

# Package ‘symphony’

August 5, 2021

**Title** Efficient and Precise Single-Cell Reference Atlas Mapping

**Version** 0.1.0

**Description** Implements the Symphony single-cell reference building and query mapping algorithms and additional functions described in Kang et al <<https://www.biorxiv.org/content/10.1101/2020.11.18.389189v2>>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown, testthat, ggthemes

**LinkingTo** Rcpp, RcppArmadillo

**Imports** methods, Rcpp, harmony, uwot, irlba, class, purrr, dplyr, ggplot2, stats, utils, magrittr, data.table, tibble, Matrix, tidy, rlang, RColorBrewer, ggrastr, ggrepel, RANN

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

**NeedsCompilation** yes

**Author** Joyce Kang [aut, cre] (<<https://orcid.org/0000-0002-1962-1291>>), Ilya Korsunsky [aut] (<<https://orcid.org/0000-0003-4848-3948>>), Soumya Raychaudhuri [aut] (<<https://orcid.org/0000-0002-1901-8265>>)

**Maintainer** Joyce Kang <[joyce\\_kang@hms.harvard.edu](mailto:joyce_kang@hms.harvard.edu)>

**Repository** CRAN

**Date/Publication** 2021-08-05 07:50:05 UTC

## R topics documented:

buildReference	2
buildReferenceFromHarmonyObj	3
calcknncorr	4
calcknncorrWithinQuery	5

calcPerCellMappingMetric . . . . .	6
calcPerClusterMappingMetric . . . . .	7
evaluate . . . . .	8
findVariableGenes . . . . .	8
knnPredict . . . . .	9
mapQuery . . . . .	10
pbmcs_exprs_small . . . . .	11
pbmcs_meta_small . . . . .	11
plotReference . . . . .	12
rowSDs . . . . .	13
runPCAQueryAlone . . . . .	13
scaleDataWithStats . . . . .	14
symphony . . . . .	14
vargenes_vst . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

buildReference	<i>Function for building a Symphony reference starting from expression matrix</i>
----------------	-----------------------------------------------------------------------------------

---

## Description

Function for building a Symphony reference starting from expression matrix

## Usage

```
buildReference(
  exp_ref,
  metadata_ref,
  vars = NULL,
  K = 100,
  verbose = FALSE,
  do_umap = TRUE,
  do_normalize = TRUE,
  vargenes_method = "vst",
  vargenes_groups = NULL,
  topn = 2000,
  tau = 0,
  theta = 2,
  save_uwot_path = NULL,
  d = 20,
  additional_genes = NULL,
  umap_min_dist = 0.1,
  seed = 111
)
```

**Arguments**

exp_ref	Reference gene expression (genes by cells)
metadata_ref	Reference cell metadata (cells by attributes)
vars	Reference variables to Harmonize over e.g. c('donor', 'technology')
K	Number of soft cluster centroids in model
verbose	Verbose output
do_umap	Perform UMAP visualization on harmonized reference embedding
do_normalize	Perform log(CP10K+1) normalization
vargenes_method	Variable gene selection method (either 'vst' or 'mvp')
vargenes_groups	Name of metadata column specifying groups for variable gene selection. If not NULL, calculate topn variable genes in each group separately, then pool
topn	Number of variable genes to subset by
tau	Tau parameter for Harmony step
theta	Theta parameter(s) for Harmony step
save_uwot_path	Absolute path to save the uwot model (used if do_umap is TRUE)
d	Number of PC dimensions
additional_genes	Any custom genes (e.g. marker genes) to include in addition to variable genes
umap_min_dist	umap parameter (see uwot documentation for details)
seed	Random seed

**Value**

Symphony reference object. Integrated embedding is stored in the `$Z_corr` slot. Other slots include cell-level metadata (`$meta_data`), variable genes means and standard deviations (`$vargenes`), loadings from PCA (`$loadings`), original PCA embedding (`$Z_orig`), reference compression terms (`$cache`), betas from Harmony integration (`$betas`), cosine normalized soft cluster centroids (`$centroids`), centroids in PC space (`$centroids_pc`), and optional umap coordinates (`$umap$embedding`).

