\textbf{R topics documented:}

- \texttt{apat} ................................................................. 2
- \texttt{betsne} ................................................................. 3
- \texttt{calcPvals} ............................................................... 4
- \texttt{grad} ................................................................. 5
- \texttt{ols} ................................................................. 5
- \texttt{sqdist} ............................................................... 6
- \texttt{ssx} ................................................................. 6
- \texttt{zeroMean} ............................................................ 7

\textbf{Index} \\

\begin{tabular}{ll}
\texttt{apat} & \texttt{A + t(A)} \\
\end{tabular}

\textbf{Description}

\texttt{A + t(A)}

\textbf{Usage}

\texttt{apat(A)}

\textbf{Arguments}

\texttt{A} \hspace{1em} \texttt{numeric matrix}

\textbf{Details}

Not exported; exists for testing C code.

\textbf{Value}

\texttt{numeric matrix (A + t(A))}
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 with the following items:

- Xred: numeric matrix, the reduced dimension input to bctsne
- Z: model matrix indicating batch membership
- perplexity: perplexity value used in computing t-SNE
- Y: batch-corrected projection matrix
- maxIter: maximum iterations used in training

Examples

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

W <- matrix(rnorm(p*kRid), kRid)
S <- matrix(rnorm(n*kRid), kRid)
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**  
\[ \text{sqdist}(X) \]

**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**  
\[ \text{ssx}(X) \]

**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

apat, 2
bctsne, 3
calcPvals, 4
grad, 5
ols, 5
sqdist, 6
ssx, 6
zeroMean, 7