Package ‘bcTSNE’

December 1, 2021

Type Package

Title Projected t-SNE for Batch Correction

Version 0.11.1

Maintainer Dayne L Filer <dayne.filer@gmail.com>

Description Implements the projected t-SNE method for batch correction of high-dimensional data. Please see Aliverti et al. (2020) <doi:10.1093/bioinformatics/btaa189> for more information.

Imports stats, RSpectra, utils, Rtsne, graphics, splatter

Suggests data.table, batchelor, kBET, scater, knitr, lisi, harmony, dlfUtils, xtable

VignetteBuilder knitr

SystemRequirements GNU make

License GPL-3

Encoding UTF-8

URL https://github.com/emanuelealiverti/BC_tSNE

RoxygenNote 7.1.2

Additional_repositories https://daynefiler.github.io/drat

NeedsCompilation yes

Author Dayne L Filer [aut, cre],
Emanuele Aliverti [aut],
Jeff Tilson [aut],
Kirk C Wilhelmsen [aut],
David B Dunson [aut]

Repository CRAN

Date/Publication 2021-12-01 07:40:05 UTC

R topics documented:

`apat` *A + t(A)*

Description

A + t(A)

Usage

```
apat(A)
```

Arguments

`A`	numeric matrix

Details

Not exported; exists for testing C code.

Value

numeric matrix (A + t(A))

bctsne *Calculate BC t-SNE by orthogonal gradient descent*

Description

Calculate BC t-SNE by orthogonal gradient descent

Usage

```
bctsne(X, Z, k = 50, outDim = 2, perplexity = 30, maxIter = 1000)
```

Arguments

`X`	numeric matrix, input matrix
`Z`	numeric matrix, covariate matrix
`k`	integer of length 1, reduced dimension (number of eigenvectors)
`outDim`	integer of length 1, the output dimension
`perplexity`	numeric of length 1, the t-SNE perplexity
`maxIter`	integer of length 1, the maximum iterations for the BC t-SNE algorithm

Details

`X` should be preprocessed (e.g. PCA, centered and scaled). `Z` is the full model matrix, excluding the intercept.

Value

`list` wth the following items:

`Xred` numeric matrix, the reduced dimension input to `bctsne`

`Z` model matrix indicating batch membership

`perplexity` perpelexity value used in computing t-SNE

`Y` batch-corrected projection matrix

`maxIter` maximum iterations used in training

Examples

```
## Create small simulated dataset, A, with embeded batch effects
set.seed(2731)
kRid <- 20
p    <- 100
n    <- 200

W <- matrix(rnorm(p*kRid), kRid)
S <- matrix(rnorm(n*kRid), n)
z <- sample(1:3, rep = TRUE, size = n)
```

```
Z <- model.matrix( ~ -1 + as.factor(z))
l <- matrix(rnorm(kRid*NCOL(Z)), kRid)
A <- (S - Z %*% t(l) ) %*% W

## Scale A to give input, X
X <- scale(A)

resUnadj <- Rtsne::Rtsne(X)                ## Standard t-SNE
resAdj   <- bctsne(X = X, Z = Z, k = 10)    ## Batch-corrected t-SNE

## Plot results, no true effects were included in the simulated data, so
## we expect all batches to overlap with bcTSNE; batch membership indicated
## by color
plot(resUnadj$Y, col = z)
plot(resAdj$Y, col = z)
```

`calcPvals` *Calculate t-SNE p-values based on a distance matrix*

Description

Calculate t-SNE p-values based on a distance matrix

Usage

```
calcPvals(D, perplexity = 30)
```

Arguments

`D`	numeric matrix, distance matrix
`perplexity`	numeric of length 1, t-SNE perplexity

Details

Not exported; exists for testing C code.

Value

numeric matrix of p-values based on the given perplexity

grad *Calculate t-SNE gradient*

Description

Calculate t-SNE gradient

Usage

```
grad(Y, pval, Z)
```

Arguments

`Y`	numeric matrix, lower dimension embedding
`pval`	numeric matrix, input data p-values
`Z`	numeric covariate matrix

Details

Not exported; exists for testing C code.

Value

numeric matrix, t-SNE gradient

ols *Ordinary least squares, solves B = AX for X.*

Description

Ordinary least squares, solves B = AX for X.

Usage

```
ols(A, B)
```

Arguments

`A`	numeric matrix
`B`	numeric matrix

Details

Not exported; exists for testing C code.

Value

numeric matrix (X)

sqdist *Calculate squared Euclidean distance*

Description

Calculate squared Euclidean distance

Usage

```
sqdist(X)
```

Arguments

`X`	numeric matrix

Details

Not exported; exists for testing C code.

Value

numeric squared distance matrix

ssx *Sum of squares*

Description

Sum of squares

Usage

```
ssx(X)
```

Arguments

`X`	numeric matrix

Details

Not exported; exists for testing C code.

Value

vector with the row sum of squares

`zeroMean` *Subtract the column means from X*

Description

Subtract the column means from X

Usage

```
zeroMean(X)
```

Arguments

`X`	numeric matrix

Details

Not exported; exists for testing C code.

Value

numeric matrix with column means subtracted

Index