CosMx_subset *A CosMix spatial transcriptomics data*

Description

This is a toy CosMix spatial transcriptomics data.

Examples

```
library(Seurat)
data(CosMx_subset)
head(CosMx_subset)
```

---

diagnostic.cor.eigs *Determine the dimension of low dimensional embedding*

Description

This function estimate the dimension of low dimensional embedding for a given cell by gene expression matrix. For more details, see Franklin et al. (1995) and Crawford et al. (2010).

Usage

```
diagnostic.cor.eigs(object, ...)

## Default S3 method:
diagnostic.cor.eigs(
  object,
  q_max = 50,
  plot = TRUE,
  n.sims = 10,
  parallel = TRUE,
  ncores = 10,
  seed = 1,
  ...
)
```

```
## S3 method for class 'Seurat'
diagnostic.cor.eigs(
  object,
  assay = NULL,
  slot = "data",
  nfeatures = 2000,
  q_max = 50,
  seed = 1,
  ...
)
```

Arguments

object	A Seurat or matrix object
...	Other arguments passed to diagnostic.cor.eigs.default.
q_max	the upper bound of low dimensional embedding. Default is 50.
plot	a indicator of whether plot eigen values.
n.sims	number of simulaton times. Default is 10.
parallel	a indicator of whether use parallel analysis.
ncores	the number of cores used in parallel analysis. Default is 10.
seed	a postive integer, specify the random seed for reproducibility
assay	an optional string, specify the name of assay in the Seurat object to be used.
slot	an optional string, specify the name of slot.
nfeatures	an optional integer, specify the number of features to select as top variable features. Default is 2000.

Value

A data.frame with attribute ‘q_est‘ and ‘plot‘, which is the estimated dimension of low dimensional embedding. In addition, this data.frame containing the following components:

- q - The index of eigen values.
- eig_value - The eigen values on observed data.
- eig_sim - The mean value of eigen values of n.sims simulated data.
- q_est - The selected dimension in attr(obj, 'q_est').
- plot - The plot saved in attr(obj, 'plot').

References

1. Franklin, S. B., Gibson, D. J., Robertson, P. A., Pohlmann, J. T., & Fralish, J. S. (1995). Parallel analysis: a method for determining significant principal components. Journal of Vegetation Science, 6(1), 99-106.

2. Crawford, A. V., Green, S. B., Levy, R., Lo, W. J., Scott, L., Svetina, D., & Thompson, M. S. (2010). Evaluation of parallel analysis methods for determining the number of factors.Educational and Psychological Measurement, 70(6), 885-901.

Examples

```
n <- 100
p <- 50
d <- 15
object <- matrix(rnorm(n*d), n, d) %*% matrix(rnorm(d*p), d, p)
diagnostic.cor.eigs(object, n.sims=2)
```

`find.signature.genes` *Find the signature genes for each group of cell/spots*

Description

Find the signature genes for each group of cell/spots based on coembedding distance and expression ratio.

Usage

```
find.signature.genes(
  seu,
  distce.assay = "distce",
  ident = NULL,
  expr.prop.cutoff = 0.1,
  assay = NULL,
  genes.use = NULL
)
```

Arguments

`seu`	a Seurat object with coembedding in the reductions slot wiht component name reduction.
`distce.assay`	an optional character, specify the assay name that constains distance matrix beween cells/spots and features, default as 'distce' (distance of coembeddings).
`ident`	an optional character in columns of metadata, specify the group of cells/spots. Default as NULL, use Idents as the group.
`expr.prop.cutoff`	an optional postive real ranging from 0 to 1, specify cutoff of expression proportion of features, default as 0.1.
`assay`	an optional character, specify the assay in seu, default as NULL, representing the default assay in seu.
`genes.use`	an optional string vector, specify genes as the signature candidates.

Details

In each data.frame object of the returned value, the row.names are gene names, and these genes are sorted by decreasing order of 'distance'. User can define the signature genes as top n genes in distance and that the 'expr.prop' larger than a cutoff. We set the cutoff as 0.1.

Value

return a list with each component a data.frame object having two columns: ‘distance‘ and ‘expr.prop‘.

References

None

See Also

None

Examples

```
library(Seurat)
data(pbmc3k_subset)
pbmc3k_subset <- pdistance(pbmc3k_subset, reduction='ncfm')
df_list_rna <- find.signature.genes(pbmc3k_subset)
```

---

get.top.signature.dat *Obtain the top signature genes and related information*

Description

Obtain the top signature genes and related information.

