Package ‘PRECAST’

July 12, 2022

Type Package

Title Embedding and Clustering with Alignment for Spatial Datasets

Version 1.2

Date 2022-07-10

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

Maintainer Wei Liu <wei.liu@duke-nus.edu.sg>

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. (2022) <doi:10.1101/2022.06.26.497672>.

License GPL-3

Depends parallel, gtools, R (>= 4.0.0)

Imports GiRaF, MASS, Matrix, mclust, methods, purrr, utils, Seurat, cowplot, scater, pbapply, patchwork, ggthemes, dplyr, ggplot2, stats, DR.SC, scales, 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

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-07-11 23:30:09 UTC
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", ...)
AddParSetting

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 many platforms to be supported, including ("Visuim", "ST", "SeqFISH", 'merFISH', 'slide-seqv2', 'seqscope', "HDST"), 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 function.

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, ...)
AddTSNE

Arguments

PRECASTobj  a PRECASTObj object created by \texttt{CreatePRECASTObject}.

...  other arguments to be passed to \texttt{model_set} function.

Details

Nothing

Value

Return a revised PRECASTObj object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

\begin{verbatim}
data(PRECASTObj)
PRECASTObj <- AddParSetting(PRECASTObj)
PRECASTObj@parameterList
\end{verbatim}

\begin{verbatim}
AddTSNE(seuInt, n\_comp=3, reduction='PRECAST', assay='PRE\_CAST', seed=1)
\end{verbatim}

Description

Run t-SNE dimensionality reduction on selected features.

Usage

\begin{verbatim}
AddTSNE(seuInt, n\_comp=3, reduction='PRECAST', assay='PRE\_CAST', seed=1)
\end{verbatim}

Arguments

\begin{verbatim}
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.
\end{verbatim}
AddUMAP

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

```r
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.
Add_embeddings for a Seurat object

Add_embed

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
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)
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", numCores_sparkx = 1, 
  gene.number = 2000, customGenelist = NULL, premin.spots = 20, 
  premin.features = 20, postmin.spots = 15, postmin.features = 15, 
  rawData.perserve = FALSE, verbose = TRUE)

Arguments

seuList  
a list consisting of Seurat objects, where each object is a SRT data batch.

project  
An optional string, name of the project, default as "PRECAST".

numCores_sparkx  
an optional integer, specify the number of CPU cores in SPARK package to use 
when selecting spatial genes.

gene.number  
an optional integer, the number of top spatially variable genes (SVGs) to be 
chosen.

customGenelist  
an optional string vector, the list of user specified genes to be used for PRECAST 
model fitting. If this argument is given, SVGs 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 com-
mon genes selected among all data batches with at least premin.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 premin.features number of 
nonzero-count features (genes), default is 15.

rawData.perserve  
An optional logical value, whether preserve the raw seuList data.

verbose  
whether display the message in the creating process.

Value

Returns PRECAST object prepared for PRECAST model fitting.
Examples

```r
data(PRECASTObj)
seuList <- PRECASTObj@seulist
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**

```r
dimPlot(seuInt, item=NULL, reduction=NULL, point_size=1, text_size=16,
cols=NULL, font_family='quotesingle.Var/quotesingle.Var', 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 background.

**Details**

Nothing

**Value**

Return a ggplot2 object.

**Note**

nothing
doHeatmap

Author(s)
Wei Liu

See Also
None

Examples

data(PRECASTObj)
PRECASTObj <- selectModel.PRECASTObj(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 expression data.

**Usage**

```r
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 parameters 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])

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.
firstup

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```r
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

```r
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

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

---

gendata_seulist  Generate simulated data

Description

Generate simulated spatial transcriptomics data with two data batches.

Usage

```r
gendata_seulist(height1=30, width1=30, height2=height1, width2=width1,
p =100, q=10, K=7, G=4, beta=1.2, sigma2=1,
alpha=8, seed=1, view=FALSE)
```

Arguments

- `height1`, `width1`, `height2`, `width2`
  Height and width of lattice grids for generating spatial coordinates, n=height * width spots for expression matrix data
- `p`
  number of genes to generate.
- `q`
  number of true latent features to generate gene expression
- `K`
  number of clusters (spatial domains/cell types).
- `G`
  the number of neighbors. The latter must be one of G = 4 or G = 8, which respectively correspond to a first order and a second order dependency structure. By default, G = 4.
- `beta`
  the smoothing parameter in Potts model.
- `sigma2`
  Variance of error term in probabilistic PCA model.
getAdj_fixedNumber

alpha  a positive factor of mixture mean values.
seed   random seed for generate data
view   Logical value indicating whether the draw should be printed. Do not display the optional borders.

Details

Nothing

Value

return a list consisting of Seurat objects, where each object is a SRT data batch, and the metadata of this Seurat object will include two columns with names "row" and "col" which are the spatial coordinates.

Note

nothing

Author(s)

Wei Liu

References

None

See Also

None

Examples

```r
## we generate two spatial transcriptomics data
seulist <- gendata_seulist(height1=20, width1=20, p=200, K=4)
seulist
```

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_auto, getAdj.
**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**

```r
ICM.EM(XList, q, K, AdjList=NULL, Adjlist_car=NULL, posList = NULL,
        platform = "ST", beta_grid=seq(0.2,4, by=0.2),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=’EEE’, 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 iDR-SC 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 iDR-SC. 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 iDR-SC, 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 ("Visum", "ST", "SeqFISH", "merFISH", 'slide-seqv2', 'seqscope', "HDST"). If AdjList is not given, 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.

