Package ‘PRECAST’

May 7, 2026

Type Package

Title Embedding and Clustering with Alignment for Spatial Omics Datasets

Version 1.8

Date 2025-12-14

Author Wei Liu [aut, cre],
Yi Yang [aut],
Jin Liu [aut]

Maintainer Wei Liu <`liuweideng@gmail.com`>

Description An efficient data integration method is provided for multiple spatial transcriptomics data with non-cluster-relevant effects such as the complex batch effects. It unifies spatial factor analysis simultaneously with spatial clustering and embedding alignment, requiring only partially shared cell/domain clusters across datasets. More details can be referred to Wei Liu, et al. (2023) <doi:10.1038/s41467-023-35947-w>.

License GPL-3

Depends parallel, gtools, R (>= 4.0.0)

Imports MASS, Matrix, mclust, methods, purrr, utils, Seurat, irlba, cowplot, patchwork, scater, pbapply, ggthemes, dplyr, ggplot2, stats, harmony, DR.SC, scales, ggpubr, graphics, colorspace, Rcpp (>= 1.0.5)

LazyData true

URL `https://github.com/feiyoung/PRECAST`

BugReports `https://github.com/feiyoung/PRECAST/issues`

Suggests knitr, rmarkdown

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-12-15 09:30:02 UTC

Contents

`AddAdjList` *Add adjacency matrix list for a PRECASTObj object*

Description

Add adjacency matrix list for a PRECASTObj object to prepare for PRECAST model fitting.

Usage

```
AddAdjList(PRECASTObj, type="fixed_distance", platform="Visium", ...)
```

Arguments

`PRECASTObj`	a PRECASTObj object created by CreatePRECASTObject.
`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 "Visium". There are more platforms to be chosen, including "Visuim", "ST" and "Other_SRT" ("Other_SRT" represents the other SRT platforms except for 'Visium' and 'ST'), which means there are spatial coordinates information in the metadata of PRECASTObj. The platform helps to calculate the adjacency matrix by defining the neighborhoods when type="fixed_distance" is chosen.
`...`	other arguments to be passed to getAdj, getAdj_auto and getAdj_fixedNumber funciton.

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 revised PRECASTObj object by adding the adjacency matrix list.

Note

nothing

Author(s)

Wei Liu

See Also

AddParSetting.

`AddParSetting` *Add model settings for a PRECASTObj object*

Description

The main interface function provides serveral PRECAST submodels, so a model setting is required to specified in advance for a PRECASTObj object.

Usage

```
AddParSetting(PRECASTObj, ...)
```

Arguments

`PRECASTObj`	a PRECASTObj object created by CreatePRECASTObject.
`...`	other arguments to be passed to model_set funciton.

Details

Nothing

Value

Return a revised PRECASTObj object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
data(PRECASTObj)
PRECASTObj <-AddParSetting(PRECASTObj)
PRECASTObj@parameterList
```

`AddTSNE` *Add tSNE embeddings for a Seurat object*

Description

Run t-SNE dimensionality reduction on selected features.

Usage

```
AddTSNE(seuInt, n_comp=3, reduction='PRECAST', assay='PRE_CAST', seed=1)
```

Arguments

`seuInt`	a Seurat object.
`n_comp`	an optional positive integer, specify the number of features to be extracted.
`reduction`	an optional string, means which dimensional reduction (e.g. PRECAST, PCA) to use for the tSNE. Default is PRECAST.
`assay`	Name of assay that that t-SNE is being run on.
`seed`	an optional integer, the random seed to evaluate tSNE.

Details

Nothing

Value

Return a revised Seurat object by adding tSNE reduction object.

Note

nothing

Author(s)

Wei Liu

See Also

None

`AddUMAP` *Add UMAP embeddings for a Seurat object*

Description

Run UMAP dimensionality reduction on selected features.

Usage

```
AddUMAP(seuInt, n_comp=3, reduction='PRECAST', assay='PRE_CAST', seed=1)
```

Arguments

`seuInt`	a Seurat object.
`n_comp`	an optional positive integer, specify the number of features to be extracted.
`reduction`	an optional string, means which dimensional reduction (e.g. PRECAST, PCA) to use for the UMAP. Default is PRECAST.
`assay`	Name of assay that that t-SNE is being run on.
`seed`	an optional integer, the random seed to evaluate UMAP.

Details

Nothing

Value

Return a revised Seurat object by adding UMAP reduction object.

Note

nothing

Author(s)

Wei Liu

See Also

None

`Add_embed` *Add embeddings for a Seurat object*

Description

Add embeddings for a Seurat object.

Usage

```
Add_embed(embed, seu, embed_name='tSNE' , assay = "RNA")
```

Arguments

`embed`	an embedding matrix.
`seu`	a Seurat object.
`embed_name`	an optional string, the name of embeddings.
`assay`	Name of assay that that embed is being put

Details

Nothing

Value

Return a revised Seurat object by adding a embedding matrix to the Reduc slot in Seurat object.

Note

nothing

Author(s)

Wei Liu

See Also

None

`boxPlot` *Boxplot for a matrix*

Description

Boxplot for a matrix.

Usage