Usage

```
get.top.signature.dat(df.list, ntop = 5, expr.prop.cutoff = 0.1)
```

Arguments

df.list	a list that is obtained by the function find.signature.genes.
ntop	an optional positive integer, specify the how many top signature genes extracted, default as 5.
expr.prop.cutoff	an optional postive real ranging from 0 to 1, specify cutoff of expression proportion of features, default as 0.1.

Details

Using this funciton, we obtain the top signature genes and organize them into a data.frame. The ‘row.names‘ are gene names. The colname ‘distance‘ means the distance between gene (i.e., VPREB3) and cells with the specific cell type (i.e., B cell), which is calculated based on the coembedding of genes and cells in the coembedding space. The distance is smaller, the association between gene and the cell type is stronger. The colname ‘expr.prop‘ represents the expression proportion of the gene (i.e., VPREB3) within the cell type (i.e., B cell). The colname ‘label‘ means the cell types and colname ‘gene‘ denotes the gene name. By the data.frame object, we know ‘VPREB3‘ is the one of the top signature gene of B cell.

Value

return a ‘data.frame‘ object with four columns: ‘distance‘,‘expr.prop‘, ‘label‘ and ‘gene‘.

References

None

See Also

None

Examples

```
library(Seurat)
data(pbmc3k_subset)
pbmc3k_subset <- pdistance(pbmc3k_subset, reduction='ncfm')
df_list_rna <- find.signature.genes(pbmc3k_subset)
dat.sig <- get.top.signature.dat(df_list_rna, ntop=5)
head(dat.sig)
```

NCFM *Cell-feature coembedding for scRNA-seq data*

Description

Cell-feature coembedding for scRNA-seq data based on FAST model.

Usage

```
NCFM(
  object,
  assay = NULL,
  slot = "data",
  nfeatures = 2000,
  q = 10,
  reduction.name = "ncfm",
  weighted = FALSE,
  var.features = NULL
)
```

Arguments

object	a Seurat object.
assay	an optional string, specify the name of assay in the Seurat object to be used, ‘NULL‘ means default assay in seu.
slot	an optional string, specify the name of slot.

nfeatures	an optional integer, specify the number of features to select as top variable features. Default is 2000.
q	an optional positive integer, specify the dimension of low dimensional embeddings to compute and store. Default is 10.
reduction.name	an optional string, specify the dimensional reduction name, 'ncfm' by default.
weighted	an optional logical value, specify whether use weighted method.
var.features	an optional string vector, specify the variable features used to calculate cell embedding.

Value

return a revised Seurat object with a new reduction slot reduction.name obtained by NCFM coembedding method, where reduction.name is default as 'ncfm'.

Examples

```
data(pbmc3k_subset)
pbmc3k_subset <- NCFM(pbmc3k_subset)
```

---

pbmc3k_subset *A toy single-cell RNA-seq data*

Description

This a toy single-cell RNA-seq data, the subset of PBMC3K.

Examples

```
library(Seurat)
data(pbmc3k_subset)
head(pbmc3k_subset)
```

---

pdistance *Calculate the cell-feature distance matrix*

Description

Calculate the cell-feature distance matrix based on coembeddings.

Usage

```
pdistance(object, reduction = "cofast", assay.name = "distce", eta = 1e-10)
```

Arguments

object	a Seurat object.
reduction	a optional string, dimensional reduction name, ‘cofast‘ by default.
assay.name	a optional string, specify the new generated assay name, ‘distce‘ by default.
eta	an optional positive real, a quantity to avoid numerical errors. 1e-10 by default.

Details

This function calculate the distance matrix between cells/spots and features, and then put the distance matrix in a new generated assay. This distance matrix will be used in the siganture gene identification.

Value

return a revised Seurat object with a assay slot ‘assay.name‘.

Examples

```
data(pbmc3k_subset)
pbmc3k_subset <- NCFM(pbmc3k_subset)
pbmc3k_subset <- pdistance(pbmc3k_subset, "ncfm")
```

---

top5_signatures *A dataframe including top five signature genes*

Description

A dataframe including top five signature genes for each cell type of PBMC3k.

Examples

```
library(Seurat)
data(top5_signatures)
head(top5_signatures)
```

Index

AddAdj, 2
AddCluster, 4
Addcoord2embed, 5
AggregationScore, 6

coembed_plot, 7
coembedding_umap, 8, *8*
coFAST, 10
CosMx_subset, 11

diagnostic.cor.eigs, 11
diagnostic.cor.eigs.default, *12*

FAST_run, *10*
find.signature.genes, 13, *14*

get.top.signature.dat, 14
getAdj_auto, *3*

NCFM, 15

pbmc3k_subset, 16
pdistance, 16

top5_signatures, 17