Package 'Polytect'

November 12, 2026

Title An R package for digital data clustering

Version 1.4.0

Description Polytect is an advanced computational tool designed for the analysis of multi-color digital PCR data. It provides automatic clustering and labeling of partitions into distinct groups based on clusters first identified by the flowPeaks algorithm. Polytect is particularly useful for researchers in molecular biology and bioinformatics, enabling them to gain deeper insights into their experimental results through precise partition classification and data visualization.

biocViews ddPCR, Clustering, MultiChannel, Classification

License Artistic-2.0

URL https://github.com/emmachenlingo/Polytect

BugReports https://github.com/emmachenlingo/Polytect/issues

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.4.0)

Imports stats, utils, grDevices, mvtnorm, sn, dplyr, flowPeaks, ggplot2, tidyverse, cowplot, mlrMBO, DiceKriging, smoof, ParamHelpers, lhs, rgenoud, BiocManager

Suggests testthat (>= 3.0.0), knitr, rmarkdown, ddPCRclust

VignetteBuilder knitr

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/Polytect

git_branch RELEASE_3_23

git_last_commit 56d5176

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Yao Chen [aut, cre] (ORCID: <https://orcid.org/0000-0001-8172-3996>)

Maintainer Yao Chen <emmachentar@live.com>

Contents

approxSilhouette 2
BPV 3
CA 4
cluster_selection 4
CNV5plex 5
CNV6plex 5
combined_vectors 6
compute_tmp_matrix 7
conc_cal 7
estep 8
fp_search 9
GMM_init 9
HIV 10
HMM_merge 10
HR 11
LR 12
MM 12
mstep_cov 13
mstep_mu 13
polytect_clust 15
polytect_merge 16
polytect_plot 17
polytect_summary 17
silhouette_coef 18
sil_plot 19

Index **20**

`approxSilhouette` *Internal Function 2*

Description

This function outputs silhouette coefficients.

Usage

```
approxSilhouette(x, clusters)
```

Arguments

x	A matrix of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.
clusters	cluster labels

Value

A data frame of silhouette coefficients for each partition.

BPV *BPV data*

Description

A 3-color dPCR data of bovine papilloma virus assay

Usage

```
data(BPV)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

channel3 fluorescence intensities of color 3

Examples

```
data(BPV)
head(BPV)
```

CA *CA data*

Description

2-color competitive assay of competition BRAF V600E assay with 1% mutant

Usage

```
data(CA)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel. data is not orthogonal.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

Examples

```
data(CA)
head(CA)
```

cluster_selection *Internal Function 11*

Description

This function outputs all combinations of primary targets

Usage

```
cluster_selection(cluster_num)
```

Arguments

cluster_num	The expected maximum number of clusters

Value

A matrix of all combinations of primary targets

CNV5plex *CNV 5-plex data*

Description

CNV 5-plex universal probes

Usage

```
data(CNV5plex)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

channel3 fluorescence intensities of color 3

channel4 fluorescence intensities of color 4

channel5 fluorescence intensities of color 5

Examples

```
data(CNV5plex)
head(CNV5plex)
```

CNV6plex *CNV 6-plex data*

Description

CNV 6-plex universal probes

Usage

```
data(CNV6plex)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

channel3 fluorescence intensities of color 3

channel4 fluorescence intensities of color 4

channel5 fluorescence intensities of color 5

channel6 fluorescence intensities of color 6

Examples

```
data(CNV6plex)
head(CNV6plex)
```

---

combined_vectors *Internal Function 4*

Description

This function outputs vectors and weights that will be used in EM algorithm

Usage

```
combined_vectors(coefs, mus, cluster_num, dim_data)
```

Arguments

coefs	coefs The coefficients to adjust for the expected cluster centers. The default is 1 which can be used for common assay designs and has to be modified for special assays such as competing assays.
mus	The cluster centers of primary targets
cluster_num	The expected maximum number of clusters.
dim_data	dimension of the dataset

Value

A list of vectors and weights

compute_tmp_matrix *Internal Function 6*

Description

This function compute the necessary elements for estep function

Usage

```
compute_tmp_matrix(g, k, cluster_num, mg, log_pih, mug_t, muh_t, covh, covg)
```