```
boxPlot(mat, ylabel='ARI', cols=NULL, ...)
```

Arguments

`mat`	a matrix with columns.
`ylabel`	an optional string, the name of ylabel.
`cols`	colors used in the plot
`...`	Other parameters passed to geom_boxplot.

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
mat <- matrix(runif(100*3, 0.6, 1), 100, 3)
colnames(mat) <- paste0("Method", 1:3)
boxPlot(mat)
```

`chooseColors` *Choose color schema from a palette*

Description

Choose color schema from a palette

Usage

```
chooseColors(
  palettes_name = c("Nature 10", "Light 13", "Classic 20", "Blink 23", "Hue n"),
  n_colors = 7,
  alpha = 1,
  plot_colors = FALSE
)
```

Arguments

`palettes_name`	a string, the palette name, one of "Nature 10", "Light 13", "Classic 20", "Blink 23" and "Hue n", default as 'Nature 10'.
`n_colors`	a positive integer, the number of colors.
`alpha`	a positive real, the transparency of the color.
`plot_colors`	a logical value, whether plot the selected colors.

Examples

```
chooseColors()
```

`coordinate_rotate` *Coordinates rotation for visualization*

Description

Coordinates rotation for visualization.

Usage

```
coordinate_rotate(pos, theta=0)
```

Arguments

`pos`	a matrix, the n-by-d coordinates, where n is the number of coordinates, d is the dimension of coordinates.
`theta`	a real number, the angle for counter-clock-wise rotation.

Details

Nothing

Value

Return a rotated coordinate matrix.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
x <- 1:100
pos <- cbind(x, sin(pi/4*x))
oldpar <- par(mfrow = c(1,2))
plot(pos)
plot(coordinate_rotate(pos, 40))
par(oldpar)
```

`CreatePRECASTObject` *Create the PRECAST object with preprocessing step.*

Description

Create the PRECAST object with preprocessing step.

Usage

```
CreatePRECASTObject(seuList,  project = "PRECAST",  gene.number=2000,
                    selectGenesMethod='SPARK-X',numCores_sparkx=1,
                    customGenelist=NULL, premin.spots = 20,
                    premin.features=20, postmin.spots=15, postmin.features=15,
                              rawData.preserve=FALSE,verbose=TRUE)
```

Arguments

`seuList`	a list consisting of Seurat objects, where each object is a SRT data batch. The default assay of each Seurat object will be used for data preprocessing and followed model fitting. The specified format about seuList argument can be referred to the details and example.
`project`	An optional string, name of the project, default as "PRECAST".
`gene.number`	an optional integer, the number of top spatially variable genes (SVGs) or highly variable genes (HVGs) to be chosen.
`selectGenesMethod`	an optional integer, the method to select genes for each sample. It supports 'SPARK-X' and 'HVGs' to select genes now. Users can provide self-selected genes using customGenelist argument.
`numCores_sparkx`	an optional integer, specify the number of CPU cores in SPARK package to use when selecting spatial genes.
`customGenelist`	an optional string vector, the list of user specified genes to be used for PRECAST model fitting. If this argument is given, SVGs/HVGs will not be selected.
`premin.spots`	An optional integer, the features (genes) are retained in raw data filtering step with at least premin.spots number of spots, default is 20.
`premin.features`	An optional integer, the locations are retained in raw data filtering step with at least premin.features number of nonzero-count features (genes), default is 20.
`postmin.spots`	An optional integer, the features (genes) are retained in filtering step after common genes selected among all data batches with at least postmin.spots number of spots, default is 15.
`postmin.features`	An optional integer, the locations are retained in filtering step after common genes selected among all data batches with at least postmin.features number of nonzero-count features (genes), default is 15.
`rawData.preserve`	An optional logical value, whether preserve the raw seuList data.
`verbose`	whether display the message in the creating process.

Details

seuList is a list with Seurat object as component, and each Seurat object includes the raw expression count matrix, spatial coordinates and meta data for each data batch, where the spatial coordinates information must be saved in the metadata of Seurat, named "row" and "col" for each data batch.

Value

Returns PRECAST object prepared for PRECAST model fitting. See PRECASTObj-class for more details.

Examples

```
data(PRECASTObj)
library(Seurat)
seuList <- PRECASTObj@seulist
## Check the input of seuList for create PRECAST object.
## Check the default assay for each data batch
lapply(seuList, DefaultAssay)
## Check the spatial coordinates in the meta data named "row" and "col".
head(seuList[[1]]@meta.data)
## Then create PRECAST object using this seuList.
## For convenience, we show the  user-specified genes' list for creating PRECAST object.
## Users can use SVGs from SPARK-X or HVGs.
PRECASTObj2 <- CreatePRECASTObject(seuList,
 customGenelist= row.names(seuList[[1]]), verbose=FALSE)
```

`dimPlot` *Low-dimensional embeddings' plot*

Description

Low-dimensional embeddings' plot colored by a specified meta data in the Seurat object.

Usage

