Package ‘retrofit’

May 22, 2024

Title RETROFIT: Reference-free deconvolution of cell mixtures in spatial transcriptomics

Version 1.4.0

Description RETROFIT is a Bayesian non-negative matrix factorization framework to decompose cell type mixtures in ST data without using external single-cell expression references. RETROFIT outperforms existing reference-based methods in estimating cell type proportions and reconstructing gene expressions in simulations with varying spot size and sample heterogeneity, irrespective of the quality or availability of the single-cell reference. RETROFIT recapitulates known cell-type localization patterns in a Slide-seq dataset of mouse cerebellum without using any single-cell data.

biocViews Transcriptomics, Visualization, RNASeq, Bayesian, Spatial, Software, GeneExpression, DimensionReduction, FeatureExtraction, SingleCell

License GPL-3

Encoding UTF-8

LazyData FALSE

URL https://github.com/qunhualilab/retrofit

BugReports https://github.com/qunhualilab/retrofit/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 4.2), Rcpp

LinkingTo Rcpp

Suggests BiocStyle, knitr, rmarkdown, testthat, DescTools, ggplot2, corrplot, cowplot, grid, colorspace, png, reshape2, pals, RCurl

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/retrofit
git_branch RELEASE_3_19
git_last_commit 7d32b7d
git_last_commit_date 2024-04-30
annotateWithCorrelations

**Repository**  Bioconductor 3.19

**Date/Publication**  2024-05-21

**Author**  Adam Park [aut, cre],
Roopali Singh [aut] (<https://orcid.org/0000-0001-6539-6622>),
Xiang Zhu [aut] (<https://orcid.org/0000-0003-1134-6413>),
Qunhua Li [aut] (<https://orcid.org/0000-0003-0675-7648>)

**Maintainer**  Adam Park <akp6031@psu.edu>

**Contents**

annotateWithCorrelations ........................................... 2
annotateWithMarkers .................................................. 3
decompose ............................................................. 4
retrofit ................................................................. 6
testSimulationData ................................................... 7
vignetteColonData ...................................................... 8
vignetteSimulationData ................................................. 8

**Index** 9

---

annotateWithCorrelations

*RETROFIT matching algorithm*

---

**Description**

Match cell types based on correlations with reference. decomp_w between matching algorithm description

**Usage**

annotateWithCorrelations(sc_ref, K, decomp_w, decomp_h)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sc_ref</td>
<td>A Matrix or Array with two dimensions (GeneExpressions, Cell types).</td>
</tr>
<tr>
<td>K</td>
<td>integer: The number of cell types to be selected</td>
</tr>
<tr>
<td>decomp_w</td>
<td>Matrix(GeneExpressions, Components): Decomposed w matrix</td>
</tr>
<tr>
<td>decomp_h</td>
<td>Matrix(Components, Spots): Decomposed h matrix</td>
</tr>
</tbody>
</table>
### Value

A list of selected components, cells, and correlations

- **w**: Filtered 2d array with GeneExpressions, Cell types
- **h**: Filtered 2d array with Cell types, Spots
- **ranked_cells**: The list of cell names
- **ranked_correlations**: The list of correlations

### See Also

papers reference

### Examples

```r
data("testSimulationData")
K = 10
sc_ref = testSimulationData$sc_ref
W = testSimulationData$decompose$w
H = testSimulationData$decompose$h
result = retrofit::annotateWithCorrelations(sc_ref=sc_ref, K=K,
                                          decomp_w=W, decomp_h=H)
H_annotated = result$h
W_annotated = result$w
ranked_cells = result$ranked_cells```

---

### Description

Match cell types based on correlations with reference. decomp_w between matching algorithm description

### Usage

`annotateWithMarkers(marker_ref, K, decomp_w, decomp_h)`

### Arguments

- **marker_ref**: Key-value list: A dictionary of key: cell type, value: GeneExpression list
- **K**: integer: The number of cell types to be selected
- **decomp_w**: Matrix(GeneExpressions, Components): Decomposed w matrix
- **decomp_h**: Matrix(Components, Spots): Decomposed h matrix
Value

A list of

• w
• h

See Also

papers reference

Examples

data("testSimulationData")
K = 10
marker_ref = testSimulationData$marker_ref
W = testSimulationData$decompose$w
H = testSimulationData$decompose$h

result = retrofit::annotateWithMarkers(marker_ref=marker_ref, K=K,
                                        decomp_w=W, decomp_h=H)

H_annotated = result$h
W_annotated = result$w
ranked_cells = result$ranked_cells

decompose  RETROFIT decomposition algorithm

Description

Receiving the input with 2d spatial transcriptomics matrix, the function returns factorized W, H, Theta. This function fulfills Structured Stochastic Variational Inference Algorithm for RETROFIT. Since exact Bayesian inference is infeasible and considering the large number of spots and genes, variational inference was adopted to approximately estimate the parameters in performant manner.

Usage

decompose(
    x,
    L = 16,
    iterations = 4000,
    init_param = NULL,
    lambda = 0.01,
    kappa = 0.5,
    verbose = FALSE
)
Arguments

x  
matrix or array with dimension (GeneExpressions, Spots). This is the main spatial transcriptomics data.

L  
integer (default:16) The number of components to be decomposed

iterations  
integer (default:4000) The number of maximum iterations to be executed

init_param  
list Vatirational initial parameters

lambda  
double (default:0.01) Background expression profile control

kappa  
double (default:0.5) Learning rate factor

verbose  
boolean (default:FALSE)

Details

init_param specification

• alpha_w_0 double (default:0.05)
• beta_w_0 double (default:0.0001)
• alpha_h_0 double (default:0.2)
• beta_h_0 double (default:0.2)
• alpha_th_0 double (default:1.25)
• beta_th_0 double (default:10)
• alpha_th_k array (default:array with dim c(K))
• beta_th_k array (default:array with dim c(K)),
• alpha_w_gk array (default:array with dim c(G,K)),
• beta_w_gk array (default:array with dim c(G,K)),
• alpha_h_ks array (default:array with dim c(K,S)),
• beta_h_ks array (default:array with dim c(K,S))

Value

A list of decomposed vectors that contains

• w: 2d array with GeneExpressions, Components
• h: 2d array with Components, Spots
• th: an array with Components
• durations: (verbose) durations vector (unit: second)
• relative_error:(verbose) errors with pre-defined norm vector

See Also

papers reference
Examples