Arguments

g	cluster index
k	cluster index
cluster_num	The expected maximum number of clusters
mg	cluster sizes of base clustering result
log_pih	log pih (the probability of cluster g belonging at level l+1 to cluster h at level l)
mug_t	the transposed matrix of cluster centers at level l+1
muh_t	the transposed matrix of cluster centers at level l
covh	the covariance matrix of clusters at level l
covg	the covariance matrix of clusters at level l+1

Value

A vector of intermediate values for zi calculation in estep function

conc_cal *concentration calculation function*

Description

This function takes a data frame of fluorescence intensities and partition clusters as input. It can be results from polytect_clust or polytect_merge. It will give the target concentration as output.

Usage

```
conc_cal(df_data, cluster_num, sampvol = 0.91, volmix = 20, voltemp = 20)
```

Arguments

df_data	A data frame containing partition fluorescence intensities and corresponding cluster label. This can be the output of polytect_merge or any data frame containing the above information.
cluster_num	the expected number of clusters
sampvol	The sample volume in microliters (µL)
volmix	The volume of the mixture
voltemp	The volume of the template

Value

a data frame of target concentration.

Examples

```
data(HR)
df_data<-polytect_clust(HR,4)
conc_cal(df_data,4)
```

---

estep *Internal Function 7*

Description

This function calculates zi in E-step of EM algorithm

Usage

```
estep(g_clusternum, cluster_num, pih, muh, covh, mg, mug, covg)
```

Arguments

g_clusternum	cluster labels from base clustering
cluster_num	The expected maximum number of clusters
pih	the probability of cluster g belonging at level l+1 to cluster h at level l
muh	the matrix of cluster centers at level l
covh	the covariance matrix of clusters at level l
mg	cluster sizes of base clustering result
mug	the matrix of cluster centers at level l+1
covg	the covariance matrix of clusters at level l+1

Value

zi for estep in EM algorithm

fp_search *Internal Function 3*

Description

This function optimizes parameters of flowPeaks

Usage

```
fp_search(data, cluster_num = 16)
```

Arguments

data	A matrix of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.
cluster_num	The expected maximum number of clusters

Value

A vector containing the optimal parameters found by the algorithm

GMM_init *Internal Function 5*

Description

This function intialize the parameters for the main clustering function

Usage

```
GMM_init(data, cluster_num, base_clust, coefs)
```

Arguments

data	A matrix or data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.
cluster_num	The expected maximum number of clusters
base_clust	The results of base clustering
coefs	The coefficients to adjust for the expected cluster centers. The default is 1 which can be used for common assay designs and has to be modified for special assays such as competing assays.

Value

A list of initial parameters for the EM algorithm

HIV *HIV data*

Description

A 4-color dPCR data of intact HIV-1 proviruses

Usage

```
data(HIV)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

channel3 fluorescence intensities of color 3

channel4 fluorescence intensities of color 4

Source

https://www.biorxiv.org/content/10.1101/2023.08.18.553846v1

Examples

```
data(HIV)
head(HIV)
```

HMM_merge *Internal Function 10*

Description

This function merges the excess clusters given by the base clustering

Usage

```
HMM_merge(
  data,
  cluster_num,
  base_clust,
  eps = 10^(-10),
  max_iter = 1000,
  lambdas = rep(2, 2),
  coefs = rep(1, 2)
)
```

Arguments

`data`	A matrix or data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.
`cluster_num`	The expected maximum number of clusters
`base_clust`	base clustering results before merging
`eps`	the convergence threshold
`max_iter`	maximum number of iterations
`lambdas`	The penalty terms for the deviation from the expected cluster centers. Higher `lambdas` penalizes the deviation more.
`coefs`	The coefficients to adjust for the expected cluster centers. The default is 1 which can be used for common assay designs and has to be modified for special assays such as competing assays.

Value

A list of membership probability, cluster center, merging probability

---

HR *HR data*

Description

A high-resolution 2-color dPCR data of RPP30 genomic DNA assay

Usage

```
data(HR)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel. good separation but some crosstalk.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

Source

https://pubmed.ncbi.nlm.nih.gov/33992770/

Examples

```
data(HR)
head(HR)
```

LR *LR data*

Description

A low-resolution 2-color dPCR data of development of genotyping assays for plants various