```
dimPlot(seuInt, item=NULL, reduction=NULL, point_size=1,text_size=16,
                    cols=NULL,font_family='', border_col="gray10",
                    fill_col="white", ...)
```

Arguments

`seuInt`	an object named "Seurat".
`item`	the item used for coloring the plot in the meta data of seuInt object.
`reduction`	the reduction used for plot in the seuInt object. If reduction is null, the last added one is used for plotting.
`point_size`	the size of point in the scatter plot.
`text_size`	the text size in the plot.
`cols`	colors used in the plot
`font_family`	the font family used for the plot.
`border_col`	the border color in the plot.
`fill_col`	the color used in backgroup.
`...`	other arguments passed to `plot_scatter`

.

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
data(PRECASTObj)
PRECASTObj <- SelectModel(PRECASTObj)
seuInt <- IntegrateSpaData(PRECASTObj, species='unknown')
dimPlot(seuInt, reduction = 'PRECAST')
## or use the Seurat::DimPlot(seuInt, reduction = 'PRECAST')
```

`doHeatmap` *Heatmap for spots-by-feature matrix*

Description

Plot heatmap for a Seurat object with expressioin data.

Usage

```
doHeatmap(seu, features=NULL, cell_label='Cell type', grp_label = FALSE,
                      pt_size=4, grp_color=NULL, ...)
```

Arguments

`seu`	an object named "Seurat". The object of class "Seurat" must include slot "scale.data".
`features`	an optional string vector, the features to be plotted.
`cell_label`	an optional string, the name of legend.
`grp_label`	an optional logical value, whether display the group names.
`pt_size`	the point size used in the plot
`grp_color`	the colors to use for the group color bar.
`...`	Other paramters passed to DoHeatmap.

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

featurePlot

Examples

```
library(Seurat)
data(PRECASTObj)
PRECASTObj <- SelectModel(PRECASTObj)
seuInt <- IntegrateSpaData(PRECASTObj, species='unknown')
seuInt <- ScaleData(seuInt)
doHeatmap(seuInt, features=row.names(seuInt)[1:5])
```

`drawFigs` *Draw a figure using a group of ggplot objects*

Description

Draw a figure using a group of ggplot objects

Usage

```
drawFigs(
  pList,
  layout.dim = NULL,
  common.legend = FALSE,
  legend.position = "right",
  ...
)
```

Arguments

`pList`	a list with component ggplot objects.
`layout.dim`	a integer vector with length 2, the layout of subplots in rows and columns.
`common.legend`	a logical value, whether use common legend for all subplots.
`legend.position`	a string, the position of legend.
`...`	other arguments that pass to `ggarrange`.

Value

return a new ggplot object.

`featurePlot` *Spatial expression heatmap*

Description

Plot spatial heatmap for a feature of Seurat object with spatial transcriptomics data.

Usage

```
featurePlot(seu, feature=NULL, cols=NULL, pt_size=1, title_size =16, quant=0.5,
 assay='RNA' , reduction="position")
```

Arguments

`seu`	an object named "Seurat". The object of class "Seurat" must include slot "scale.data".
`feature`	an optional string, specify the name of feature to be plotted. If it is null, the first feature will be plotted.
`cols`	colors used in the plot
`pt_size`	the size of point in the spatial heatmap plot.
`title_size`	the title size used for the plot.
`quant`	the quantile value to generate the gradient color map.
`assay`	the assay selected for plot.
`reduction`	the Reduc object for plot.

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
library(Seurat)
data(PRECASTObj)
PRECASTObj <- SelectModel(PRECASTObj)
seuInt <- IntegrateSpaData(PRECASTObj, species='unknown')
seuInt <- ScaleData(seuInt)
featurePlot(seuInt, assay='PRE_CAST')
```

`firstup` *Set the first letter of a string vector to captial*

Description

Set the first letter of a string vector to captial.

Usage

```
firstup(x)
```

Arguments

`x`	a string vector.

Details

Nothing

Value

Return a string vector with first letter capital.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
x <- c("good", "Morning")
firstup(x)
```

`getAdj_fixedNumber` *Calculate adjacency matrix by user-specified number of neighbors*

Description

an efficient function to find the neighborhood based on the matrix of position and a user-specified number of neighbors of each spot.

Usage

```
getAdj_fixedNumber(pos, number=6)
```

Arguments

`pos`	is a n-by-d matrix of position, where n is the number of spots, and d is the dimension of coordinates.
`number`	is the number of neighbors of each spot. Euclidean distance to decide whether a spot is an neighborhood of another spot.

