

# Package: demuxmix (via r-universe)

June 15, 2026

**Title** Demultiplexing oligo-barcoded scRNA-seq data using regression mixture models

**Version** 1.14.0

**Date** 2024-01-29

**Description** A package for demultiplexing single-cell sequencing experiments of pooled cells labeled with barcode oligonucleotides. The package implements methods to fit regression mixture models for a probabilistic classification of cells, including multiplet detection. Demultiplexing error rates can be estimated, and methods for quality control are provided.

**License** Artistic-2.0

**Depends** R (>= 4.0.0)

**Imports** stats, MASS, Matrix, ggplot2, gridExtra, methods

**Suggests** BiocStyle, cowplot, DropletUtils, knitr, reshape2, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**biocViews** SingleCell, Sequencing, Preprocessing, Classification, Regression

**BiocType** Software

**URL** <https://github.com/huklein/demuxmix>

**BugReports** <https://github.com/huklein/demuxmix/issues>

**Encoding** UTF-8

**RoxygenNote** 7.2.2

**Config/testthat/edition** 3

**Repository** <https://bioc-release.r-universe.dev>

**Date/Publication** 2026-04-28 12:58:56 UTC

**RemoteUrl** <https://github.com/bioc/demuxmix>

**RemoteRef** RELEASE\_3\_23

**RemoteSha** 97ef8198213776959570e0673bde1125a61f9a17

## Contents

csf . . . . .	2
demuxmix . . . . .	3
Demuxmix-class . . . . .	6
dmmClassify . . . . .	8
dmmOverlap . . . . .	9
dmmSimulateHto . . . . .	11
plotDmmHistogram . . . . .	12
plotDmmPosteriorP . . . . .	14
plotDmmScatter . . . . .	15
summary . . . . .	16
<b>Index</b>	<b>18</b>

---

csf	<i>Hashtag oligonucleotide (HTO) counts from 2,590 droplets</i>
-----	---

---

### Description

Cerebral spinal fluid (CSF) cells and peripheral blood mononuclear cells (PBMCs) were pooled and prepared for single-cell sequencing using the 10x Chromium System. Due to the low numbers of cells obtained from CSF, only the PBMCs but not the CSF cells were stained using oligonucleotide-labeled antibodies (BioLegend TotalSeq-A0257). CSF cells and PBMCs in this dataset were obtained from two genetically diverse individuals so that genetic demultiplexing could be used to validate the HTO-based demultiplexing. Genetic demultiplexing was performed with freemuxlet, which is part of the popscl software package.

### Usage

```
data(csf)
```

### Format

A data frame with 2,590 rows and 4 variables:

**HTO** Number of HTO counts observed

**NumGenes** Number of genes detected in the cell

**freemuxlet** Genetic demultiplexing result

**freemuxlet.prob** Posterior probability from genetic demultiplexing in logarithmic scale

Raw sequencing data was aligned and processed using Cell Ranger 6.0.1. All droplets that passed Cell Ranger's default filtering step were read in. Genes with at least one read were considered as detected. Since Cell Ranger's threshold to identify non-empty droplets is relatively lenient, some droplets have as few as 30-50 genes detected. For most analyses, it is recommended to remove droplets with less than about 200 detected genes before demultiplexing.

## Source

Center for Translational and Computational Neuroimmunology, Department of Neurology, Columbia University Irving Medical Center, contact: Hans-Ulrich Klein (hk2948@cumc.columbia.edu)

## Examples

```
data(csf)
csf <- csf[csf$NumGenes >= 200, ]
hto <- t(matrix(csf$HTO, dimnames = list(rownames(csf), "HTO")))
dmm <- demuxmix(hto, model = "naive")
summary(dmm)
certain <- exp(csf$freemuxlet.prob) >= 0.999
table(dmmClassify(dmm)$HTO[certain], csf$freemuxlet[certain])
```

---

demuxmix

*Demultiplexing using mixture models*

---

## Description

This method uses mixture models as probabilistic framework to assign droplets to hashtags and to identify multiplets based on counts obtained from a hashtag oligonucleotide (HTO) library. If the numbers of detected genes from the corresponding RNA library are passed as second argument, regression mixture models may be used, which often improves the classification accuracy by leveraging the relationship between HTO and RNA read counts.

