

# Package ‘retrofit’

May 2, 2026

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

**Version** 1.13.0

**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** devel

**git\_last\_commit** ca4aee1

**git\_last\_commit\_date** 2026-04-28

**Repository** Bioconductor 3.24

**Date/Publication** 2026-05-01

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

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

## Contents

annotateWithCorrelations . . . . .	2
annotateWithMarkers . . . . .	3
decompose . . . . .	4
retrofit . . . . .	5
testSimulationData . . . . .	7
vignetteColonData . . . . .	7
vignetteSimulationData . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

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

sc_ref	A Matrix or Array with two dimensions (GeneExpressions, Cell types).
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 selected components, cells, and correlations

- w: Filtered 2d array with GeneExpressions, Cell types
- h: Filtered2d array with Cell types, Spots
- ranked\_cells: The list of cell names
- ranked\_correlations: The list of correlations

### See Also

papers reference

### Examples

```

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
    
```

annotateWithMarkers     *RETROFIT matching algorithm*

### 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 Vairational 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

```
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
```

---

retrofit

*RETROFIT*

---

### Description

The main algorithm

### Usage

```
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 Vairational 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: Filtered2d array with Cell types, Spots
- annotateWithMarkers:
  - w: Filtered 2d array with GeneExpressions, Cell types
  - h: Filtered2d 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	<i>simulation data</i>
--------------------	------------------------

---

**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

**Details**

- testSimulationData

---

vignetteColonData	<i>colon vignette</i>
-------------------	-----------------------

---

**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, [7](#)

vignetteSimulationData, [8](#)

annotateWithCorrelations, [2](#)

annotateWithMarkers, [3](#)

decompose, [4](#)

retrofit, [5](#)

testSimulationData, [7](#)

vignetteColonData, [7](#)

vignetteSimulationData, [8](#)