

Package: DDRTree2 (via r-universe)

June 8, 2026

Type Package

Title Learning Principal Graphs with DDRTree (Performance-Optimized Fork)

Version 0.2.0

Date 2025-10-03

Author Xiaojie Qiu [aut], Cole Trapnell [aut], Qi Mao [aut], Li Wang [aut], Billsfriend [cre, ctb] (OpenMP parallelization and performance optimizations)

Maintainer Billsfriend <Billsfriend1999@outlook.com>

Depends irlba

Imports Rcpp

LinkingTo Rcpp, RcppEigen, BH

Description Provides an implementation of the framework of reversed graph embedding (RGE) which projects data into a reduced dimensional space while constructing a principal tree which passes through the middle of the data simultaneously. DDRTree shows superiority to alternatives (Wishbone, DPT) for inferring the ordering as well as the intrinsic structure of single cell genomics data. In general, it could be used to reconstruct the temporal progression as well as bifurcation structure of any datatype. This version includes OpenMP parallelization for significant performance improvements (11x speedup on typical datasets).

License Artistic-2.0 | file LICENSE

URL <https://github.com/Billsfriend/DDRTree2>

BugReports <https://github.com/Billsfriend/DDRTree2/issues>

RoxygenNote 7.3.3

Encoding UTF-8

SystemRequirements OpenMP (optional, for parallel execution)

NeedsCompilation yes

Note This is a performance-optimized fork of the original CRAN package (v0.1.5) by Xiaojie Qiu et al. The original algorithm and core functionality remain unchanged. Modifications include OpenMP parallelization and code optimizations for improved performance on large datasets.

Repository <https://billsfriend.r-universe.dev>

Date/Publication 2025-10-11 09:24:07 UTC

RemoteUrl <https://github.com/billsfriend/DDRTree2>

RemoteRef HEAD

RemoteSha f30bc84052d4950ec192d7f3976121dbc7f9303f

Contents

DDRTree	2
get_major_eigenvalue	4
pca_projection_R	4
sqdist_R	5
Index	6

DDRTree	<i>Perform DDRTree construction</i>
---------	-------------------------------------

Description

Perform DDRTree construction

Usage

```
DDRTree(
  X,
  dimensions = 2,
  initial_method = NULL,
  maxIter = 20,
  sigma = 0.001,
  lambda = NULL,
  ncenter = NULL,
  param.gamma = 10,
  tol = 0.001,
  verbose = F,
  ...
)
```

Arguments

X	a matrix with $D \times N$ dimension which is needed to perform DDRTree construction
dimensions	reduced dimension
initial_method	a function to take the data transpose of X as input and then output the reduced dimension, row number should not larger than observation and column number should not be larger than variables (like isomap may only return matrix on valid sample sets). Sample names of returned reduced dimension should be preserved.
maxIter	maximum iterations
sigma	bandwidth parameter
lambda	regularization parameter for inverse graph embedding
ncenter	number of nodes allowed in the regularization graph
param.gamma	regularization parameter for k-means (the prefix of 'param' is used to avoid name collision with gamma)
tol	relative objective difference
verbose	emit extensive debug output
...	additional arguments passed to DDRTree

Value

a list with W, Z, stree, Y, history W is the orthogonal set of d (dimensions) linear basis vector Z is the reduced dimension space stree is the smooth tree graph embedded in the low dimension space Y represents latent points as the center of Z

Examples

```
data('iris')
subset_iris_mat <- as.matrix(t(iris[c(1, 2, 52, 103), 1:4])) #subset the data
#run DDRTree with ncenters equal to species number
DDRTree_res <- DDRTree(subset_iris_mat, dimensions = 2, maxIter = 5, sigma = 1e-2,
lambda = 1, ncenter = 3, param.gamma = 10, tol = 1e-2, verbose = FALSE)
Z <- DDRTree_res$Z #obtain matrix
Y <- DDRTree_res$Y
stree <- DDRTree_res$stree
plot(Z[1, ], Z[2, ], col = iris[c(1, 2, 52, 103), 'Species']) #reduced dimension
legend("center", legend = unique(iris[c(1, 2, 52, 103), 'Species']), cex=0.8,
col=unique(iris[c(1, 2, 52, 103), 'Species']), pch = 1) #legend
title(main="DDRTree reduced dimension", col.main="red", font.main=4)
dev.off()
plot(Y[1, ], Y[2, ], col = 'blue', pch = 17) #center of the Z
title(main="DDRTree smooth principal curves", col.main="red", font.main=4)

#run DDRTree with ncenters equal to species number
DDRTree_res <- DDRTree(subset_iris_mat, dimensions = 2, maxIter = 5, sigma = 1e-3,
lambda = 1, ncenter = NULL,param.gamma = 10, tol = 1e-2, verbose = FALSE)
Z <- DDRTree_res$Z #obtain matrix
Y <- DDRTree_res$Y
```

```
stree <- DDRTree_res$stree
plot(Z[1, ], Z[2, ], col = iris[c(1, 2, 52, 103), 'Species']) #reduced dimension
legend("center", legend = unique(iris[c(1, 2, 52, 103), 'Species']), cex=0.8,
col=unique(iris[c(1, 2, 52, 103), 'Species']), pch = 1) #legend
title(main="DDRTree reduced dimension", col.main="red", font.main=4)
dev.off()
plot(Y[1, ], Y[2, ], col = 'blue', pch = 2) #center of the Z
title(main="DDRTree smooth principal graphs", col.main="red", font.main=4)
```

get_major_eigenvalue *Get the top L eigenvalues*

Description

Get the top L eigenvalues

Usage

```
get_major_eigenvalue(C, L)
```

Arguments

C	data matrix used for eigendecomposition
L	number for the top eigenvalues

pca_projection_R *Compute the PCA projection*

Description

Compute the PCA projection

Usage

```
pca_projection_R(C, L)
```

Arguments

C	data matrix used for PCA projection
L	number for the top principal components

sqdist_R	<i>calculate the square distance between a, b</i>
----------	---

Description

calculate the square distance between a, b

Usage

```
sqdist_R(a, b)
```

Arguments

a	a matrix with $D \times N$ dimension
b	a matrix with $D \times N$ dimension

Value

a numeric value for the different between a and b

Index

DDRTree, [2](#)

get_major_eigenvalue, [4](#)

pca_projection_R, [4](#)

sqdist_R, [5](#)