```r
data("testSimulationData")
x = testSimulationData$extra5_x
res = retrofit::decompose(x, L=16, iterations=10, verbose=TRUE)
W = res$w
H = res$h
TH = res$th
```

---

**Description**

The main algorithm

**Usage**

```r
retrofit(
  x,
  sc_ref = NULL,
  marker_ref = NULL,
  L = 16,
  K = 8,
  iterations = 4000,
  init_param = NULL,
  lambda = 0.01,
  kappa = 0.5,
  verbose = FALSE
)
```

**Arguments**

- `x`: A matrix or array with dimension (GeneExpressions, Spots). This is the main spatial transcriptomics data.
- `sc_ref`: A matrix or array with two dimensions (GeneExpressions, Cell types).
- `marker_ref`: A list with (keys, values) = (cell types, an array of genes).
- `L`: integer (default:16) The number of components to be decomposed
- `K`: integer: The number of cell types to be selected
- `iterations`: integer (default:4000) The number of maximum iterations to be executed
- `init_param`: list Variational initial parameters
- `lambda`: double (default:0.01) Background expression profile control
- `kappa`: double (default:0.5) Learning rate factor
- `verbose`: boolean (default:FALSE)
Value

A list of decomposed vectors that contains

- **decompose:**
  - `w`: Decomposed 2d array with GeneExpressions, Components
  - `h`: Decomposed 2d array with Components, Spots
  - `th`: 1d array with Components

- **annotateWithCorrelations:**
  - `w`: Filtered 2d array with GeneExpressions, Cell types
  - `h`: Filtered 2d array with Cell types, Spots

- **annotateWithMarkers:**
  - `w`: Filtered 2d array with GeneExpressions, Cell types
  - `h`: Filtered 2d array with Cell types, Spots

See Also

papers reference

Examples

data("testSimulationData")
iterations = 10
L = 16
K = 8
x = testSimulationData$extra5_x
sc_ref = testSimulationData$sc_ref
res = retrofit::retrofit(x, sc_ref=sc_ref, L=L, K=K, iterations=iterations)
W = res$decompose$w
W_annotated = res$annotateWithCorrelations$w
ranked_cells= res$annotateWithCorrelations$ranked_cells

---

testSimulationData simulation data

Description

A dataset with input and output of retrofit functions for reproducibility tests.

Usage

data(testSimulationData)

Format

Includes input x, references and results with large iterations
vignetteSimulationData

Details

• vignetteColonData

vignetteColonData  colon vignette

Description

A dataset supporting the colon vignette process

Usage

data(vignetteColonData)

Format

Includes colon scenario x, references, a large iterations results.

Details

• vignetteColonData

vignetteSimulationData

simulation vignette

Description

A dataset supporting the simulation vignette process

Usage

data(vignetteSimulationData)

Format

Includes n10m3 scenario x, references, a large iterations results.

Details

• vignetteSimulationData
Index

* datasets
  testSimulationData, 7
  vignetteColonData, 8
  vignetteSimulationData, 8
annotateWithCorrelations, 2
annotateWithMarkers, 3
decompose, 4
retrofit, 6
testSimulationData, 7
vignetteColonData, 8
vignetteSimulationData, 8