Package ‘scPOP’

August 4, 2021

Type Package

Title Metrics for Benchmarking scRNA-Seq Batch Correction

Version 0.1.0

Description Evaluate batch effect correction algorithms for scRNA-seq using multiple established methods, including the Adjusted Rand Index, Normalized Mutual Information, Local Inverse Simpson Index, and Silhouette Width. Methods for aggregating and weighing multiple metrics together are also included. For further explanation of methods, see Swamy et al. (2021) <doi:10.1101/2021.03.26.437190>.

Maintainer Vinay Swamy <swamyvinny@gmail.com>

License GPL-3

Encoding UTF-8

Imports Rcpp (>= 1.0.5), Matrix, RANN, cluster

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.1

Depends R (>= 2.10)

NeedsCompilation yes

Author Vinay Swamy [aut, cre],
    David McGaughey [aut]

Repository CRAN

Date/Publication 2021-08-04 15:10:02 UTC

R topics documented:

ari ................................................................. 2
calc_sumZscore .................................................. 2
compute_simpson_index .................................... 4
lisi ................................................................. 4
nmi ................................................................. 5
run_all_metrics .............................................. 6
sceiad_subset_data ......................................... 7
### ari  
*Adjusted Rand Index*

**Description**

A function to compute the adjusted rand index between two classifications

**Usage**

ari(c1, c2)

**Arguments**

- **c1**  
  a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.

- **c2**  
  a vector containing the labels of the second classification.

**Value**

a scalar with the adjusted rand index.

**Examples**

```r
## calculate Adjusted Rand Index on two sets of labels
data(sceiad_subset_data)
ari(sceiad_subset_data$CellType_predict, sceiad_subset_data$cluster)
```

---

### calc_sumZscore  
*Calc_sumZscore*

**Description**

Aggregate multiple integration metrics across multiple integration runs, i.e., from different batch correction algorithms, or different parameters for the same algorithms.

**Usage**

calc_sumZscore(metric_df_list, batch_key)
calc_sumZscore

Arguments

metric_df_list a list of data.frames generated by applying run_all_metrics to multiple sets of integrations

batch_key name of batch column in metadata used when generating run_all_metrics

Value

a vector of aggregated, z-scored metrics

Examples

library(scPOP)
data(sceiad_subset_data)

features <- sceiad_subset_data[, paste0('scviDim_', 1:8)]
metadata_1 <- sceiad_subset_data[,c('Barcode', 'cluster', 'subcluster', 'batch', 'CellType', 'CellType_predict')]

## scramble example dataset to generate multiple integration runs
metadata_2 <- metadata_1
metadata_2$batch <- sample(metadata_2$batch, length(metadata_2$batch))
metadata_2$CellType_predict <- sample(metadata_2$CellType_predict, length(metadata_2$CellType_predict))
metadata_2$cluster <- sample(metadata_2$cluster, length(metadata_2$cluster))

metadata_3 <- metadata_1
metadata_3$batch <- sample(metadata_3$batch, length(metadata_3$batch))
metadata_3$CellType_predict <- sample(metadata_3$CellType_predict, length(metadata_3$CellType_predict))
metadata_3$cluster <- sample(metadata_3$cluster, length(metadata_3$cluster))

integration_data_list <- list(metadata_1, metadata_2, metadata_3)
metric_df_list <- lapply(integration_data_list, function(x)
  run_all_metrics(reduction = features,
  metadata = x,
  batch_key = 'batch',
  label1_key = 'CellType_predict',
  label2_key = 'cluster',
  run_name = 'example',
  quietly =TRUE
  )
)

calc_sumZscore(metric_df_list,'batch')
**compute_simpson_index**  
*Compute the Local Inverse Simpson Index (LISI)*

**Description**
Compute the Local Inverse Simpson Index (LISI)

**Usage**
```r
compute_simpson_index(
  D,
  knn_idx,
  batch_labels,
  n_batches,
  perplexity = 15,
  tol = 1e-05
)
```

**Arguments**
- **D**: Distance matrix of K nearest neighbors.
- **knn_idx**: Adjacency matrix of K nearest neighbors.
- **batch_labels**: A categorical variable.
- **n_batches**: The number of categories in the categorical variable.
- **perplexity**: The effective number of neighbors around each cell.
- **tol**: Stop when the score converges to this tolerance.

---

**lisi**  
*Compute Local Inverse Simpson’s Index (LISI)*

**Description**
Use this function to compute LISI scores of one or more labels.

**Usage**
```r
lisi(X, meta_data, label_colnames, perplexity = 30, nn_eps = 0)
```

**Arguments**
- **X**: A matrix with cells (rows) and features (columns).
- **meta_data**: A data frame with one row per cell.
- **label_colnames**: Which variables to compute LISI for.
- **perplexity**: The effective number of each cell’s neighbors.
- **nn_eps**: Error bound for nearest neighbor search with RANN::nn2(). Default of 0.0 implies exact nearest neighbor search.
**Value**

A data frame of LISI values. Each row is a cell and each column is a different label variable.