Usage

```
data(LR)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel. barely separable on x-axis.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

Examples

```
data(LR)
head(LR)
```

MM *MM data*

Description

A multi-mode 2-color dPCR data of HIV gBlock sequences

Usage

```
data(MM)
```

Format

A data frame of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel. obvious multimodality.

channel1 fluorescence intensities of color 1

channel2 fluorescence intensities of color 2

Source

https://pubmed.ncbi.nlm.nih.gov/37827643/

Examples

```
data(MM)
head(MM)
```

`mstep_cov` *Internal Function 9*

Description

This function calculates mu in M-step of EM algorithm

Usage

```
mstep_cov(cluster_num, dim_data, g_clusternum, zi, mg, covg, mug, muh)
```

Arguments

`cluster_num`	The expected maximum number of clusters
`dim_data`	the dimension of the dataset
`g_clusternum`	cluster labels from base clustering
`zi`	the expected log-likelihood found on the E step
`mg`	cluster sizes of base clustering result
`covg`	the covariance matrix of clusters at level l+1
`mug`	the matrix of cluster centers at level l+1
`muh`	the matrix of cluster centers at level l

Value

covh the covariance matrix of clusters at level l in the EM algorithm

---

`mstep_mu` *Internal Function 8*

Description

This function calculates mu in M-step of EM algorithm

Usage

```
mstep_mu(
  zi,
  g_clusternum,
  dim_data,
  cluster_num,
  weights,
  muh,
  covh,
  mg,
  mug,
  neg_assum,
  lambdas,
  coefs
)
```

Arguments

zi	the expected log-likelihood found on the E step
g_clusternum	cluster labels from base clustering
dim_data	the dimension of the dataset
cluster_num	The expected maximum number of clusters
weights	combinations of coefficients of the cluster centers
muh	the matrix of cluster centers at level l
covh	the covariance matrix of clusters at level l
mg	cluster sizes of base clustering result
mug	the matrix of cluster centers at level l+1
neg_assum	the estimated cluster center of negative population
lambdas	The penalty terms for the deviation from the expected cluster centers. Higher lambdas penalizes the deviation more.
coefs	The coefficients to adjust for the expected cluster centers. The default is 1 which can be used for common assay designs and has to be modified for special assays such as competing assays.

Value

muh the cluster centers at level l in the EM algorithm

`polytect_clust` *Main function for clustering*

Description

This is the main function for clustering. The function will start with flowPeaks, then merge the excess clusters. It will return a data frame of fluorescence intensities and partition labels.

Usage

```
polytect_clust(
  data,
  cluster_num,
  fp_par = "default",
  fp_optim = c(0.1, 1, 1.5),
  lambdas = rep(2, 64 - log2(64)),
  coefs = rep(1, 6)
)
```

Arguments

`data`	A matrix of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.
`cluster_num`	The expected maximum number of clusters.
`fp_par`	The parameters for flowPeaks. `fp_par`=c("default","manual","auto"). When "default" is chosen, the default parameters of flowPeaks will be used. With "manual", you have to fill in `fp_optim`.
`fp_optim`	The paramters for flowPeaks that users have to fill in manually when `fp_par` is set at "manual".
`lambdas`	The penalty terms for the deviation from the expected cluster centers. Higher `lambdas` penalizes the deviation more.
`coefs`	The coefficients to adjust for the expected cluster centers. The default is 1 which can be used for common assay designs and has to be modified for special assays such as competing assays.

Value

A data frame containing the original fluorescence intensity and the cluster labels.

Examples

```
data(HR)
head(polytect_clust(HR, 4))
```

polytect_merge *Function for merging*

Description

This function takes the clustering result as input. Users can first perform any clustering algorithm, then use this function. It will return a data frame of fluorescence intensities and partition labels.

Usage

```
polytect_merge(
  data,
  cluster_num,
  base_clust,
  lambdas = rep(2, 64 - log2(64)),
  coefs = rep(1, 6)
)
```

Arguments

data	A matrix of fluorescence intensities in each channel. Each row represents each partitions, and each column each channel.
cluster_num	The expected maximum number of clusters.
base_clust	A list that contains partition labels given by initial clustering.
lambdas	The penalty terms for the deviation from the expected cluster centers. Higher lambdas penalizes the deviation more.
coefs	The coefficients to adjust for the expected cluster centers. The default is 1 which can be used for common assay designs and has to be modified for special assays such as competing assays.

Value

A data frame containing the original fluorescence intensity and the cluster labels.

Examples