## Usage

```
demuxmix(  
  hto,  
  rna,  
  pAcpt = 0.9^nrow(hto),  
  model = "auto",  
  alpha = 0.9,  
  beta = 0.9,  
  correctTails = TRUE,  
  tol = 10^-5,  
  maxIter = 100,  
  k.hto = 1.5,  
  k.rna = 1.5,  
  clusterInit = list()  
)
```

## Arguments

**hto** A matrix of HTO counts where each row corresponds to a hashtag and each column to a droplet. The matrix must have unique row names.

<code>rna</code>	An optional numeric vector with the number of genes detected in the RNA library for each droplet. Same length as columns in <code>hto</code> . If missing, parameter <code>model</code> must be set to "naive".
<code>pAcpt</code>	Acceptance probability that must be reached in order to assign a droplet to a hashtag. Droplets with lower probabilities are classified as "uncertain". This parameter can be changed after running <code>demuxmix</code> by applying <code>pAcpt&lt;-</code> to the returned object.
<code>model</code>	A character specifying the type of mixture model to be used. Either "naive", "regpos", "reg" or "auto". The last three options require parameter <code>rna</code> to be specified. "auto" selects the best model based on the classification error probability summed over all droplets.
<code>alpha</code>	Threshold defining the left tail of the mixture distribution where droplets should not be classified as "positive". Threshold must be between 0 and 1. See details.
<code>beta</code>	Threshold for defining the right tail of the mixture distribution where droplets should not be classified as "negative". Threshold must be between 0 and 1. See details.
<code>correctTails</code>	If TRUE, droplets meeting the threshold defined by <code>alpha</code> ( <code>beta</code> ) are classified as "negative" ("positive") even if the mixture model suggests a different classification. See details.
<code>tol</code>	Convergence criterion for the EM algorithm used to fit the mixture models. The algorithm stops when the relative increase of the log likelihood is less than or equal to <code>tol</code> .
<code>maxIter</code>	Maximum number of iterations for the EM algorithm and for the alternating iteration process fitting the NB regression models within each EM iteration.
<code>k.hto</code>	Factor to define outliers in the HTO counts. Among droplets positive for the hashtag based on initial clustering, HTO counts larger than the 0.75 quantile + <code>k.hto</code> * IQR are considered outliers. See details.
<code>k.rna</code>	Factor to define outliers in the numbers of detected genes. Numbers of detected genes larger than the 0.75 quantile + <code>k.rna</code> * IQR are considered outliers. See details.
<code>clusterInit</code>	Optional list of numeric vectors to manually specify the droplet to component assignment used to initialize the EM algorithm. The name of each list element must match a row name of <code>hto</code> . The length of each element must match the number of columns of <code>hto</code> . Only the values 1 and 2 are allowed, where 1 indicates the respective droplet belongs to the negative component with lower mean count.

## Details

The single cell dataset should undergo basic filtering to remove low quality or empty droplets before calling this function, but the HTO counts should not be transformed or pre-processed otherwise. The number of detected genes passed via the optional argument `rna` is typically defined as the number of genes in the RNA library with at least one read.

The method fits a two-component negative binomial mixture model for each hashtag. The type of mixture model used can be specified by `model`. "naive" fits a standard mixture model. "reg"

fits a regression mixture model using the given number of detected genes (`rna`) as covariate in the regression model. "regpos" uses a regression model only for the positive but not for the negative component. If model is set to "auto", all three models are fitted and the model with the lowest posterior classification error probability summed over all droplets is selected. Details are stored in the slot `modelSelection` of the returned object. In most real HTO datasets, regression mixture models outperform the naive mixture model.

The `demuxmix` method consists of 3 steps, which can be tuned by the respective parameters. The default settings work well for a wide range of datasets and usually do not need to be adapted unless any issues arise during model fitting and quality control. An exception is the acceptance probability `pAcpt`, which may be set to smaller or larger value depending on the desired trade-off between number of unclassified/discarded droplets and expected error rate. Steps 1 and 2 are executed for each HTO separately; step 3 classifies the droplets based on the results from all HTOs. Therefore, parameters affecting steps 1 and 2 (incl. `model`) can be specified for each HTO using a vector with one element per HTO. Shorter vectors will be extended.