**Examples**

```r
data(sceiad_subset_data)
features <- sceiad_subset_data[, paste0('scviDim_`, 1:8)]
metadata <- sceiad_subset_data[,c('Barcode', 'cluster', 'subcluster', 'CellType', 'CellType_predict')]
lisi_scores <- lisi(features, metadata, c('CellType_predict'))
head(lisi_scores)
```

---

**nmi**

**Normalized mutual information (NMI)**

**Description**

A function to compute the NMI between two classifications

**Usage**

```r
nmi(c1, c2, variant = c("max", "min", "sqrt", "sum", "joint"))
```

**Arguments**

- `c1`: a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
- `c2`: a vector containing the labels of the second classification.
- `variant`: a string in ("max", "min", "sqrt", "sum", "joint"): different variants of NMI. Default use "max".

**Value**

A scalar with the normalized mutual information.

**Examples**

```r
## calculate Normalized Mutal Information score for two sets of labels
data(sceiad_subset_data)
nmi(sceiad_subset_data$CellType_predict, sceiad_subset_data$cluster)
```
run_all_metrics  

Running All Metrics

Description

Running All Metrics

Usage

run_all_metrics(
  reduction,
  metadata,
  batch_key,
  label1_key,
  label2_key,
  run_name = NULL,
  sil_width_prop = 1,
  sil_width_group_key = NULL,
  quietly = F
)

Arguments

reduction  A matrix of reduced dimensions
metadata    A data.frame containing information like batch, cell type, etc
batch_key   Name of column in metadata corresponding to batch
label1_key  Name of column in metadata corresponding to primary cell label, eg Cell type
label2_key  Name of column in metadata corresponding to secondary cell label, eg cluster identity
run_name    (optional) name to refer to dataset
sil_width_prop  (optional) proportion of data to use for silhouette_width
sil_width_group_key  (optional) which column in metadata to use for stratified sampling of data
quietly     (optional) if TRUE dont print anything

Value

A one row data.frame of calculated metrics
**sceiad_subset_data**

Example scRNA-seq data from the single cell eye in a disk(sceiad) the original data set this was pulled from can be found at this link 'https://hpc.nih.gov/~mcgaugheyd/scEiaD/colab/scEiaD_all_anndata_mini_ref.h5ad'

**Description**

Example scRNA-seq data from the single cell eye in a disk(sceiad) the original data set this was pulled from can be found at this link 'https://hpc.nih.gov/~mcgaugheyd/scEiaD/colab/scEiaD_all_anndata_mini_ref.h5ad'

**Usage**

```r
data(sceiad_subset_data)
```

**Format**

An object of class "data.frame"

**Source**

`<"https://hpc.nih.gov/~mcgaugheyd/scEiaD/colab/scEiaD_all_anndata_mini_ref.h5ad">`

**Examples**

```r
data(sceiad_subset_data)
head(sceiad_subset_data)
```

---

**scPOP**

*scPOP: Metrics for Benchmarking scRNA-Seq Batch Correction*

**Description**

Evaluate using batch effect correction for scRNA-seq using multiple established methods, including the Adjusted Rand Index, Normalized Mutual Information, Local Inverse Simpson Index, and Silhouette Width. We also included metrics for aggregating and weighing multiple metrics together.
silhouette_width  

**Description**

Determine batch/bio effect using the silhouette coefficient (adopted from scone):

**Usage**

```
silhouette_width(reduction, meta.data, keys)
```

**Arguments**

- `reduction`: a matrix of reduced dimensions
- `meta.data`: dataframe with meta.data associated with reduction
- `keys`: columns in meta.data to calculate silhouette for to use (default: all)

**Value**

The average silhouette width for all clusters. For batch effect, the smaller the better. For biological effect, the larger the better.

**Examples**

```r
## calculate the silhouette width score on two sets of labels
## NOTE: this requires computation of a distance matrix, so does not
## scale well to large datasets
data(sceiad_subset_data)
features <- sceiad_subset_data[, paste0('scviDim_', 1:8)]
metadata <- sceiad_subset_data[,c('Barcode', 'cluster',
    'subcluster', 'CellType', 'CellType_predict')]
silhouette_width(features, metadata, 'CellType_predict')
```

---

stratified_sample  

**Generate a stratified subsample for a vector given a grouping**

**Description**

Use this function to compute LISI scores of one or more labels.
stratified_sample

Usage

stratified_sample(
    indexer,
    grouping,
    sample_proportion = 0.1,
    min_count = 0,
    seed = 424242
)

Arguments

indexer A vector containing cell barcodes/labels to subsample

grouping A vector containg a groups to stratify by (same size as indexer)

sample_proportion proportion to sample data (default: .1)

min_count Minimum number of samples in a group to keep

seed seed value for set.seed

Value

A subsampled vector generated from indexer

Examples

data(sceiad_subset_data)
rownames(sceiad_subset_data) <- sceiad_subset_data$Barcode
res = stratified_sample(sceiad_subset_data$Barcode, sceiad_subset_data$cluster)
dim(sceiad_subset_data[res,])
Index

* datasets
  - sceiad_subset_data, 7

ari, 2

calc_sumZscore, 2
compute_simpson_index, 4

lisi, 4

nmi, 5

run_all_metrics, 3, 6
sceiad_subset_data, 7
scPOP, 7
silhouette_width, 6, 8
stratified_sample, 8