

# Package ‘Polytect’

May 9, 2026

**Title** An R package for digital data clustering

**Version** 1.5.0

**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, smooth, 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** devel

**git\_last\_commit** 48dacdd

**git\_last\_commit\_date** 2026-04-28

**Repository** Bioconductor 3.24

**Date/Publication** 2026-05-08

**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 . . . . .	3
cluster_selection . . . . .	4
CNV5plex . . . . .	4
CNV6plex . . . . .	5
combined_vectors . . . . .	5
compute_tmp_matrix . . . . .	6
conc_cal . . . . .	6
estep . . . . .	7
fp_search . . . . .	7
GMM_init . . . . .	8
HIV . . . . .	8
HMM_merge . . . . .	9
HR . . . . .	10
LR . . . . .	10
MM . . . . .	11
mstep_cov . . . . .	11
mstep_mu . . . . .	12
polytect_clust . . . . .	13
polytect_merge . . . . .	14
polytect_plot . . . . .	15
polytect_summary . . . . .	15
silhouette_coef . . . . .	16
sil_plot . . . . .	16
<b>Index</b>	<b>18</b>

---

approxSilhouette	<i>Internal Function 2</i>
------------------	----------------------------

---

### 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	<i>Internal Function 11</i>
-------------------	-----------------------------

---

**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	<i>CNV 5-plex data</i>
----------	------------------------

---

**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	<i>CNV 6-plex data</i>
----------	------------------------

---

**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	<i>Internal Function 4</i>
------------------	----------------------------

---

**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  $z_i$  in E-step of EM algorithm

**Usage**

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

**Arguments**

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

**Value**

$z_i$  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	<i>Internal Function 5</i>
----------	----------------------------

---

**Description**

This function initialize 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	<i>HIV data</i>
-----	-----------------

---

**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	<i>HR data</i>
----	----------------

---

**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	<i>LR data</i>
----	----------------

---

**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	<i>MM data</i>
----	----------------

---

**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	<i>Internal Function 9</i>
-----------	----------------------------

---

**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	<i>Main function for clustering</i>
----------------	-------------------------------------

---

## 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. <code>fp_par=c("default","manual","auto")</code> . When "default" is chosen, the default parameters of flowPeaks will be used. With "manual", you have to fill in <code>fp_optim</code> .
fp_optim	The paramters for flowPeaks that users have to fill in manually when <code>fp_par</code> is set at "manual".
lambdas	The penalty terms for the deviation from the expected cluster centers. Higher <code>lambdas</code> 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	<i>Function for merging</i>
----------------	-----------------------------

---

### 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	<i>Plotting function for clustering results</i>
---------------	---

---

**Description**

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

**Usage**

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

**Arguments**

<code>df_data</code>	A data frame containing partition fluorescence intensities and corresponding cluster label. This can be the output of <code>polytect_clust</code> and <code>polytect_merge</code> or any data frame containing the above information.
<code>cluster_num</code>	the expected number of clusters
<code>cluster_selected</code>	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	<i>Summary function</i>
------------------	-------------------------

---

**Description**

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

**Usage**

```
polytect_summary(df_data)
```

**Arguments**

<code>df_data</code>	A data frame containing partition fluorescence intensities and corresponding cluster label. This can be the output of <code>polytect_clust</code> and <code>polytect_merge</code> 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	<i>Internal Function 1</i>
-----------------	----------------------------

---

**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	<i>Plotting function for silhouette coefficients</i>
----------	--

---

**Description**

This function takes results from polytect\_clust and polytect\_merge, or a data frame containing fluorescence 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, [3](#)  
CNV5plex, [4](#)  
CNV6plex, [5](#)  
HIV, [8](#)  
HR, [10](#)  
LR, [10](#)  
MM, [11](#)

## \* internal

approxSilhouette, [2](#)  
cluster\_selection, [4](#)  
combined\_vectors, [5](#)  
compute\_tmp\_matrix, [6](#)  
estep, [7](#)  
fp\_search, [7](#)  
GMM\_init, [8](#)  
HMM\_merge, [9](#)  
mstep\_cov, [11](#)  
mstep\_mu, [12](#)  
silhouette\_coef, [16](#)

[approxSilhouette, 2](#)

[BPV, 3](#)

[CA, 3](#)

[cluster\\_selection, 4](#)

[CNV5plex, 4](#)

[CNV6plex, 5](#)

[combined\\_vectors, 5](#)

[compute\\_tmp\\_matrix, 6](#)

[conc\\_cal, 6](#)

[estep, 7](#)

[fp\\_search, 7](#)

[GMM\\_init, 8](#)

[HIV, 8](#)

[HMM\\_merge, 9](#)

[HR, 10](#)

[LR, 10](#)

[MM, 11](#)

[mstep\\_cov, 11](#)

[mstep\\_mu, 12](#)

[polytect\\_clust, 13](#)

[polytect\\_merge, 14](#)

[polytect\\_plot, 15](#)

[polytect\\_summary, 15](#)

[sil\\_plot, 16](#)

[silhouette\\_coef, 16](#)