---

buildReferenceFromHarmonyObj

*Function for building a Symphony reference from a Harmony object. Useful if you would like your code to be more modular. Note that you must have saved vargenes\_means\_sds and PCA loadings.*

---

**Description**

Function for building a Symphony reference from a Harmony object. Useful if you would like your code to be more modular. Note that you must have saved vargenes\_means\_sds and PCA loadings.

**Usage**

```
buildReferenceFromHarmonyObj(
  harmony_obj,
  metadata,
  vargenes_means_sds,
  pca_loadings,
  verbose = TRUE,
  do_umap = TRUE,
  save_uwot_path = NULL,
  umap_min_dist = 0.1,
  seed = 111
)
```

**Arguments**

harmony_obj	Harmony object (output from HarmonyMatrix())
metadata	Reference cell metadata (cells by attributes)
vargenes_means_sds	Variable genes in dataframe with columns named ('symbol', 'mean', 'stddev')
pca_loadings	Gene loadings from PCA (e.g. irlba(ref_exp_scaled, nv = 20)\$u)
verbose	Verbose output
do_umap	Perform UMAP visualization on harmonized reference embedding
save_uwot_path	Absolute path to save the uwot model (if do_umap is TRUE)
umap_min_dist	UMAP parameter (see uwot documentation for details)
seed	Random seed

**Value**

Symphony reference object. Integrated embedding is stored in the \$Z\_corr slot. Other slots include cell-level metadata (\$meta\_data), variable genes means and standard deviations (\$vargenes), loadings from PCA or other dimensional reduction such as CCA (\$loadings), original PCA embedding (\$Z\_orig), reference compression terms (\$cache), betas from Harmony integration (\$betas), cosine-normalized soft cluster centroids (\$centroids), centroids in PC space (\$centroids\_pc), and optional umap coordinates (\$umap\$embedding).

---

calcknncorr

*Calculates the k-NN correlation, which measures how well the sorted ordering of k nearest reference neighbors in a gold standard embedding correlate with the ordering for the same reference cells in an alternative embedding (i.e. from reference mapping). NOTE: it is very important for the order of reference cells (cols) in gold\_ref matches that of alt\_ref (same for matching columns of gold\_query and alt\_query).*

---

**Description**

Calculates the k-NN correlation, which measures how well the sorted ordering of k nearest reference neighbors in a gold standard embedding correlate with the ordering for the same reference cells in an alternative embedding (i.e. from reference mapping). NOTE: it is very important for the order of reference cells (cols) in gold\_ref matches that of alt\_ref (same for matching columns of gold\_query and alt\_query).

**Usage**

```
calcknncorr(gold_ref, alt_ref, gold_query, alt_query, k = 500)
```

**Arguments**

gold_ref	Reference cells in gold standard embedding (PCs by cells)
alt_ref	Reference cells in alternative embedding (PCs by cells)
gold_query	Query cells in gold standard embedding (PCs by cells)
alt_query	Query cells in alternative embedding (PCs by cells)
k	Number of reference neighbors to use for kNN-correlation calculation

**Value**

Vector of k-NN correlations for query cells

---

calcknncorrWithinQuery

*Calculates the k-NN correlation within the query cells only, which measures how well the sorted ordering of k nearest query neighbors in a query de novo PCA embedding correlate with the ordering for the cells in the reference mapping embedding.*

---

**Description**

Calculates the k-NN correlation within the query cells only, which measures how well the sorted ordering of k nearest query neighbors in a query de novo PCA embedding correlate with the ordering for the cells in the reference mapping embedding.

