Package 'Canek'

July 18, 2026

Type Package

Title Batch Correction of Single Cell Transcriptome Data

Version 0.3.1

Description Non-linear/linear hybrid method for batch-effect correction that uses Mutual Nearest Neighbors (MNNs) to identify similar cells between datasets. Reference: Loza M. et al. (NAR Genomics and Bioinformatics, 2020) <doi:10.1093/nargab/lqac022>.

Depends R (>= 3.5.0)

Imports FNN, irlba, numbers, fpc, bluster, igraph, matrixStats, methods, parallel, utils

Suggests testthat (>= 2.1.0), Seurat, SingleCellExperiment, SummarizedExperiment, scater, batchelor, scran, knitr, rmarkdown, patchwork, ggplot2

License MIT + file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

URL https://martinloza.github.io/Canek/

BugReports https://github.com/MartinLoza/Canek/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Martin Loza [aut, cre] (ORCID: <https://orcid.org/0000-0002-3342-2643>), Diego Diez [aut] (ORCID: <https://orcid.org/0000-0002-2325-4893>)

Maintainer Martin Loza <martin.loza.lopez@gmail.com>

Repository CRAN

Date/Publication 2026-07-17 23:50:01 UTC

Contents

CheckZeroCV 2
CorrectBatch 3
CorrectBatches 5
EkfBE 8
Fuzzy 9
MeanBE 10
MedianBE 10
PairsFiltering 11
RunCanek 12
SimBatches 13

Index **15**

`CheckZeroCV` *CheckZeroCV*

Description

CheckZeroCV

Usage

```
CheckZeroCV(
  MST = NULL,
  cluMem = NULL,
  corGene = NULL,
  fuzzyPCA = fuzzyPCA,
  memCorrData = NULL,
  zeroCorrection = NULL
)
```

Arguments

`MST`	Minimum Spanning Tree
`cluMem`	Clusters used on MST
`corGene`	Data to correct
`fuzzyPCA`	Number of PCs to use in the fuzzy process.
`memCorrData`	Data to correct
`zeroCorrection`	Vector indicating which membership has a zero correction vector

CorrectBatch *CorrectBatch*

Description

Batch effect correction on two single-cell batches

Usage

```
CorrectBatch(
  refBatch,
  queBatch,
  cnRef = NULL,
  cnQue = NULL,
  queNumCelltypes = NULL,
  maxMem = 5,
  pairs = NULL,
  kNN = 30,
  ncores = 1,
  sampling = FALSE,
  numSamples = NULL,
  idxQuery = NULL,
  idxRef = NULL,
  pcaDim = 50,
  perCellMNN = 0.08,
  fuzzy = TRUE,
  fuzzyPCA = 10,
  estMethod = "Median",
  clusterMethod = "louvain",
  pairsFilter = FALSE,
  doCosNorm = FALSE,
  correctEmbeddings = FALSE,
  maxLoop = 1,
  loopTol = 0.001,
  verbose = FALSE
)
```

Arguments

refBatch	Reference batch.
queBatch	Query batch (batch to correct).
cnRef	Cosine normalization of the reference batch.
cnQue	Cosine normalization of the query batch.
queNumCelltypes	Number of cell types in the query batch. By default Canek searches the number of cell types using an heuristic algorithm. Change this parameter if you know the number of cell types in advanced.

maxMem — Maximum number of memberships from the query batch. This parameter is used on the heuristic algorithm to find the number of cell types.

pairs — A numerical matrix containing MNNs pairs cell indexes. First column corresponds to query batch cell indexes.

kNN — Number of k-nearest-neighbors used to define the MNNs pairs.

ncores — Number of cores to use for finding MNN pairs. Defaults to 1 (sequential). Requesting more than 1 errors on Windows (no fork() support there) or if it exceeds the number of cores parallel::detectCores() reports, rather than silently falling back or reducing.

sampling — Use MNNs pairs sampling when using a Kalman filter to estimate the correction vector.

numSamples — If sampling. Number of MNNs pairs samples to use on the estimation process.

idxQuery — Numerical vector indicating the index of the cells from the query batch to use on the correction vector estimation.

idxRef — Numerical vector indicating the index of the cells from the reference batch to use on the correction vector estimation.

pcaDim — Number of PCA dimensions to use.

perCellMNN — Threshold value to decide if a membership's correction value is calculated. As a rough interpretation, this values can be thought as the proportion of cells from a membership with an associated MNN pair. If the proportion is low, an specific correction vectors is not calculated for this membership.

fuzzy — Use fuzzy logic to join the local correction vectors.