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 value, tolerance value of relative variation rate of the observed pseudo log-loglikelihood value, default 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 beta_r are distinct, 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 iDR-SC.

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 parameters, default as FALSE which means the ordinary PCA is used.

int.model an optional string, specify which Gaussian mixture model is used in evaluating the initial values for PRECAST, default as "EEE"; and see Mclust for more models’ names.

seed an optional integer, the random seed in fitting PRECAST model.

coreNum an optional positive integer, means the number of thread used in parallel computing.

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.
ICM.EM

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, Xu Liao, Yi Yang, Huazhen Lin, Joe Yeong, Xiang Zhou, Xingjie Shi and Jin Liu. (2022) Joint dimension reduction and clustering analysis for single-cell RNA-seq and spatial transcriptomics data

See Also
None

Examples

```R
## 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

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.
Description

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

Usage

IntegrateSpaData(PRECASTObj, species="Human", custom_housekeep=NULL)
Arguments

PRECASTObj  a PRECASTObj object after finishing the PRECAST model fitting and model selection.

species  an optional string, specify the species of the SRT data to help choose the housekeeping genes.

custom_housekeep  user-specified housekeeping genes.

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

See Also

None

Examples

```r
data(PRECASTObj)
PRECASTObj <- selectModel.PRECASTObj(PRECASTObj)
seuInt <- IntegrateSpaData(PRECASTObj, species='unknown')
```

Description

Set the PRECAST model structure and parameters in the algorithm.
Usage

```r
model_set(Sigma_equal=FALSE, Sigma_diag=TRUE, mix_prop_heter=TRUE, error_heter=TRUE, Sp2=TRUE, wpca_int=FALSE, int.model="EEE", coreNum = 1, coreNum_int=coreNum, beta_grid=seq(0.2,4, by=0.2), 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 distinct, 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 homogenous 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 parameters, default as `FALSE` which means the ordinary PCA is used.
- **int.model**: an optional string, specify which Gaussian mixture model is used in evaluating the initial values for PRECAST, default as "EEE"; and see Mclust for more models’ names.
- **coreNum**: an optional positive integer, means the number of threads used in parallel computing.
- **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 values, 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 value, tolerance value of relative variation rate of the observed pseudo log-loglikelihood value, default 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 parameters’ 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)
plot_scatter

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

```r
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')
```
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.

Details

Nothing

Value

Return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

See Also

None

Examples

```r
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')
```
Description

Fit a PRECAST model.

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRECASTObj</td>
<td>an object named &quot;PRECASTObj&quot;. The object PRECASTObj is created by CreatePRECASTObject.</td>
</tr>
<tr>
<td>K</td>
<td>An optional integer or integer vector, specify the candidates of number of clusters. if K=NULL, it will be set to 4~12.</td>
</tr>
<tr>
<td>q</td>
<td>An optional integer, specify the number of low-dimensional embeddings to extract in PRECAST.</td>
</tr>
</tbody>
</table>

Details

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

Value

Return a revised PRECASTObj object.

Note

nothing

Author(s)

Wei Liu

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 each 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 SVGs in all data batches, 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 SVGs for each SRT data batch obtained by SPARK or SPARK-X.

IntFeatures the number of common 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

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
```r
## 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 parameters' estimators in PRECAST.

Details
Nothing

Value
Return a revised PRECASTObj object.

Note
nothing

Author(s)
Wei Liu

See Also
None
SpaPlot

Examples

```r
data(PRECASTObj)
PRECASTObj <- selectModel(PRECASTObj)
```

---

**SpaPlot**

*Spatial heatmap*

**Description**

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

**Usage**

```r
SpaPlot(seuInt, batch=NULL, item=NULL, point_size=2, text_size=16,
        cols=NULL, font_family='', border_col="gray10",
        fill_col='white', ncol=2, combine = TRUE, title_name="Sample")
```

**Arguments**

- **seuInt**: an object named "Seurat", "maxtrix" or "dgCMatrix". The object of class "Seurat" must include slot "scale.data".
- **batch**: an optional positive integer, specify the number of features to be extracted.
- **item**: an optional string, which column is plotted in the meta data of seuInt.
- **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.
- **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.

**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)

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

Add_embed, 6
AddAdjList, 2
AddParSetting, 3, 3
AddTSNE, 4
AddUMAP, 5

boxPlot, 7
coordinate_rotate, 8
CreatePRECASTObject, 3, 4, 9, 27
data.frame, 17, 24
dimPlot, 10
DoHeatmap, 11
doHeatmap, 11

featurePlot, 12, 12
firstup, 13
gndata_seulist, 14
getAdj, 3, 16
getAdj_auto, 3, 16
getAdj_fixedNumber, 3, 15, 16
getAdj_reg, 16

Human_HK_genes, 17
ICM.EM, 17
ICM.EM_structure, 20
IntegrateSpaData, 21

list, 18, 20, 24
model_set, 4, 20, 22
Mouse_HK_genes, 24

plot_RGB, 24
plot_scatter, 25
PRECAST, 27
PRECASTObj, 28
PRECASTObj-class, 28, 28

selectIntFeatures, 29
selectModel, 30
SpaPlot, 31
volinPlot, 32