**Usage**

```
calcknncorrWithinQuery(
  query,
  var = NULL,
  k = 100,
  topn = 2000,
  d = 20,
  distance = "euclidean"
)
```

**Arguments**

query	Query object (returned from mapQuery)
var	Query metadata batch variable (PCA is calculated within each batch separately); if NULL, do not split by batch
k	Number of neighbors to use for kNN-correlation calculation
topn	number of variable genes to calculate within each query batch for query PCA
d	number of dimensions for query PCA within each query batch
distance	either 'euclidean' or 'cosine'

**Value**

Vector of within-query k-NN correlations for query cells

---

**calcPerCellMappingMetric**

*Per-cell Confidence Score: Calculates the weighted Mahalanobis distance for the query cells to reference clusters. Returns a vector of distance scores, one per query cell. Higher distance metric indicates less confidence.*

---

**Description**

Per-cell Confidence Score: Calculates the weighted Mahalanobis distance for the query cells to reference clusters. Returns a vector of distance scores, one per query cell. Higher distance metric indicates less confidence.

**Usage**

```
calcPerCellMappingMetric(
  reference,
  query,
  Z_orig = TRUE,
  metric = "mahalanobis"
)
```

**Arguments**

reference	Reference object as returned by Symphony buildReference()
query	Query object as returned by Symphony mapQuery()
Z_orig	Define reference distribution using original PCA embedding or harmonized PC embedding
metric	Uses Mahalanobis by default, but added as a parameter for potential future use

**Value**

A vector of per-cell mapping metric scores for each cell.

---

**calcPerClusterMappingMetric**

*Per-cluster Confidence Score: Calculates the Mahalanobis distance from user-defined query clusters to their nearest reference centroid after initial projection into reference PCA space. All query cells in a cluster get the same score. Higher distance indicates less confidence. Due to the instability of estimating covariance with small numbers of cells, we do not assign a score to clusters smaller than  $u * d$ , where  $d$  is the dimensionality of the embedding and  $u$  is specified.*

---

**Description**

Per-cluster Confidence Score: Calculates the Mahalanobis distance from user-defined query clusters to their nearest reference centroid after initial projection into reference PCA space. All query cells in a cluster get the same score. Higher distance indicates less confidence. Due to the instability of estimating covariance with small numbers of cells, we do not assign a score to clusters smaller than  $u * d$ , where  $d$  is the dimensionality of the embedding and  $u$  is specified.

**Usage**

```
calcPerClusterMappingMetric(
  reference,
  query,
  query_cluster_labels,
  metric = "mahalanobis",
  u = 2,
  lambda = 0
)
```

**Arguments**

reference	Reference object as returned by Symphony buildReference()
query	Query object as returned by Symphony mapQuery()
query_cluster_labels	Vector of user-defined labels denoting clusters / putative novel cell type to calculate the score for
metric	Uses Mahalanobis by default, but added as a parameter for potential future use
u	Do not assign scores to clusters smaller than $u * d$ (see above description)
lambda	Optional ridge parameter added to covariance diagonal to help stabilize numeric estimates

**Value**

A data.frame of per-cluster mapping metric scores for each user-specified query cluster.

---

evaluate	<i>Function for evaluating F1 by cell type, adapted from automated cell type identification benchmarking paper (Abdelaal et al. Genome Biology, 2019)</i>
----------	-----------------------------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Function for evaluating F1 by cell type, adapted from automated cell type identification benchmarking paper (Abdelaal et al. Genome Biology, 2019)

### Usage

```
evaluate(true, predicted)
```

### Arguments

true	vector of true labels
predicted	vector of predicted labels

### Value

A list of results with confusion matrix (\$Conf), median F1-score (\$MedF1), F1 scores per class (\$F1), and accuracy (\$Acc).