Value

A sparse matrix containing the neighbourhood.

See Also

getAdj, getAdj_auto.

`getAdj_reg` *Calculate adjacency matrix for regular spatial coordinates.*

Description

Calculate adjacency matrix for regular spatial coordinates from ST or Visium platform.

Usage

```
getAdj_reg(pos, platform= "Visium")
```

Arguments

`pos`	is a n-by-d matrix of position, where n is the number of spots, and d is the dimension of coordinates.
`platform`	a string, specify the platform of the provided data, default as "Visium", and only support "ST" and "Visium" platform.

Value

A sparse matrix containing the neighbourhood.

See Also

getAdj, getAdj_auto, getAdj_fixedNumber.

`getneighborhood_fast` *getneighborhood_fast*

Description

an efficient function to find the neighborhood based on the matrix of position and a pre-defined cutoff

Usage

```
getneighborhood_fast(x, radius)
```

Arguments

`x`	is a n-by-2 matrix of position.
`radius`	is a threashold of Euclidean distance to decide whether a spot is an neighborhood of another spot. For example, if the Euclidean distance between spot A and B is less than cutoff, then A is taken as the neighbourhood of B.

Value

A sparse matrix containing the neighbourhood

`Human_HK_genes` *Human housekeeping genes database*

Description

Human housekeeping genes database.

Details

This data is a data.frame and include the Human housekeeping genes information in the columns named "Gene" and "Ensembl".

`ICM.EM` *ICM-EM algorithm implementation*

Description

ICM-EM algorithm for fitting PRECAST model

Usage

```
ICM.EM(XList, q, K, AdjList=NULL,  Adjlist_car=NULL,
       posList = NULL, platform = "ST",
       beta_grid=seq(0.2,4, by=0.2), init.nstart=5,
       maxIter_ICM=6,maxIter=20, epsLogLik=1e-5, verbose=TRUE,
       mix_prop_heter=TRUE, Sigma_equal=FALSE, Sigma_diag=TRUE,
       error_heter=TRUE, Sp2=TRUE,
       wpca_int=FALSE,int.model=c('kmeans', 'mclust'),
       seed=1,coreNum = 1, coreNum_int=coreNum)
```

Arguments

`XList`	an M-length list consisting of multiple matrices with class `dgCMatrix` or `matrix` that specify the log-normalization gene expression matrix for each data sample used for PRECAST model.
`q`	a positive integer, specify the number of latent features to be extracted, default as 15.
`K`	a positive integer allowing scalar or vector, specify the number of clusters in model fitting.
`AdjList`	an M-length list of sparse matrices with class `dgCMatrix`, specify the adjacency matrix used for Potts model in PRECAST. We provide this interface for those users who would like to define the adjacency matrix by their own.

`Adjlist_car`	an M-length list of sparse matrices with class `dgCMatrix`, specify the adjacency matrix used for CAR model in PRECAST, default as AdjList in the Potts model. We provide this interface for those users who would like to use the different adjacency matrix in CAR model.
`posList`	an M-length list composed by spatial coordinate matrix for each data sample.
`platform`	a string, specify the platform of the provided data, default as "Visium". There are many platforms to be supported, including ("Visuim", "ST", "SeqFISH", 'merFISH', 'slide-seqv2', 'seqscope', "HDST"). If AdjList is not given, the The platform helps to calculate the adjacency matrix by defining the neighbors.
`beta_grid`	an optional vector of positive value, the candidate set of the smoothing parameter to be searched by the grid-search optimization approach.
`init.nstart`	an optional positive integer, how many times to be run for obtaining the initial values, default as 5.
`maxIter_ICM`	an optional positive value, represents the maximum iterations of ICM.
`maxIter`	an optional positive value, represents the maximum iterations of EM.
`epsLogLik`	an optional positive vlaue, tolerance vlaue of relative variation rate of the observed pseudo log-loglikelihood value, defualt as '1e-5'.
`verbose`	an optional logical value, whether output the information of the ICM-EM algorithm.
`mix_prop_heter`	an optional logical value, specify whether betar are distict, default as `TRUE`.
`Sigma_equal`	an optional logical value, specify whether Sigmaks are equal, default as FALSE.
`Sigma_diag`	an optional logical value, specify whether Sigmaks are diagonal matrices, default as `TRUE`.
`error_heter`	an optional logical value, whether use the heterogenous error for DR-SC model, default as `TRUE`. If `error_heter=FALSE`, then the homogenuous error is used for probabilistic PCA model in PRECAST.
`Sp2`	an optional logical value, whether add the ICAR model component in the model, default as TRUE. We provide this interface for those users who don't want to include the ICAR model.
`wpca_int`	an optional logical value, means whether use the weighted PCA to obtain the initial values of loadings and other paramters, default as `FALSE` which means the ordinary PCA is used.
`int.model`	an optional string, specify which model is used in evaluting the initial values for PRECAST; "kmeans" and "mclust" are supported, default as "kmeans".
`seed`	an optional integer, the random seed in fitting PRECAST model.
`coreNum`	an optional positive integer, means the number of thread used in parallel computating.
`coreNum_int`	an optional positive integer, means the number of cores used in parallel computation for initial values when `K` is a vector, default as same as `coreNum`.

