Package 'CytOpT'

May 7, 2026

Type Package

Title Optimal Transport for Gating Transfer in Cytometry Data with Domain Adaptation

Version 0.9.8

Date 2025-03-31

Maintainer Boris Hejblum <boris.hejblum@u-bordeaux.fr>

SystemRequirements Python (>= 3.7)

Description Supervised learning from a source distribution (with known segmentation into cell subpopulations)
to fit a target distribution with unknown segmentation. It relies regularized optimal transport to directly
estimate the different cell population proportions from a biological sample characterized with flow cytometry
measurements. It is based on the regularized Wasserstein metric to compare cytometry measurements from
different samples, thus accounting for possible misalignment of a given cell population across sample
(due to technical variability from the technology of measurements). Supervised learning technique based
on the Wasserstein metric that is used to estimate an optimal re-weighting of class proportions in a mixture model Details are presented in Freulon P, Bigot J and Hejblum BP (2023) <doi:10.1214/22-AOAS1660>.

License GPL (>= 2)

Repository CRAN

URL https://sistm.github.io/CytOpT-R/, https://github.com/sistm/CytOpT-R/

Depends R (>= 3.6)

LazyData true

RoxygenNote 7.3.2

Encoding UTF-8

Imports ggplot2 (>= 3.0.0), MetBrewer, patchwork, reshape2, reticulate, stats, testthat (>= 3.0.0)

Suggests rmarkdown, knitr, covr

Config/testthat/edition 3

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Boris Hejblum [aut, cre],
Paul Freulon [aut],
Kalidou Ba [aut, trl]

Date/Publication 2025-04-01 06:50:07 UTC

Contents

`barplot_prop` *Function to display a bland plot in order to visually assess the agreement between CytOpt estimation of the class proportions and the estimate of the class proportions provided through manual gating.*

Description

Function to display a bland plot in order to visually assess the agreement between CytOpt estimation of the class proportions and the estimate of the class proportions provided through manual gating.

Usage

```
barplot_prop(proportions, title = "", xaxis_angle = 45)
```

Arguments

`proportions`	`data.frame` of (true and) estimated proportions from `CytOpt()`
`title`	plot title. Default is `""`, i.e. no title.
`xaxis_angle`	scalar indicating an angle to tilt the labels of x_axis. Default is `45`.

Value

a ggplot object

Examples

```
if(interactive()){

res <- CytOpT(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
              Lab_source = HIPC_Stanford_1228_1A_labels,
              eps = 0.0001, lbd = 0.0001, n_iter = 10000, n_stoc=10,
              step_grad = 10, step = 5, power = 0.99,
              method='minmax')
barplot_prop(res$proportions)

}
```

Bland_Altman *Bland & Altman plot*

Description

Function to display a Bland & Altman plot in order to visually assess the agreement between CytOpt estimation of the class proportions and the estimate of the class proportions provided through manual gating. Requires that either theta_true or Lab_target was provided when running CytOpT().

Usage

```
Bland_Altman(proportions, additional_info_shape = NULL)
```

Arguments

proportions	data.frame of true and estimated proportion returned from CytOpT().
additional_info_shape	vector of additional information to be used for shape in the plot. Not implemented yet.
	#' @return a ggplot object

See Also

CytOpT

Examples

```
if(interactive()){

gold_standard_manual_prop <- c(table(HIPC_Stanford_1369_1A_labels) /
 length(HIPC_Stanford_1369_1A_labels))
res <- CytOpT(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
```

```
              Lab_source = HIPC_Stanford_1228_1A_labels,
              theta_true = gold_standard_manual_prop,
              eps = 0.0001, lbd = 0.0001, n_iter = 10000, n_stoc=10,
              step_grad = 10, step = 5, power = 0.99,
              method='both')
Bland_Altman(res$proportions)

}
```

CytOpT *Function to estimate the type cell proportions in an unclassified cytometry data set denoted X_s by using the classification Lab_source from an other cytometry data set X_s. With this function the computation of the estimate of the class proportions is done with a descent ascent or minmax or two algorithms.*

Description

Function to estimate the type cell proportions in an unclassified cytometry data set denoted X_s by using the classification Lab_source from an other cytometry data set X_s. With this function the computation of the estimate of the class proportions is done with a descent ascent or minmax or two algorithms.

