Package 'harmony'

October 1, 2023

Title Fast, Sensitive, and Accurate Integration of Single Cell Data

Version 1.0.3

Description Implementation of the Harmony algorithm for single cell integration, described in Korsunsky et al <doi:10.1038/s41592-019-0619-0>. Package includes a standalone Harmony function and interfaces to external frameworks.

URL software.broadinstitute.org/harmony

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Depends R(>= 3.5.0), Rcpp

LazyData true

LazyDataCompression gzip

LinkingTo Rcpp, RcppArmadillo, RcppProgress

Imports dplyr, cowplot, ggplot2, Matrix, methods, tibble, rlang, RhpcBLASctl

Suggests SingleCellExperiment, Seurat (>= 4.1.1), testthat, knitr, rmarkdown, ggthemes, ggrepel, patchwork, tidyverse, tidyr, data.table

VignetteBuilder knitr

NeedsCompilation yes

Author Ilya Korsunsky [cre, aut] (<https://orcid.org/0000-0003-4848-3948>), Martin Hemberg [aut] (<https://orcid.org/0000-0001-8895-5239>), Nikolaos Patikas [aut, ctb] (<https://orcid.org/0000-0002-3978-0134>), Hongcheng Yao [aut, ctb] (<https://orcid.org/0000-0002-0743-4835>), Nghia Millard [aut] (<https://orcid.org/0000-0002-0518-7674>), Jean Fan [aut, ctb] (<https://orcid.org/0000-0002-0212-5451>), Kamil Slowikowski [aut, ctb] (<https://orcid.org/0000-0002-2843-6370>), Miles Smith [ctb], Soumya Raychaudhuri [aut] (<https://orcid.org/0000-0002-1901-8265>)

Maintainer Ilya Korsunsky <ilya.korsunsky@gmail.com>

Repository CRAN

Date/Publication 2023-09-30 23:00:02 UTC

R topics documented:

`cell_lines` *List of metadata table and scaled PCs matrix*

Description

List of metadata table and scaled PCs matrix

Usage

```
cell_lines
```

Format

: meta_data: data.table of 9478 rows with defining dataset and cell_type scaled_pcs: data.table of 9478 rows (cells) and 20 columns (PCs)

Source

https://www.10xgenomics.com

`cell_lines_small` *Same as cell_lines but smaller (300 cells).*

Description

Same as cell_lines but smaller (300 cells).

Usage

```
cell_lines_small
```

Format

An object of class `list` of length 2.

Source

https://www.10xgenomics.com

---

harmony *Harmony: fast, accurate, and robust single cell integration.*

Description

Algorithm for single cell integration.

Usage

?RunHarmony to run Harmony on cell embeddings matrix, Seurat or SingleCellExperiment objects.

Useful links

1. Report bugs at https://github.com/immunogenomics/harmony/issues
2. Read the manuscript doi:10.1038/s4159201906190

---

harmony_options *Set advanced options for RunHarmony*

Description

Set advanced options for RunHarmony

Usage

```
harmony_options(
  lambda_range = c(0.1, 10),
  tau = 0,
  block.size = 0.05,
  max.iter.cluster = 20,
  epsilon.cluster = 1e-05,
  epsilon.harmony = 1e-04
)
```

Arguments

lambda_range	Default lambda_range = c(0.1, 10). Lambda is ridge regression penalty parameter and smaller values result in more aggressive correction. During harmony iterations, the appropriate value of lambda is dynamically estimated. And parameter 'lambda_range' set the allowed range for lambda estimation. e.g. 'lambda_range' = c(0.1, 10) means that lambda can only vary between 0.1 and 10 when being dynamically estimated. Note that when setting the upper and lower bound of lambda_range to the same value would result in using a fixed lambda throughout harmony iterations. e.g. 'lambda_range' = c(1,1) would make harmony using a fixed lambda = 1.
tau	Protection against overclustering small datasets with large ones. 'tau' is the expected number of cells per cluster.
block.size	What proportion of cells to update during clustering. Between 0 to 1, default 0.05. Larger values may be faster but less accurate.
max.iter.cluster	Maximum number of rounds to run clustering at each round of Harmony.
epsilon.cluster	Convergence tolerance for clustering round of Harmony. Set to -Inf to never stop early.
epsilon.harmony	Convergence tolerance for Harmony. Set to -Inf to never stop early. When 'epsilon.harmony' is set to not NULL, then user-supplied values of 'early_stop' is ignored.

Value

Return a list for '.options' argument of 'RunHarmony'

Examples

```
## If want to set lambda to be fixed to 1, do
## Not run:
RunHarmony(data_meta, meta_data, vars_use,
              .options = harmony_options(lambda = c(1, 1)))

## End(Not run)
```

---

moe_ridge_get_betas *Get beta Utility*

Description

Utility function to get ridge regression coefficients from trained Harmony object

Usage

```
moe_ridge_get_betas(harmonyObj)
```

Arguments

harmonyObj	Trained harmony object. Get this by running RunHarmony function with return_object=TRUE.

Value

Returns nothing, modifies object in place.

---

pbmc.ctrl

Gene expression data of control PBMC from Kang et al. 2017. This contains a sample of 1000 cells from that condition and is used for the Seurat Vignette.

Description

Gene expression data of control PBMC from Kang et al. 2017. This contains a sample of 1000 cells from that condition and is used for the Seurat Vignette.

Usage

```
pbmc.ctrl
```

Format

An object of class `dgCMatrix` with 9015 rows and 1000 columns.

Source

doi:10.1038/nbt.4042

---

pbmc.stim

Gene expression data of stimulated PBMC from Kang et al. 2017. This contains a sample of 1000 cells from that condition and is used for the Seurat Vignette.