---

findVariableGenes	<i>Function to find variable genes using mean variance relationship method</i>
-------------------	--------------------------------------------------------------------------------

---

### Description

Function to find variable genes using mean variance relationship method

### Usage

```
findVariableGenes(
  X,
  groups,
  min_expr = 0.1,
  max_expr = Inf,
  min_dispersion = 0,
  max_dispersion = Inf,
  num.bin = 20,
  binning.method = "equal_width",
  return_top_n = 0
)
```



**Arguments**

X	expression matrix
groups	vector of groups
min_expr	min expression cutoff
max_expr	max expression cutoff
min_dispersion	min dispersion cutoff
max_dispersion	max dispersion cutoff
num.bin	number of bins to use for scaled analysis
binning.method	how bins are computed
return_top_n	returns top n genes

**Value**

A data.frame of variable genes

---

knnPredict	<i>Predict annotations of query cells from the reference using k-NN method</i>
------------	--------------------------------------------------------------------------------

---

**Description**

Predict annotations of query cells from the reference using k-NN method

**Usage**

```
knnPredict(
  query_obj,
  ref_obj,
  train_labels,
  k = 5,
  save_as = "cell_type_pred_knn",
  confidence = TRUE,
  seed = 0
)
```

**Arguments**

query_obj	Symphony query object
ref_obj	Symphony reference object
train_labels	vector of labels to train
k	number of neighbors
save_as	string that result column will be named in query metadata
confidence	return k-NN confidence scores (proportion of neighbors voting for the predicted annotation)
seed	random seed (k-NN has some stochasticity in the case of ties)

**Value**

Symphony query object, with predicted reference labels stored in the 'save\_as' slot of the query\$meta\_data

---

 mapQuery

*Function for mapping query cells to a Symphony reference*


---

**Description**

Function for mapping query cells to a Symphony reference

**Usage**

```
mapQuery(
  exp_query,
  metadata_query,
  ref_obj,
  vars = NULL,
  verbose = TRUE,
  do_normalize = TRUE,
  do_umap = TRUE,
  sigma = 0.1
)
```

**Arguments**

exp_query	Query gene expression (genes by cells)
metadata_query	Query metadata (cells by attributes)
ref_obj	Reference object as returned by Symphony buildReference()
vars	Query batch variable(s) to integrate over (column names in metadata)
verbose	Verbose output
do_normalize	Perform log(CP10K+1) normalization on query expression
do_umap	Perform umap projection into reference UMAP (if reference includes a uwot model)
sigma	Fuzziness parameter for soft clustering (sigma = 1 is hard clustering)

**Value**

Symphony query object. Mapping embedding is in the \$Z slot. Other slots include query expression matrix (\$exp), query cell-level metadata (\$meta\_data), query cell embedding in pre-Harmonized reference PCs (\$Zq\_pca), query cell soft cluster assignments (\$R), and query cells in reference UMAP coordinates (\$umap).

---

pbmcs_exprs_small	<i>Log(CP10k+1) normalized counts matrix (genes by cells) for 10x PBMCs dataset for vignette.</i>
-------------------	---------------------------------------------------------------------------------------------------

---

**Description**

Log(CP10k+1) normalized counts matrix (genes by cells) for 10x PBMCs dataset for vignette.

**Usage**

pbmcs\_exprs\_small

**Format**

: Sparse matrix (dgCMatrix): dimensions 1,764 genes by 1,200 cells

**Source**

<https://support.10xgenomics.com/single-cell-gene-expression/datasets>

---

pbmcs_meta_small	<i>Metadata for 10x PBMCs dataset for vignette.</i>
------------------	-----------------------------------------------------

---

**Description**

Metadata for 10x PBMCs dataset for vignette.

**Usage**

pbmcs\_meta\_small

**Format**

: A data frame with 1,200 cells and 7 metadata fields.