fuzzyPCA — Number of PCs to use in the fuzzy process.

estMethod — Method to use when estimating the correction vectors:

- Median. Use the cells median distance.
- EKF. Use an extended Kalman filter.

clusterMethod — Method used to identify memberships.

pairsFilter — Filter MNNs pairs before estimating the correction vectors. If TRUE, the pairs are filtered from outliers using an interquartile range method.

doCosNorm — Whether to do cosine normalization.

correctEmbeddings — Whether to perform the correction on PCA embeddings instead of gene expression.

maxLoop — Number of times to repeat the correction, using each pass's corrected query batch as the input to the next. Only used when correctEmbeddings = TRUE; ignored (with a warning) otherwise, since each pass would need to recompute the PCA over gene expression.

loopTol — Used to stop repeating early. Once a new pass changes the correction, on average per cell, by less than this fraction of the previous pass, looping stops.

verbose — Print output.

Details

CorrectBatch is a method to correct batch-effect from two single-cell batches. Batch-effects observations are defined using mutual nearest neighbors (MNNs) pairs and cell groups from the query batch are distinguished using clustering. We estimate a correction vector for each cluster using its MNNs pairs and use these vectors to remove the batch effect from the query batch in two ways:

- A linear correction is performed by equally correcting the cells from the same cluster.
- A non-linear correction is performed by differently correcting each cell using fuzzy logic.

Value

A list containing the input batches, the corrected query batch, and the correction data

Examples

```
x <- SimBatches$batches[[1]]
y <- SimBatches$batches[[2]]
z <- CorrectBatch(x, y)
Corrected <- z$`Corrected Query Batch`

Uncorrected_PCA <- prcomp(t(cbind(x,y)))
plot(Uncorrected_PCA$x[,1:2])
Corrected_PCA <- prcomp(t(cbind(x,z$`Corrected Query Batch`)))
plot(Corrected_PCA$x[,1:2])
```

`CorrectBatches` *CorrectBatches*

Description

Batch-effect correction over a list of single cell batches

Usage

```
CorrectBatches(
  lsBatches,
  hierarchical = TRUE,
  queNumCelltypes = NULL,
  maxMem = 5,
  sampling = FALSE,
  numSamples = NULL,
  kNN = 30,
  ncores = 1,
  pcaDim = 50,
  pairsFilter = FALSE,
  perCellMNN = 0.08,
```

```
  fuzzy = TRUE,
  fuzzyPCA = 10,
  estMethod = "Median",
  clusterMethod = "louvain",
  doCosNorm = FALSE,
  fracSampling = NULL,
  debug = FALSE,
  correctEmbeddings = FALSE,
  precomputedEmbeddings = FALSE,
  maxLoop = 1,
  loopTol = 0.001,
  verbose = FALSE,
  ...
)
```

Arguments

lsBatches	List of batches to integrate. Batches should contain the same number of genes as rows.
hierarchical	Use hierarchical integration scheme when correcting more than two batches: at each step, pick the remaining batch most similar to the current reference (by MNN pair count). If set to FALSE, the input batches are sorted by number of cells and integrated on descending order. When correctEmbeddings is TRUE, scoring candidates reuses the already-computed embedding coordinates instead of recomputing a PCA. For gene-expression-space correction it recomputes a PCA per candidate per step.
queNumCelltypes	Number of cell types in the query batch. By default Canek searches the number of cell types using an heuristic algorithm. Change this parameter if you know the number of cell types in advanced.
maxMem	Maximum number of memberships from the query batch. This parameter is used on the heuristic algorithm to find the number of cell types.
sampling	Use MNNs pairs sampling when using a Kalman filter to estimate the correction vector.
numSamples	If sampling. Number of MNNs pairs samples to use on the estimation process.
kNN	Number of k-nearest-neighbors used to define the MNNs pairs.
ncores	Number of cores to use for finding MNN pairs. Defaults to 1 (sequential). Requesting more than 1 errors on Windows (no fork() support there) or if it exceeds the number of cores parallel::detectCores() reports, rather than silently falling back or reducing.
pcaDim	Number of PCA dimensions to use.
pairsFilter	Filter MNNs pairs before estimating the correction vectors. If TRUE, the pairs are filtered from outliers using an interquartile range method.
perCellMNN	Threshold value to decide if a membership's correction value is calculated. As a rough interpretation, this values can be thought as the proportion of cells from a membership with an associated MNN pair. If the proportion is low, an specific correction vectors is not calculated for this membership.

`fuzzy` Use fuzzy logic to join the local correction vectors.