Description

Gene expression data of stimulated PBMC from Kang et al. 2017. This contains a sample of 1000 cells from that condition and is used for the Seurat Vignette.

Usage

```
pbmc.stim
```

Format

An object of class dgCMatrix with 9015 rows and 1000 columns.

Source

doi:10.1038/nbt.4042

RunHarmony *Run harmony algorithm generic function*

Description

This is a generic that provides wrappers for Seurat and SingleCellExperiment objects. Also, it allows harmony standalone with a matrix and a metadata dataframe.

Usage

```
RunHarmony(...)

## S3 method for class 'Seurat'
RunHarmony(
  object,
  group.by.vars,
  reduction.use = "pca",
  dims.use = NULL,
  verbose = TRUE,
  reduction.save = "harmony",
  project.dim = TRUE,
  ...
)

## S3 method for class 'SingleCellExperiment'
RunHarmony(
  object,
  group.by.vars,
  dims.use = NULL,
  verbose = TRUE,
  reduction.save = "HARMONY",
  ...
)

HarmonyMatrix(...)
```

Arguments

...	harmony algorithm parameters to be passed on RunHarmony.default
object	SingleCellExperiment with the PCA reducedDim cell embeddings populated
group.by.vars	the name(s) of covariates that harmony will remove its effect on the data.
reduction.use	Name of dimension reduction to use. Default is pca.
dims.use	a vector of indices that allows only selected cell embeddings features to be used.
verbose	enable verbosity
reduction.save	the name of the new slot that is going to be created by harmony. By default, HARMONY.
project.dim	Project dimension reduction loadings. Default TRUE.

Value

Seurat object. Harmony dimensions placed into a new slot in the Seurat object according to the reduction.save. For downstream Seurat analyses, use reduction='harmony'.

SingleCellExperiment object. After running RunHarmony, the corrected cell embeddings can be accessed with reducedDim(object, "Harmony").

RunHarmony.default *Main Harmony interface*

Description

Use this to run the Harmony algorithm directly on cell embedding matrix.

Usage

```
## Default S3 method:
RunHarmony(
  data_mat,
  meta_data,
  vars_use,
  theta = NULL,
  sigma = 0.1,
  lambda = 1,
  nclust = NULL,
  max_iter = 10,
  early_stop = TRUE,
  ncores = 1,
  plot_convergence = FALSE,
  return_object = FALSE,
  verbose = TRUE,
  .options = harmony_options(),
  ...
)
```

Arguments

data_mat	Matrix of cell embeddings. Cells can be rows or columns and will be inferred by the rows of meta_data.
meta_data	Either (1) Dataframe with variables to integrate or (2) vector with labels.
vars_use	If meta_data is dataframe, this defined which variable(s) to remove (character vector).
theta	Diversity clustering penalty parameter. Specify for each variable in vars_use Default theta=2. theta=0 does not encourage any diversity. Larger values of theta result in more diverse clusters.
sigma	Width of soft kmeans clusters. Default sigma=0.1. Sigma scales the distance from a cell to cluster centroids. Larger values of sigma result in cells assigned to more clusters. Smaller values of sigma make soft kmeans cluster approach hard clustering.
lambda	Ridge regression penalty. Default lambda=1. Bigger values protect against over correction. If several covariates are specified, then lambda can also be a vector which needs to be equal length with the number of variables to be corrected. In this scenario, each covariate level group will be assigned the scalars specified by the user. If set to NULL, harmony will determine lambdas automatically and try to minimize overcorrection (beta).
nclust	Number of clusters in model. nclust=1 equivalent to simple linear regression.
max_iter	Maximum number of rounds to run Harmony. One round of Harmony involves one clustering and one correction step.
early_stop	Enable early stopping for harmony. The harmonization process will stop when the change of objective function between corrections drops below 1e-4
ncores	Number of processors to be used for math operations when optimized BLAS is available. If BLAS is not supporting multithreaded then this option has no effect. By default, ncore=1 which runs as a single-threaded process. Although Harmony supports multiple cores, it is not optimized for multithreading. Increase this number for large datasets iff single-core performance is not adequate.
plot_convergence	Whether to print the convergence plot of the clustering objective function. TRUE to plot, FALSE to suppress. This can be useful for debugging.
return_object	(Advanced Usage) Whether to return the Harmony object or only the corrected PCA embeddings.
verbose	Whether to print progress messages. TRUE to print, FALSE to suppress.
.options	Advanced parameters of RunHarmony. This must be the result from a call to 'harmony_options'. See ¿harmony_options' for more details.
...	other parameters that are not part of the API

Value

By default, matrix with corrected PCA embeddings. If return_object is TRUE, returns the full Harmony object (R6 reference class type).

Examples

```
## By default, Harmony inputs a cell embedding matrix
## Not run:
harmony_embeddings <- RunHarmony(cell_embeddings, meta_data, 'dataset')

## End(Not run)

## If PCA is the input, the PCs need to be scaled
data(cell_lines_small)
pca_matrix <- cell_lines_small$scaled_pcs
meta_data <- cell_lines_small$meta_data
harmony_embeddings <- RunHarmony(pca_matrix, meta_data, 'dataset')

## Output is a matrix of corrected PC embeddings
dim(harmony_embeddings)
harmony_embeddings[seq_len(5), seq_len(5)]

## Finally, we can return an object with all the underlying data structures
harmony_object <- RunHarmony(pca_matrix, meta_data, 'dataset', return_object=TRUE)
dim(harmony_object$Y) ## cluster centroids
dim(harmony_object$R) ## soft cluster assignment
dim(harmony_object$Z_corr) ## corrected PCA embeddings
head(harmony_object$O) ## batch by cluster co-occurence matrix
```

Index