Details

Nothing

Value

ICM.EM returns a list with class "SeqKiDRSC_Object" with the number of components equal to the length of `K`, where each component includes the model fitting results for one number of cluster and is a list consisting of following components:

`cluster`	an M-length list that includes the inferred class labels for each data sample.
`hZ`	an M-length list that includes the batch corrected low-dimensional embeddings for each data sample.
`hV`	an M-length list that includes the estimate the ICAR component for each sample.
`Rf`	an M-length list that includes the posterior probability of domain clusters for each sample.
`beta`	an M-length vector that includes the estimated smoothing parameters for each sample.
`Mu`	mean vectors of mixtures components.
`Sigma`	covariance matrix of mixtures components.
`W`	estimated loading matrix
`Lam`	estimated variance of errors in probabilistic PCA model
`loglik`	pseudo observed log-likelihood.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Liao, X., Luo, Z. et al, Jin Liu* (2023). Probabilistic embedding, clustering, and alignment for integrating spatial transcriptomics data with PRECAST. Nature Communications, 14, 296

See Also

None

Examples

```
## we generate the spatial transcriptomics data with lattice neighborhood, i.e. ST platform.
library(Matrix)
q <- 10; K <- 4
data(PRECASTObj)
posList <- lapply(PRECASTObj@seulist, function(x) cbind(x$row, x$col))
AdjList <- lapply(posList, getAdj_reg, platform='ST')
XList <- lapply(PRECASTObj@seulist, function(x) t(x[['RNA']]@data))
XList <- lapply(XList, scale, scale=FALSE)
## For illustration, maxIter is set to 4
```

```
resList <- ICM.EM(XList,AdjList = AdjList, maxIter=4,
                  q=q, K=K, verbose=TRUE)
```

`ICM.EM_structure` *ICM-EM algorithm implementation with organized paramters*

Description

Efficient data integration as well as spatial clustering for multiple spatial transcriptomics data

Usage

```
ICM.EM_structure(XList,  K, AdjList, q=15,parameterList=NULL)
```

Arguments

`XList`	an M-length list consisting of multiple matrices with class `dgCMatrix` or `matrix` that specify the log-normalization gene expression matrix for each data sample used for PRECAST model.
`K`	a positive integer allowing scalar or vector, specify the number of clusters in model fitting.
`AdjList`	an M-length list of sparse matrices with class `dgCMatrix`, specify the adjacency matrix used for Potts model and Intrisic CAR model in PRECAST model. We provide this interface for those users who would like to define the adjacency matrix by their own.
`q`	a positive integer, specify the number of latent features to be extracted, default as 15.
`parameterList`	Other arguments in PRECAST model, it can be set by model_set.

Details

Nothing

Value

ICM.EM_structure returns a list with class "SeqK_PRECAST_Object" with the number of components equal to the length of `K`, where each component includes the model fitting results for one number of cluster and is a list consisting of following components:

`cluster`	an M-length list that includes the inferred class labels for each data sample.
`hZ`	an M-length list that includes the batch corrected low-dimensional embeddings for each data sample.
`hV`	an M-length list that includes the estimate the ICAR component for each sample.
`Rf`	an M-length list that includes the posterior probability of domain clusters for each sample.

`beta`	an M-length vector that includes the estimated smoothing parameters for each sample.
`Mu`	mean vectors of mixtures components.
`Sigma`	covariance matrix of mixtures components.
`W`	estimated loading matrix
`Lam`	estimated variance of errors in probabilistic PCA model
`loglik`	pseudo observed log-likelihood.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Liao, X., Luo, Z. et al, Jin Liu* (2023). Probabilistic embedding, clustering, and alignment for integrating spatial transcriptomics data with PRECAST. Nature Communications, 14, 296

See Also

None

Examples

```
## we generate the spatial transcriptomics data with lattice neighborhood, i.e. ST platform.
 library(Matrix)
 q <- 10; K <- 4
 data(PRECASTObj)
 posList <- lapply(PRECASTObj@seulist, function(x) cbind(x$row, x$col))
 AdjList <- lapply(posList, getAdj_reg, platform='ST')
 XList <- lapply(PRECASTObj@seulist, function(x) t(x[['RNA']]@data))
 XList <- lapply(XList, scale, scale=FALSE)
 parList <- model_set(maxIter=4)
 resList <- ICM.EM_structure(XList,  AdjList = AdjList,
                  q=q, K=K, parameterList=parList)
```

`IntegrateSpaData` *Integrate multiple SRT data*

Description

Integrate multiple SRT data based on the PRECASTObj by PRECAST model fitting.