`fuzzyPCA` Number of PCs to use in the fuzzy process.

`estMethod` Method to use when estimating the correction vectors:

- Median. Use the cells median distance
- EKF. Use an extended Kalman filter

`clusterMethod` Method used to identify memberships.

`doCosNorm` Whether to do cosine normalization.

`fracSampling` Fraction of cells to sample in the hierarchical selection (default is NULL, no sampling).

`debug` Return correction's information

`correctEmbeddings` Whether to perform the correction on PCA embeddings instead of gene expression. When TRUE, batches are transformed to embedding space before correcting (see precomputedEmbeddings), and hierarchical batch selection reuses those coordinates instead of recomputing a PCA per candidate.

`precomputedEmbeddings` Whether to use precomputed PCA embeddings. Set this to TRUE if lsBatches are already embeddings (e.g. PCA coordinates) rather than gene expression, so Canek uses them instead of computing its own PCA. Only used when correctEmbeddings = TRUE.

`maxLoop` Number of times to repeat the correction of a batch pair, using each pass's corrected query batch as the input to the next. Only used when correctEmbeddings = TRUE; ignored (with a warning) otherwise, since each pass would need to recompute the PCA over gene expression.

`loopTol` Used to stop repeating early. Once a new pass changes the correction, on average per cell, by less than this fraction of the previous pass, looping stops.

`verbose` Print output.

`...` Pass down methods from RunCanek().

Details

CorrectBatches is a method to correct batch-effect from two or more single-cell batches. Batch-effects observations are defined using mutual nearest neighbors (MNNs) pairs and cell groups from the query batch are distinguished using clustering. We estimate a correction vector for each cluster using its MNNs pairs and use these vectors to remove the batch effect from the query batch in two ways:

- A linear correction is performed by equally correcting the cells from the same cluster.
- A non-linear correction is performed by differently correcting each cell using fuzzy logic.

Value

A list containing the integrated datasets as matrix and the correction data .

Examples

```
Batches <- SimBatches$batches
z <- CorrectBatches(Batches)

Uncorrected_PCA <- prcomp(t(cbind(Batches[[1]], Batches[[2]])))
plot(Uncorrected_PCA$x[,1:2])
Corrected_PCA <- prcomp(t(z))
plot(Corrected_PCA$x[,1:2])
```

EkfBE *Correction vector estimation*

Description

Batch effect estimation using an extended Kalman filter

Usage

```
EkfBE(
  refBatch,
  queBatch,
  pairs,
  sampling = FALSE,
  numSamples = NULL,
  verbose = FALSE
)
```

Arguments

refBatch	Reference batch.
queBatch	Query batch.
pairs	A numerical matrix containing MNNs pairs cell indexes. First column corresponds to query batch cells.
sampling	Sample MNNs pairs.
numSamples	If sampling, number of MNNs pairs samples to use on the estimation process.
verbose	Print output.

Details

The input batches must have the same number of genes. The model used on the estimation has the form of g_ref = g_que + be, where the batch effect is represented as a value added to the reference gene expression, causing a linear deviation between the reference and the query batches.

Value

A list containing the estimated correction vector and the estimation data. The length of the correction vector is equal to the number of genes.

`Fuzzy` *Title Fuzzy*

Description

Function to score cell's memberships by fuzzy logic

Usage

```
Fuzzy(
  cluMem = NULL,
  pcaQue = NULL,
  corCell = NULL,
  fuzzyPCA = 10,
  MST = NULL,
  verbose = FALSE
)
```

Arguments

`cluMem`	Memberships' clustering data.
`pcaQue`	PCA representation of the cells.
`corCell`	Matrix containing the initial membership assignment. Matrix dimensions are expected as #Cell x #Memberships, with each row sum equal to 1.
`fuzzyPCA`	Number of PCs to use in the fuzzy process.
`MST`	Minimum spanning tree
`verbose`	Print output.

Details

This function perform the fuzzification for the cells' membership. A minimum spanning tree (MST) is created among memberships, and the fuzzification is performed for each of the edges of the MST.#'

MeanBE *MeanBE*

Description

Batch effect estimation using the MNNs pairs.

Usage

```
MeanBE(refBatch, queBatch, pairs)
```

Arguments

refBatch	Reference batch.
queBatch	Query batch.
pairs	A numerical matrix containing MNNs pairs cell indexes. First column corresponds to query batch cells.

Details

The input batches must have the same number of genes. The model used on the estimation has the form of g_ref = g_que + be, where the batch effect is represented as a value added to the reference gene expression. The batch effect is estimated as the median of the gene expression difference among the reference and the query batch, e.g. Median(g_ref - g_que).