**cell\_id** unique cell ID

**donor** dataset (3pv1, 3pv2, or 5p)

**nUMI** number of UMIs

**nGene** number of genes

**percent\_mito** percent mito genes

**cell\_type** cell type assigned in Symphony publication

**cell\_type\_broad** cell subtype assigned in Symphony publication

**Source**

<https://support.10xgenomics.com/single-cell-gene-expression/datasets>

---

plotReference                      *Function to plot reference, colored by cell type*

---

### Description

Function to plot reference, colored by cell type

### Usage

```
plotReference(  
  reference,  
  as.density = TRUE,  
  bins = 10,  
  bandwidth = 1.5,  
  title = "Reference",  
  color.by = "cell_type",  
  celltype.colors = NULL,  
  show.legend = TRUE,  
  show.labels = TRUE,  
  show.centroids = FALSE  
)
```

### Arguments

reference	Symphony reference object (must have UMAP stored)
as.density	if TRUE, plot as density; if FALSE, plot as individual cells
bins	for density, nbins parameter for stat_density_2d
bandwidth	for density, bandwidth parameter for stat_density_2d
title	Plot title
color.by	metadata column name for phenotype labels
celltype.colors	custom color mapping
show.legend	Show cell type legend
show.labels	Show cell type labels
show.centroids	Plot soft cluster centroid locations

### Value

A ggplot object.

---

rowSDs	<i>Calculate standard deviations by row</i>
--------	---------------------------------------------

---

**Description**

Calculate standard deviations by row

**Usage**

```
rowSDs(A, row_means = NULL, weights = NULL)
```

**Arguments**

A	expression matrix (genes by cells)
row_means	row means
weights	weights for weighted standard dev calculation

**Value**

A vector of row standard deviations

---

runPCAQueryAlone	<i>Runs a standard PCA pipeline on query (1 batch). Assumes query_exp is already normalized.</i>
------------------	--------------------------------------------------------------------------------------------------

---

**Description**

Runs a standard PCA pipeline on query (1 batch). Assumes query\_exp is already normalized.

**Usage**

```
runPCAQueryAlone(query_exp, topn = 2000, d = 20, seed = 1)
```

**Arguments**

query_exp	Query expression matrix (genes x cells)
topn	Number of variable genes to use
d	Number of dimensions
seed	random seed

**Value**

A matrix of PCs by cells

---

scaleDataWithStats	<i>Scale data with given mean and standard deviations</i>
--------------------	-----------------------------------------------------------

---

**Description**

Scale data with given mean and standard deviations

**Usage**

```
scaleDataWithStats(A, mean_vec, sd_vec, margin = 1, thresh = 10)
```

**Arguments**

A	expression matrix (genes by cells)
mean_vec	vector of mean values
sd_vec	vector of standard deviation values
margin	1 for row-wise calculation
thresh	threshold to clip max values

**Value**

A matrix of scaled expression values.

---

symphony	<i>symphony</i>
----------	-----------------

---

**Description**

Efficient single-cell reference atlas mapping (Kang et al.)

---

vargenes_vst	<i>Function to find variable genes using variance stabilizing transform (vst) method</i>
--------------	------------------------------------------------------------------------------------------

---

**Description**

Function to find variable genes using variance stabilizing transform (vst) method

**Usage**

```
vargenes_vst(object, groups, topn, loess.span = 0.3)
```

**Arguments**

object	expression matrix
groups	finds variable genes within each group then pools
topn	Return top n genes
loess.span	Loess span parameter used when fitting the variance-mean relationship

**Value**

A data.frame of variable genes, with means and standard deviations.

# Index

## \* datasets

pbmcs\_exprs\_small, 11

pbmcs\_meta\_small, 11

buildReference, 2

buildReferenceFromHarmonyObj, 3

calcknncorr, 4

calcknncorrWithinQuery, 5

calcPerCellMappingMetric, 6

calcPerClusterMappingMetric, 7

evaluate, 8

findVariableGenes, 8

knnPredict, 9

mapQuery, 10

pbmcs\_exprs\_small, 11

pbmcs\_meta\_small, 11

plotReference, 12

rowSDs, 13

runPCAQueryAlone, 13

scaleDataWithStats, 14

symphony, 14

vargenes\_vst, 14