Usage

```
IntegrateSpaData(PRECASTObj, species="Human",
                 custom_housekeep=NULL, covariates_use=NULL,
                 seuList=NULL, subsample_rate=1, sample_seed=1)
```

Arguments

`PRECASTObj`	a PRECASTObj object after finishing the PRECAST model fitting and model selection.
`species`	an optional string, one of 'Human', 'Mouse' and 'Unknown', specify the species of the SRT data to help choose the housekeeping genes. 'Unknown' means only using the PRECAST results reconstruct the alligned gene expression.
`custom_housekeep`	user-specified housekeeping genes.
`covariates_use`	a string vector, the colnames in ‘PRECASTObj@seulist[[1]]@meta.data‘, representing other biological covariates to considered when removing batch effects. This is achieved by adding additional covariates for biological conditions in the regression, such as case or control. Default as ‘NULL‘, denoting no other covariates to be considered.
`seuList`	an optional Seurat list object, ‘seuList‘ plays a crucial role in the integration process. If ‘seuList‘ is set to ‘NULL‘ and ‘PRECASTObj@seuList‘ is not NULL, then ‘seuList‘ will adopt the values of ‘PRECASTObj@seuList‘. Subsequently, the genes within ‘seuList‘ will be utilized for integration. Conversely, if ‘seuList‘ is not NULL, the integration will directly employ the genes specified within ‘seuList‘. In the event that both ‘seuList‘ and ‘PRECASTObj@seuList‘ are set to NULL, integration will proceed using the genes outlined in ‘PRECASTObj@seulist‘, i.e., the variable genes. To preserve the ‘seuList‘ not NULL in ‘PRECASTObj@seuList‘, user can set ‘rawData.preserve=TRUE‘ when running ‘CreatePRECASTObject‘. This parameter empowers users to integrate the entire set of genes in ‘seuList‘ when implementing the integration, as opposed to exclusively considering the variable genes within ‘PRECASTObj@seuList‘.
`subsample_rate`	an optional real number ranging from zero to one, this parameter specifies the subsampling rate during integration to enhance computational efficiency, default as 1 (without subsampling).
`sample_seed`	an optional integer, with a default value of 1, serves to designate the random seed when ‘subsample_rate‘ is set to a value less than one, ensuring reproducibility in the sampling process.

Details

Nothing

Value

Return a Seurat object by integrating all SRT data batches into a SRT data, where the column "batch" in the meta.data represents the batch ID, and the column "cluster" represents the clusters obtained by PRECAST.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Liao, X., Luo, Z. et al, Jin Liu* (2023). Probabilistic embedding, clustering, and alignment for integrating spatial transcriptomics data with PRECAST. Nature Communications, 14, 296

Gagnon-Bartsch, J. A., Jacob, L., & Speed, T. P. (2013). Removing unwanted variation from high dimensional data with negative controls. Berkeley: Tech Reports from Dep Stat Univ California, 1-112.

See Also

None

Examples

```
data(PRECASTObj)
PRECASTObj <- SelectModel(PRECASTObj)
seuInt <- IntegrateSpaData(PRECASTObj, species='unknown')
```

`model_set` *PRECAST model setting*

Description

Set the PRECAST model structure and paramters in the algorithm.

Usage

```
model_set(Sigma_equal=FALSE, Sigma_diag=TRUE,mix_prop_heter=TRUE,
                      error_heter=TRUE, Sp2=TRUE, wpca_int=FALSE,
                      int.model= c('mclust', 'kmeans'),
                      coreNum = 1, coreNum_int=coreNum,
                      beta_grid=seq(0.2,4, by=0.2),init.nstart=5,
                maxIter_ICM=6,maxIter=20, epsLogLik=1e-5, verbose=TRUE, seed=1)
```

Arguments

`Sigma_equal`	an optional logical value, specify whether Sigmaks are equal, default as FALSE.
`Sigma_diag`	an optional logical value, specify whether Sigmaks are diagonal matrices, default as `TRUE`.
`mix_prop_heter`	an optional logical value, specify whether betar are distict, default as `TRUE`.
`error_heter`	an optional logical value, whether use the heterogenous error i.e. lambdarj != lambdark for each sample r, default as `TRUE`. If `error_heter=FALSE`, then the homogenuous error is used for probabilistic PCA model.
`Sp2`	an optional logical value, whether add the ICAR model component in the model, default as TRUE. We provide this interface for those users who don't want to include the ICAR model.
`wpca_int`	an optional logical value, means whether use the weighted PCA to obtain the initial values of loadings and other paramters, default as `FALSE` which means the ordinary PCA is used.
`int.model`	an optional string, specify which model is used in evaluting the initial values for PRECAST; "kmeans" and "mclust" are supported, default as "mclust".
`init.nstart`	an optional positive integer, how many times to be run for obtaining the initial values, default as 5.
`coreNum`	an optional positive integer, means the number of thread used in parallel computating.
`coreNum_int`	an optional positive integer, means the number of cores used in parallel computation for initial values when `K` is a vector, default as same as `coreNum`.
`beta_grid`	an optional vector of positive value, the candidate set of the smoothing parameter to be searched by the grid-search optimization approach.
`maxIter_ICM`	an optional positive value, represents the maximum iterations of ICM.
`maxIter`	an optional positive value, represents the maximum iterations of EM.
`epsLogLik`	an optional positive vlaue, tolerance vlaue of relative variation rate of the observed pseudo log-loglikelihood value, defualt as '1e-5'.
`verbose`	an optional logical value, whether output the information of the ICM-EM algorithm.
`seed`	an optional integer, the random seed in fitting PRECAST model.