Usage

```
CytOpT(
  X_s,
  X_t,
  Lab_source,
  Lab_target = NULL,
  theta_true = NULL,
  method = c("minmax", "desasc", "both"),
  eps = 1e-04,
  n_iter = 10000,
  power = 0.99,
  step_grad = 10,
  step = 5,
  lbd = 1e-04,
  n_out = 5000,
  n_stoc = 10,
  minMaxScaler = TRUE,
  monitoring = FALSE,
  thresholding = TRUE
)
```

Arguments

X_s	a cytometry dataframe with only d numerical variables for ns observations. The columns correspond to the different biological markers measured. One line corresponds to the cytometry measurements performed on one cell. The classification of this Cytometry data set must be provided with the Lab_source parameters.
X_t	a cytometry dataframe with only d numerical variables for nt observations. The columns correspond to the different biological markers measured. One line corresponds to the cytometry measurements performed on one cell. The CytOpT algorithm targets the cell type proportion in this Cytometry data set
Lab_source	a vector of length ns Classification of the X_s cytometry data set
Lab_target	a vector of length nt Classification of the X_t cytometry data set
theta_true	If available, gold-standard proportions in the target data set X_t derived from manual gating. It allows to assess the gap between the estimate and the gold-standard. Default is NULL, in which case no assessment is performed.
method	a character string indicating which method to use to compute the cytopt, either 'minmax', 'desasc' or 'both' for comparing both Min-max swapping and descent-ascent procedures. Default is 'minmax'.
eps	a float value of regularization parameter of the Wasserstein distance. Default is 1e-04
n_iter	an integer Constant that iterate method select. Default is 10000
power	a float constant the step size policy of the gradient ascent method is step/n^power. Default is 0.99
step_grad	an integer number step size of the gradient descent algorithm of the outer loop. Default is 10
step	an integer constant that multiply the step-size policy. Default is 5
lbd	a float constant that multiply the step-size policy. Default is 1e-04
n_out	an integer number of iterations in the outer loop. This loop corresponds to the gradient descent algorithm to minimize the regularized Wasserstein distance between the source and target data sets. Default is 1000
n_stoc	an integer number of iterations in the inner loop. This loop corresponds to the stochastic algorithm that approximates a maximizer of the semi dual problem. Default is 10
minMaxScaler	a logical flag indicating to whether to scale observations between 0 and 1. Default is TRUE.
monitoring	a logical flag indicating to possibly monitor the gap between the estimated proportions and the manual gold-standard. Default is FALSE.
thresholding	a logical flag indicating whether to threshold negative values. Default is TRUE.

Value

a object of class CytOpt, which is a list of two elements:

- proportions a data.frame with the (optionally true and) estimated proportions for each method
- monitoring a list of estimates over the optimization iterations for each method (listed within)

Examples

```
if(interactive()){

res <- CytOpT(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
              Lab_source = HIPC_Stanford_1228_1A_labels,
              method='minmax')
summary(res)
plot(res)

}
```

`cytopt_desasc_r` *Function to estimate the type cell proportions in an unclassified cytometry data set denoted X_s by using the classification Lab_source from an other cytometry data set X_s. With this function the computation of the estimate of the class proportions is done with a descent ascent algorithm.*

Description

Function to estimate the type cell proportions in an unclassified cytometry data set denoted X_s by using the classification Lab_source from an other cytometry data set X_s. With this function the computation of the estimate of the class proportions is done with a descent ascent algorithm.

Usage

```
cytopt_desasc_r(
  X_s,
  X_t,
  Lab_source,
  theta_true = NULL,
  eps = 1e-04,
  n_out = 5000,
  n_stoc = 10,
  step_grad = 10,
  monitoring = FALSE
)
```

Arguments

X_s	a cytometry dataframe. The columns correspond to the different biological markers tracked. One line corresponds to the cytometry measurements performed on one cell. The classification of this Cytometry data set must be provided with the Lab_source parameters.
X_t	a cytometry dataframe. The columns correspond to the different biological markers tracked. One line corresponds to the cytometry measurements performed on one cell. The CytOpT algorithm targets the cell type proportion in this Cytometry data set

Lab_source	a vector of length n Classification of the X_s cytometry data set
theta_true	If available, gold-standard proportions in the target data set X_t derived from manual gating. It allows to assess the gap between the estimate and the gold-standard. Default is NULL, in which case no assessment is performed.
eps	an float value of regularization parameter of the Wasserstein distance. Default is 1e-04.
n_out	an integer number of iterations in the outer loop. This loop corresponds to the gradient descent algorithm to minimize the regularized Wasserstein distance between the source and target data sets. Default is 5000.
n_stoc	an integer number of iterations in the inner loop. This loop corresponds to the stochastic algorithm that approximates a maximizer of the semi-dual problem. Default is 10.
step_grad	an integer number step size of the gradient descent algorithm of the outer loop. Default is 10.
monitoring	boolean indicating whether Kullback-Leibler divergence should be monitored and store throughout the optimization iterations. Default is FALSE.

Value

A list with the following elements:h_hat

---

cytopt_minmax_r *Function to estimate the type cell proportions in an unclassified cytometry data set denoted X_s by using the classification Lab_source from an other cytometry data set X_s. With this function an additional regularization parameter on the class proportions enables a faster computation of the estimator.*

Description

Function to estimate the type cell proportions in an unclassified cytometry data set denoted X_s by using the classification Lab_source from an other cytometry data set X_s. With this function an additional regularization parameter on the class proportions enables a faster computation of the estimator.

