Package 'maple'

March 14, 2022

Title Bayesian Analysis of Multi-Sample Spatial Transcriptomics Experiments

Version 0.99.5

Description Allows for robust probabilistic analysis of multi-sample spatial transcriptomics experiments (Allen et. al, 2021 <doi:10.1101/2021.06.23.449615>).

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.2

Imports spruce, Seurat, stats, dbarts, ggplot2, dplyr, ggalluvial, plotly, shiny, rlang

Depends R (>= 4.0.0)

NeedsCompilation no

Author Carter Allen [aut, cre] (<`https://orcid.org/0000-0001-6937-7234`>), Dongjun Chung [aut]

Maintainer Carter Allen <`carter.allen12@gmail.com`>

Repository CRAN

Date/Publication 2022-03-14 19:00:02 UTC

R topics documented:

`fit_maple` *Fit Maple multi-sample Bayesian spatial mixture model*

Description

This function allows you to detect sub-populations and explain membership with relevant covariates in multi-sample spatial transcriptomics experiments.

Usage

```
fit_maple(
  seurat_obj,
  K,
  emb = "PCs",
  n_dim = 8,
  covars = NULL,
  MCAR = FALSE,
  CAR = FALSE,
  smooth = TRUE,
  r = 3,
  nsim = 2000,
  burn = 1000,
  z_init = NULL
)
```

Arguments

`seurat_obj` An integrated Seurat object.

`K` The number of sub-populations to infer. Each should be present in each sample.

`emb` The cell spot embedding to use. Either one of "PCs", "scGNN", "harmony", "HVGs", or "SVGs".

`n_dim` The number of dimensions to use.

`covars` Column names of Seurat meta data to use as covariates. If none specified, will fit a global intercept and sample-indicator model for cell type membership probabilities.

`MCAR` Logical. Include multivariate CAR random intercepts in gene expression model?

`CAR` Logical. Include univariate CAR random intercepts in multinomial gene expression model?

`smooth` Logical. Use manual spatial smoothing controlled by r parameter?

`r` Spatial smoothing parameter for if smooth == TRUE. Should be greater than 0 with larger values enforcing stronger prior spatial association.

`nsim` Number of total MCMC iterations to conduct.

`burn` Number of initial MCMC iterations to discard as burn in. The number of saved iterations is nsim-burn.

`z_init` Initialized cluster allocation vector to aid in MCMC convergence. If NULL z_init will be set using hierarchical clustering.

Value

A list of MCMC samples, including the MAP estimate of cluster indicators (z)

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
brain <- RunPCA(brain)
brain_fit_PCs <- fit_maple(brain,K = 6,emb = "PCs")

## End(Not run)
```

`get_maple_scores` *Get posterior probability scores*

Description

This function allows you to compute posterior uncertainty and continuous phenotype scores

Usage

```
get_maple_scores(fit)
```

Arguments

`fit`	A list returned by fit_maple()

Value

A fit object (list)

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
```

```
brain <- RunPCA(brain)
brain_fit_PCs <- fit_maple(brain,K = 6,emb = "PCs")
brain_fit_scores <- get_maple_scores(brain_fit_PCs)

## End(Not run)
```

maple_viz *Plot tissue architecture labels*

Description

This function allows you to plot (static or interactive) cell spot labels and uncertainty measures

Usage

```
maple_viz(
  fit,
  pt.size = 1,
  interactive = FALSE,
  shade_uncertainty = FALSE,
  feature = NULL
)
```

Arguments

`fit` A list returned by fit_maple()

`pt.size` The size of each cell spot point

`interactive` Logical parameter controlling static or interactive nature of plot

`shade_uncertainty`
Logical parameter for shading of cell spots by posterior uncertainty. Must run get_maple_scores() first.

`feature` A user-provided feature (e.g., gene of interest) to visualize over tissue spaces instead of sub-population labels.

Value

A ggplot object or shiny app window

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
```

```
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
brain <- RunPCA(brain)
brain_fit_PCs <- fit_maple(brain,K = 6,emb = "PCs")
maple_viz(brain_fit_PCs)

## End(Not run)
```

`offset_images` *Offset multiple Seurat images*

Description

Internal function for defining coordinates with from multi-sample Seurat objects

Usage

```
offset_images(seurat_obj)
```

Arguments

`seurat_obj`	A Seurat object

Value

A coordinate data frame

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
brain <- RunPCA(brain)
brain <- offset_images(brain)

## End(Not run)
```

plot_props_alluvial *Plot grouped alluvial plots of cell type proportions*

Description

This function allows you to visualize the relative abundance of sub-populations after running fit_maple()

Usage

```
plot_props_alluvial(fit, group)
```

Arguments

`fit`	A list returned by fit_maple()
`group`	A column name of fit$W or a grouping vector of length N (nrow(fit$W))

Value

A ggplot object

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
brain <- RunPCA(brain)
brain_fit_PCs <- fit_maple(brain,K = 6,emb = "PCs")
plot_props_alluvial(brain_fit_PCs, group = brain$orig.ident)

## End(Not run)
```

plot_props_grouped *Plot grouped bar charts of cell type proportions*

Description

This function allows you to visualize the relative abundance of sub-populations after running fit_maple()

Usage

```
plot_props_grouped(fit, group)
```

Arguments

`fit`	A list returned by fit_maple()
`group`	A column name of fit$W or a grouping vector of length N (nrow(fit$W))

Value

A ggplot object

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
brain <- RunPCA(brain)
brain_fit_PCs <- fit_maple(brain,K = 6,emb = "PCs")
plot_props_grouped(brain_fit_PCs, group = brain$orig.ident)

## End(Not run)
```

`plot_props_pie` *Plot grouped pie charts of cell type proportions*

Description

This function allows you to visualize the relative abundance of sub-populations after running fit_maple()

Usage

```
plot_props_pie(fit, group)
```

Arguments

`fit`	A list returned by fit_maple()
`group`	A column name of fit$W or a grouping vector of length N (nrow(fit$W))

Value

A ggplot object

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
brain <- RunPCA(brain)
brain_fit_PCs <- fit_maple(brain,K = 6,emb = "PCs")
plot_props_pie(brain_fit_PCs, group = brain$orig.ident)

## End(Not run)
```

plot_props_stacked *Plot stacked bar charts of cell types proportions*

Description

This function allows you to visualize the relative abundance of sub-populations after running fit_maple()

Usage

```
plot_props_stacked(fit, group)
```

Arguments

fit	A list returned by fit_maple()
group	A column name of fit$W or a grouping vector of length N (nrow(fit$W))

Value

A ggplot object

Examples

```
## Not run:
brain1 <- LoadData("stxBrain", type = "anterior1")
brain2 <- LoadData("stxBrain", type = "anterior2")
brain1 <- SCTransform(brain1, assay = "Spatial", verbose = FALSE)
brain2 <- SCTransform(brain2, assay = "Spatial", verbose = FALSE)
brain <- merge(brain1,brain2)
DefaultAssay(brain) <- "SCT"
VariableFeatures(brain) <- c(VariableFeatures(brain1),VariableFeatures(brain2))
brain <- RunPCA(brain)
brain_fit_PCs <- fit_maple(brain,K = 6,emb = "PCs")
plot_props_stacked(brain_fit_PCs, group = brain$orig.ident)

## End(Not run)
```

Index