Details

Nothing

Value

Return a list including all paramters' setting.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
model_set()
```

`Mouse_HK_genes` *Mouse housekeeping genes database*

Description

Mouse housekeeping genes database.

Details

This data is a data.frame and include the mouse housekeeping genes information in the columns named "Gene" and "Ensembl".

`plot_RGB` *Spatial RGB heatmap*

Description

Plot spatial RGB heatmap.

Usage

```
plot_RGB(position, embed_3d, pointsize=2,textsize=15)
```

Arguments

`position`	a coordinates matrix with two columns: x-coordinate and y-coordinate.
`embed_3d`	a embedding matrix with three columns: x, y and z embeddings.
`pointsize`	the size of point in the scatter plot.
`textsize`	the text size in the plot.

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

None

`plot_scatter` *Scatter plot for two-dimensional embeddings*

Description

Scatter plot for two-dimensional embeddings

Usage

```
plot_scatter(embed_use, meta_data, label_name,
  xy_names=c('tSNE1', 'tSNE2'), no_guides = FALSE,
  cols = NULL,
  point_size = 0.5, point_alpha=1,
  base_size = 12, do_points = TRUE, do_density = FALSE, border_col='gray',
  legend_pos='right', legend_dir='vertical', nrow.legend=NULL)
```

Arguments

`embed_use`	an object named "Seurat", "maxtrix" or "dgCMatrix". The object of class "Seurat" must include slot "scale.data".
`meta_data`	an optional positive integer, specify the number of features to be extracted.
`label_name`	the size of point in the scatter plot.
`xy_names`	the text size in the plot.
`no_guides`	whether display the legend.
`cols`	colors used in the plot.
`point_size`	the point size of scatter plot.
`point_alpha`	the transparency of the plot.
`base_size`	the base text size.
`do_points`	Plot point plot.
`do_density`	Plot density plot
`border_col`	the border color in the plot.
`legend_pos`	the position of legend.
`legend_dir`	the direction of legend.
`nrow.legend`	the number of rows of legend.

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
embed_use <- cbind(1:100, sin((1:100)*pi/2))
meta_data <- data.frame(cluster=factor(rep(1:2, each=50)))
plot_scatter(embed_use, meta_data, label_name='cluster')
```

`PRECAST` *Fit a PRECAST model*

Description

Fit a PRECAST model.

Usage

```
PRECAST(PRECASTObj, K=NULL, q= 15)
```

Arguments

`PRECASTObj`	an object named "PRECASTObj". The object PRECASTObj is created by CreatePRECASTObject.
`K`	An optional integer or integer vector, specify the candidates of number of clusters. if `K=NULL`, it will be set to 4~12.
`q`	An optional integer, specify the number of low-dimensional embeddings to extract in PRECAST.

Details

The model fitting results are saved in the slot of resList.

Value

Return a revised PRECASTObj object.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Liao, X., Luo, Z. et al, Jin Liu* (2023). Probabilistic embedding, clustering, and alignment for integrating spatial transcriptomics data with PRECAST. Nature Communications, 14, 296

See Also

None

`PRECASTObj` *A simple PRECASTObj for example*

Description

A simple PRECASTObj for example.

Details

This PRECASTObj include the basic slots in PRECAST object; see PRECASTObj-class for more details.

`PRECASTObj-class` *Each PRECASTObj object has a number of slots which store information.*

Description

Each PRECASTObj object has a number of slots which store information. Key slots to access are listed below.

Slots

`seuList` A list with Seurat object as component, representing the raw expression count matrix, spatial coordinates and meta data for each data batch, where the spatial coordinates information is saved in the metadata of Seurat, named "row" and "col" for eahc data batch.

`seulist` A Seurat list after the preprocessing step in preparation for PRECAST model.

`AdjList` The adjacency matrix list for a PRECASTObj object.

`parameterList` The model parameter settings for a PRECASTObj object

`resList` The results after fitting PRECAST models.

`project` Name of the project.