Usage

```
cytopt_minmax_r(
  X_s,
  X_t,
  Lab_source,
  theta_true = NULL,
  eps = 1e-04,
  lbd = 1e-04,
  n_iter = 10000,
```

```
  step = 5,
  power = 0.99,
  monitoring = FALSE
)
```

Arguments

X_s	Cytometry data set. The columns correspond to the different biological markers tracked. One line corresponds to the cytometry measurements performed on one cell. The classification of this Cytometry data set must be provided with the Lab_source parameters.
X_t	Cytometry data set. The columns correspond to the different biological markers tracked. One line corresponds to the cytometry measurements performed on one cell. The CytOpT algorithm targets the cell type proportion in this Cytometry data set.
Lab_source	Classification of the X_s Cytometry data set
theta_true	If available, gold-standard proportions in the target data set X_t derived from manual gating. It allows to assess the gap between the estimate and the gold-standard. Default is NULL, in which case no assessment is performed.
eps	Regularization parameter of the Wasserstein distance
lbd	an float constant that multiply the step-size policy. Default is 1e-04.
n_iter	an integer Constant that iterate method select. Default is 10000.
step	Constant that multiply the step-size policy. Default is 5.
power	the step size policy of the gradient ascent method is step/n^power. Default is 0.99.
monitoring	boolean indicating whether Kullback-Leibler divergence should be monitored and store throughout the optimization iterations. Default is FALSE.

Value

A list with the following elements:Results_Minmax

HIPC_Stanford *HIPC_Stanford data*

Description

HIPC T cell data set from HIPC program for patients 1228 and 1369 (replicate 1A from Stanford).

Usage

```
data(HIPC_Stanford)
```

Format

The data are composed of 4 objects:

`HIPC_Stanford_1228_1A:` a `data.frame` of 31342 cells and 7 markers.

`HIPC_Stanford_1228_1A_labels:` a `factor` vector with the cell type of each of the 31342 observed cells.

`HIPC_Stanford_1369_1A:` a `data.frame` of 33992 cells and 7 markers.

`HIPC_Stanford_1369_1A_labels:` a `factor` vector with the cell type of each of the 33992 observed cells.

Details

This immunophenotyping T cell panel from the Lyoplate HIPC dataset was used as part of the FlowCAP III Lyoplate challenge.

Flow cytometry data set from the HIPC T-cell panel study. In the HIPC T-cell panel study, Flow cytometry was measured in 3 samples for each 3 patients (IDs: 1228, 1349 and 1369) with 3 replicates each (1A, 2B and 3C) in 7 centers (NHLBI, Yale, UCLA, CIMR, Baylor, Stanford and Miami), i.e. 63 data sets in total. Manual gating was performed in the different centers to cluster te observed cells into one of 10 cellular populations:

1. CD8 Effector
2. CD8 Naive
3. CD8 Central Memory
4. CD8 Effector Memory
5. CD8 Activated
6. CD4 Effector
7. CD4 Naive
8. CD4 Central Memory
9. CD4 Effector Memory
10. CD4 Activated

Source

https://www.hipc-dashboard.org/ https://immunespace.org/ https://datatools.immunespace.org/project/HIPC/Lyoplate/begin.view?pageId=study.DATA_ANALYSIS

References

Maecker HT, McCoy JP & Nussenblatt R (2012). Standardizing immunophenotyping for the human immunology project. Nature Reviews Immunology, 12(3):191–200. DOI: 10.1038/nri3158

Finak G, Langweiler M, Jaimes M, Malek M, Taghiyar J, Korin Y, Raddassi K, Devine L, Obermoser G, Pekalski ML, Pontikos N, Diaz A, Heck S, Villanova F, Terrazzini N, Kern F, Qian Y, Stanton R, Wang K, Brandes A, Ramey J, Aghaeepour N, Mosmann T, Scheuermann RH, Reed E, Palucka K, Pascual V, Blomberg BB, Nestle F, Nussenblatt RB, Brinkman RR, Gottardo R, Maecker H & McCoy JP (2016). Standardizing Flow Cytometry Immunophenotyping Analysis from the Human ImmunoPhenotyping Consortium. Scientific Reports. 10(6):20686. DOI: 10.1038/srep20686.

KL_plot *Kullback-Leibler divergence plot*

Description

A plotting function for displaying Kullback-Liebler (KL) divergence across iterations of the optimization algorithm(s).

Usage

```
KL_plot(
  monitoring,
  n_0 = 10,
  n_stop = 1000,
  title = "Kullback-Liebler divergence trace"
)
```

Arguments

monitoring	list of monitoring estimates from CytOpt() output.
n_0	first iteration to plot. Default is 10.
n_stop	last iteration to plot. Default is 1000.
title	plot title. Default is "Kullback-Liebler divergence trace".