Value

A list containing the estimated correction vector and the estimation data. The length of the correction vector is equal to the number of genes.

MedianBE *Correction vector estimation*

Description

Batch effect estimation using the MNNs pairs.

Usage

```
MedianBE(refBatch, queBatch, pairs)
```

Arguments

refBatch	Reference batch.
queBatch	Query batch.
pairs	A numerical matrix containing MNNs pairs cell indexes. First column corresponds to query batch cells.

Details

The input batches must have the same number of genes. The model used on the estimation has the form of g_ref = g_que + be, where the batch effect is represented as a value added to the reference gene expression. The batch effect is estimated as the median of the gene expression difference among the reference and the query batch, e.g. Median(g_ref - g_que).

Value

A list containing the estimated correction vector and the estimation data. The length of the correction vector is equal to the number of genes.

`PairsFiltering` *Title PairsFiltering*

Description

Function to filter MNNs pairs

Usage

```
PairsFiltering(refBatch, queBatch, pairs, verbose = FALSE)
```

Arguments

`refBatch`	Reference batch single-cell data.
`queBatch`	Query's batch single-cell data.
`pairs`	A matrix containing MNNs pairs. First column corresponds to query-batch cell indexes.
`verbose`	Print output.

Details

Filter MNN pairs by quantiles.

Value

A matrix containing the filtered pairs. First column corresponds to query-batch cell indexes.

RunCanek *RunCanek*

Description

Runs Canek integration.

Usage

```
RunCanek(x, ...)

## S3 method for class 'Seurat'
RunCanek(
  x,
  batches = NULL,
  slot = "data",
  assay = NULL,
  features = NULL,
  selection.method = "vst",
  nfeatures = 2000,
  fvf.nfeatures = 2000,
  integration.name = "Canek",
  debug = FALSE,
  correctEmbeddings = TRUE,
  pcaDim = NULL,
  maxLoop = 5,
  loopTol = 0.001,
  ...
)

## S3 method for class 'SingleCellExperiment'
RunCanek(
  x,
  batches = NULL,
  assay = "logcounts",
  integration.name = "Canek",
  debug = FALSE,
  ...
)

## S3 method for class 'list'
RunCanek(x, ...)
```

Arguments

x	object with expression counts or list of matrices.

...	additional arguments passed down to methods, e.g. ncores to parallelize MNN pair finding (see CorrectBatches; defaults to 1, sequential).
batches	for S4 objects the column containing batch information.
slot	slot used for Seurat objects (default: data).
assay	assay used for Seurat objects. If NULL (default), an existing "SCT" assay is preferred over the object's default assay.
features	optional vector of features to use for correction.
selection.method	method used for FindVariableFeatures on Seurat objects when features is NULL.
nfeatures	number of features returned by SelectIntegrationFeatures.
fvf.nfeatures	number of features returned by FindVariableFeatures.
integration.name	name for the integrated assay.
debug	whether to store information about correction vector.
correctEmbeddings	whether to perform the correction on PCA embeddings instead of gene expression (Seurat objects only).
pcaDim	number of PCA dimensions to use when correctEmbeddings is TRUE. If NULL (default), it is inferred from the object's existing "pca" reduction; if none is found, used 30 as default with a warning. When correctEmbeddings is TRUE and a "pca" reduction already exists on the object, it is reused directly instead of being recomputed. If the assay in use is SCTransform-normalized and no "pca" reduction exists, an error is raised instead of computing one internally, since SCTransform fit separately per batch produces residuals that are not on a comparable scale across batches.
maxLoop	number of times to iterate the correction (correctEmbeddings = TRUE only), using each iteration's corrected result as the input to the next. Defaults to 5.
loopTol	average change in median correction magnitude used to stop iterating early. Defaults to 1e-3. Ignored if maxLoop = 1.

Value

An object of the appropriate type.

SimBatches *Dataset with simulated single cell RNA-seq from 2 batches.*

Description

Dataset with simulated single cell RNA-seq from 2 batches.

Usage

```
SimBatches
```

Format

A list with the following elements:

batches a list with two matrices representing the two batches

pairs matrix of pairs between the two batches.

cell_types a factor with the cell clusters. ...

Index

∗ **datasets**
SimBatches, 13

CheckZeroCV, 2
CorrectBatch, 3
CorrectBatches, 5, *13*

EkfBE, 8

Fuzzy, 9

MeanBE, 10
MedianBE, 10

PairsFiltering, 11

RunCanek, 12

SimBatches, 13