`selectIntFeatures` *Select common genes for multiple data batches*

Description

selectIntFeatures prioritizes genes based on the number of times they were selected as HVGs/SVGs in all data baches, and chose the top genes as the input for the analysis. We broke ties by examining the ranks of the tied genes in each original dataset and taking those with the highest median rank.

Usage

```
selectIntFeatures(seulist, spaFeatureList, IntFeatures=2000)
```

Arguments

`seulist`	a list consisting of Seurat objects, where each object is a SRT data batch.
`spaFeatureList`	an list consisting of SVGs vectors, where each vector is the top HVGs/SVGs for each SRT data batch.
`IntFeatures`	the number of common HVGs/SVGs genes to be chosen.

Details

Nothing

Value

Return a string vector, the selected gene list for integration in PRECAST.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Liao, X., Luo, Z. et al, Jin Liu* (2023). Probabilistic embedding, clustering, and alignment for integrating spatial transcriptomics data with PRECAST. Nature Communications, 14, 296

See Also

None

`SelectModel` *Select best PRECAST model from candidated models*

Description

Select best PRECAST model from candidated models with different number of clusters.

Usage

```
  ## S3 method for class 'SeqK_PRECAST_Object'
SelectModel(obj, criteria = 'MBIC',pen_const=1, return_para_est=FALSE)
  ## S3 method for class 'PRECASTObj'
SelectModel(obj, criteria = 'MBIC',pen_const=1, return_para_est=FALSE)
```

Arguments

`obj`	a SeqK_PRECAST_Object or PRECASTObj object after PRECAST model fitting.
`criteria`	a string, specify the criteria used for selecting the number of clusters, supporting "MBIC", "BIC" and "AIC".
`pen_const`	an optional positive value, the adjusted constant used in the MBIC criteria.
`return_para_est`	an optional logical value, whether return the other paramters' estimators in PRECAST.

Details

Nothing

Value

Return a revised PRECASTObj object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
data(PRECASTObj)
PRECASTObj <- SelectModel(PRECASTObj)
```

SpaPlot *Spatial heatmap*

Description

Plot spatial heatmap for a Seurat object with spatial transcriptomics data.

Usage

```
SpaPlot(seuInt, batch=NULL, item=NULL, point_size=2,text_size=12,
                  cols=NULL,font_family='', border_col="gray10",
                  fill_col='white', ncol=2, combine = TRUE,
                  title_name="Sample", ...)
```

Arguments

`seuInt`	an object named "Seurat".
`batch`	an optional positive integer or integer vector, specify the batches to be extracted. Users can check the batches' names by `unique(seuInt$batch)`.
`item`	an optional string, which column is plotted in the meta data of seuInt. Users can check the meta data by `head(seuInt@meta.data)`. If `item` takes value from ("RGB_UMAP", "RGB_tSNE"), this function will plot the RGB plot.
`point_size`	the size of point in the scatter plot.
`text_size`	the text size in the plot.
`cols`	colors used in the plot
`font_family`	the font family used for the plot, default as Times New Roman.
`border_col`	the border color in the plot.
`fill_col`	the color used in backgroup.
`ncol`	the number of columns in the layout of plots.
`combine`	an optional logical value, whether plot all on a figure. If TRUE, all figures are plotted; otherwise, return a list with each plot as component.
`title_name`	an optional string, title name in the plot.
`...`	other arguments passed to `plot_scatter`
.	

Details

Nothing

Value

Return a ggplot2 object or list of ggplots objects.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```
data(PRECASTObj)
PRECASTObj <- SelectModel(PRECASTObj)
seuInt <- IntegrateSpaData(PRECASTObj, species='unknown')
SpaPlot(seuInt)
```

`sp_means_Rcpp` *Calculate column-wise or row-wise mean*

Description

Calculate column-wise or row-wise mean

Usage

```
sp_means_Rcpp(sp_data, rowMeans = FALSE)
```

Arguments

`sp_data`	A sparse matrix
`rowMeans`	A boolean value, whether to calculate row-wise mean

Value

A n x 1 or p x 1 matrix

`sp_sums_Rcpp` *Calculate column-wise or row-wise sum*

Description

Calculate column-wise or row-wise sum

Usage

```
sp_sums_Rcpp(sp_data, rowSums = FALSE)
```

Arguments

`sp_data`	A sparse matrix
`rowSums`	A boolean value, whether to calculate row-wise sum

Value

A n x 1 or p x 1 matrix

`volinPlot` *Volin/boxplot plot*

Description

Plot volin/boxplot.

Usage

```
  volinPlot(mat, ylabel='ARI', cols=NULL)
```

Arguments

`mat`	a matrix with columns.
`ylabel`	an optional string, the name of ylabel.
`cols`	colors used in the plot

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

See Also

None

Examples

```
mat <- matrix(runif(100*3, 0.6, 1), 100, 3)
colnames(mat) <- paste0("Method", 1:3)
volinPlot(mat)
```

Index