Value

a ggplot object

Examples

```
if(interactive()){

gold_standard_manual_prop <- c(table(HIPC_Stanford_1369_1A_labels) /
 length(HIPC_Stanford_1369_1A_labels))
res <- CytOpT(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
              Lab_source = HIPC_Stanford_1228_1A_labels,
              theta_true = gold_standard_manual_prop,
              eps = 0.0001, lbd = 0.0001, n_iter = 10000, n_stoc=10,
              step_grad = 10, step = 5, power = 0.99,
              method='both', monitoring = TRUE)
plot(res)

}
```

`Label_Prop_sto_r` *Computes a classification on the target data*

Description

Computes a classification on the target data thanks to the approximation of the transport plan and the classification of the source data. Transport plan is approximated with the stochastic algorithm.

Usage

```
Label_Prop_sto_r(
  X_s,
  X_t,
  Lab_source,
  eps = 1e-04,
  const = 0.1,
  n_iter = 4000,
  minMaxScaler = TRUE,
  monitoring = TRUE,
  thresholding = TRUE
)
```

Arguments

`X_s`	a cytometry dataframe. The columns correspond to the different biological markers tracked. One line corresponds to the cytometry measurements performed on one cell. The classification of this Cytometry data set must be provided with the Lab_source parameters.
`X_t`	a cytometry dataframe. The columns correspond to the different biological markers tracked. One line corresponds to the cytometry measurements performed on one cell. The CytOpT algorithm targets the cell type proportion in this Cytometry data set
`Lab_source`	a vector of length `n` Classification of the X_s cytometry data set
`eps`	an float value of regularization parameter of the Wasserstein distance. Default is `1e-04`
`const`	an float constant. Default is `1e-01`
`n_iter`	an integer Constant that iterate method select. Default is `4000`
`minMaxScaler`	a logical flag indicating to possibly Scaler
`monitoring`	a logical flag indicating to possibly monitor the gap between the estimated proportions and the manual gold-standard. Default is `FALSE`
`thresholding`	a logical flag.

Value

a `ggplot` object

a vector of length `nrow(X_t)` with the propagated labels

Examples

```
if(interactive()){

res <- Label_Prop_sto_r(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
             Lab_source = HIPC_Stanford_1228_1A_labels)

}
```

`plot.CytOpt` *CytOpt plot*

Description

plot S3 method for CytOpt object

Usage

```
## S3 method for class 'CytOpt'
plot(x, ...)
```

Arguments

`x`	an object of class `CytOpt` to plot.
`...`	further arguments passed to or from other methods. Not implemented.

Value

a `ggplot` object, potentially composed through `patchwork`

Examples

```
if(interactive()){

res <- CytOpT(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
             Lab_source = HIPC_Stanford_1228_1A_labels,
             eps = 0.0001, lbd = 0.0001, n_iter = 10000, n_stoc=10,
             step_grad = 10, step = 5, power = 0.99,
             method='minmax')
plot(res)

}
```

print.CytOpt *CytOpt print*

Description

print S3 method for CytOpt object

Usage

```
## S3 method for class 'CytOpt'
print(x, ...)
```

Arguments

x	an object of class CytOpt to print.
...	further arguments passed to or from other methods. Not implemented.

Value

the proportions data.frame from x

Examples

```
if(interactive()){

res <- CytOpT(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
              Lab_source = HIPC_Stanford_1228_1A_labels,
              eps = 0.0001, lbd = 0.0001, n_iter = 10000, n_stoc=10,
              step_grad = 10, step = 5, power = 0.99,
              method='minmax')
print(res)

}
```

print.summary.CytOpt *CytOpt print summary*

Description

print S3 method for summary.CytOpt object

Usage

```
## S3 method for class 'summary.CytOpt'
print(x, ...)
```

Arguments

x	an object of class summary.CytOpt to print.
...	further arguments passed to or from other methods. Not implemented.

---

summary.CytOpt *CytOpt summary*

Description

summary S3 method for CytOpt object

Usage

```
## S3 method for class 'CytOpt'
summary(object, ...)
```

Arguments

object	an object of class CytOpt to summarized.
...	further arguments passed to or from other methods. Not implemented.

Value

a list object

Examples

```
if(interactive()){

res <- CytOpT(X_s = HIPC_Stanford_1228_1A, X_t = HIPC_Stanford_1369_1A,
              Lab_source = HIPC_Stanford_1228_1A_labels,
              eps = 0.0001, lbd = 0.0001, n_iter = 10000, n_stoc=10,
              step_grad = 10, step = 5, power = 0.99,
              method='minmax', monitoring=TRUE)
summary(res)

}
```

Index