```
data(HR)
dist_matrix <- dist(HR)
hc <- hclust(dist_matrix, method = "ward.D2")
hc_clusters <- cutree(hc, k = 6)
base_clust<-list()
base_clust$cluster<-hc_clusters
head(polytect_merge(HR, 4, base_clust))
```

polytect_plot *Plotting function for clustering results*

Description

This function takes results from polytect_clust and polytect_merge, or a data frame containing flurescence intensities and partition labels. It will output all combination of 2-color plots.

Usage

```
polytect_plot(df_data, cluster_num, cluster_selected = TRUE)
```

Arguments

df_data	A data frame containing partition fluorescence intensities and corresponding cluster label. This can be the output of polytect_clust and polytect_merge or any data frame containing the above information.
cluster_num	the expected number of clusters
cluster_selected	Indicator of whether all the clusters are present in the plots. If TRUE, then only selected ones (the ones only positive in the selected 2 dimensions) are shown. The default value is "TRUE".

Value

2-color plots.

Examples

```
data(HR)
df_data<-polytect_clust(HR,4)
polytect_plot(df_data,4)
```

polytect_summary *Summary function*

Description

This function takes results from polytect_clust and polytect_merge, or a data frame containing flurescence intensities and partition labels. It will summarise cluster centers, cluster sizes and cluster silhouette coefficients.

Usage

```
polytect_summary(df_data)
```

Arguments

df_data	A data frame containing partition fluorescence intensities and corresponding cluster label. This can be the output of polytect_clust and polytect_merge or any data frame containing the above information.

Value

a data frame of the summary of cluster centers, cluster sizes and cluster silhouette coefficients.

Examples

```
data(HR)
df_data<-polytect_clust(HR,4)
polytect_summary(df_data)
```

---

silhouette_coef *Internal Function 1*

Description

This function outputs silhouette coefficients.

Usage

```
silhouette_coef(data, clustering, plot = FALSE)
```

Arguments

data	A data frame containing standardized partition fluorescence intensities and corresponding cluster label.
clustering	cluster labels
plot	TRUE or FALSE, whether a plot should be shown. The default value is "FALSE".

Value

A list of silhouette coefficients for each partition and the mean silhouette coefficients for each cluster.

sil_plot *Plotting function for silhouette coefficients*

Description

This function takes results from polytect_clust and polytect_merge, or a data frame containing flurescence intensities and partition labels. It will output the silhouette coefficients of each cluster.

Usage

```
sil_plot(df_data)
```

Arguments

df_data	A data frame containing partition fluorescence intensities and corresponding cluster label. This can be the output of polytect_clust and polytect_merge or any data frame containing the above information.

Value

plot of silhouette coefficients for each cluster.

Examples

```
data(HR)
df_data<-polytect_clust(HR,4)
sil_plot(df_data)
```

Index

∗ **datasets**
  BPV, 3
  CA, 4
  CNV5plex, 5
  CNV6plex, 5
  HIV, 10
  HR, 11
  LR, 12
  MM, 12
∗ **internal**
  approxSilhouette, 2
  cluster_selection, 4
  combined_vectors, 6
  compute_tmp_matrix, 7
  estep, 8
  fp_search, 9
  GMM_init, 9
  HMM_merge, 10
  mstep_cov, 13
  mstep_mu, 13
  silhouette_coef, 18

approxSilhouette, 2

BPV, 3

CA, 4
cluster_selection, 4
CNV5plex, 5
CNV6plex, 5
combined_vectors, 6
compute_tmp_matrix, 7
conc_cal, 7

estep, 8

fp_search, 9

GMM_init, 9

HIV, 10
HMM_merge, 10
HR, 11

LR, 12

MM, 12
mstep_cov, 13
mstep_mu, 13

polytect_clust, 15
polytect_merge, 16
polytect_plot, 17
polytect_summary, 17

sil_plot, 19
silhouette_coef, 18