1. Preprocessing (`k.hto`, `k.rna`). Droplets are clustered into a negative and a positive group based on the HTO counts using k-means. Droplets in the positive group with HTO counts larger than the 0.75 quantile + `k.hto` times the IQR of the HTO counts in the positive group are marked as outliers. Outliers are still classified but will not be used to fit the mixture model for this HTO in step 2. If the parameter `rna` is given and the `model` is "reg" or "regpos", all droplets (both groups) with number of detected genes larger than the 0.75 quantile + `k.rna` times the IQR are marked as outliers, too, since these cells could affect the fitting of the regression model negatively. If more than 15% of the cells are marked as outliers, a warning message is printed and larger values for `k.hto` and `k.rna` might be preferable. If the model fit seems to be affected by a few large values (very high variance of the positive component), smaller values should be chosen. On rare occasions, k-means clustering can result in inadequate clusters, and the derived distributional parameters are invalid. Poor clustering can be observed if (i) the HTO failed and the distribution is not bimodal or (ii) the fraction of positive cells tagged by the HTO is very small. An error message is displayed, and if (ii) is determined as the cause, an initial manual assignment can be specified by `clusterInit` to bypass the k-means clustering.
2. Model fitting (`model`, `alpha`, `beta`, `correctTails`, `tol`, `maxIter`). An EM algorithm is used to fit the mixture model to the HTO counts which were not marked as outliers in step 1. `maxIter` defines the maximum number of iterations of the EM algorithm, and, if `model` is "reg", "regpos" or "auto", it also defines the maximum number of iterations to fit the negative binomial regression models within each EM iteration. `tol` defines the convergence criterion for the EM algorithm. The algorithm stops if  $\Delta LL/LL \leq \text{tol}$ . After the mixture model has been fitted, the posterior probability that the  $i$ -th droplet is positive for the hashtag  $P(C_i = \text{pos})$  is calculated. Depending on the given data, these probabilities can be inaccurate at the far tails of the mixture distribution. Specifically, a positive component with large variance can have a larger value close to zero than the negative component, if the negative component is narrow and shifted to the right due to background HTO reads. If `correctTails` is TRUE, the following two rules are applied to avoid false classifications at the far tails. First, if the  $i$ -th droplet is classified as positive based on the posterior probability, but the probability to detected more than the observed  $y_i$  HTO counts in a negative droplet is  $P(Y \geq y_i | \text{neg}) > \text{alpha}$ , then  $P(C_i = \text{pos})$  is set to 0 (left tail). Second, if the  $i$ -th droplet is classified as negative, but  $P(Y \leq y_i | \text{pos}) > \text{beta}$ ,  $P(C_i = \text{pos})$  is set to 1 (right tail). For most datasets, these rules will not apply and it is recommended not to change these values. If `correctTails` is FALSE, posterior probabilities will not be altered, but potential problems at the tails will still

be logged in the slot `tailException` of the returned object.

3. Classification (`pAcpt`). The posterior probabilities obtained from the models fitted to each HTO separately are used to calculate the most likely class for each cell. The following classes are considered: one class for each HTO (singlets), one class for each possible multiplet, and a negative class representing droplets negative for all HTOs (i.e. empty droplets or droplets containing only cell debris). Each droplet is assigned to the most likely class unless the probability is smaller than `pAcpt`, in which case the droplet is assigned to the class "uncertain". Classification results can be accessed by running `dmmClassify` on an object returned by `demuxmix`. The acceptance probability can be changed after running `demuxmix` using `pAcpt<-`.

### Value

`demuxmix` returns an object of class `Demuxmix`. Classification results can be extracted with `dmmClassify`. Various plot methods (see below) are available to assess the model fit.

### See Also

`dmmClassify` to extract the classification results and `summary` to summarize the results. `plotDmmHistogram`, `plotDmmScatter`, `plotDmmPosteriorP`, and `dmmOverlap` to assess the model fit.

### Examples

```
set.seed(2642)
simdata <- dmmSimulateHto(class = rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                       c(rep(FALSE, 200), rep(TRUE, 220))))

dmm <- demuxmix(simdata$hto, model = "naive")
dmm
table(dmmClassify(dmm)$HTO, simdata$groundTruth)

dmmreg <- demuxmix(simdata$hto, rna = simdata$rna)
dmmreg
table(dmmClassify(dmmreg)$HTO, simdata$groundTruth)
summary(dmmreg)

pAcpt(dmmreg) <- 0.5
summary(dmmreg)

dmmOverlap(dmmreg)

plotDmmHistogram(dmmreg)
plotDmmScatter(dmmreg, hto="HTO_1")
```

## Description

Objects of this class store mixture models fitted to HTO data to demultiplex oligonucleotide-labeled cells. One mixture model is stored for each hashtag in the dataset. An object of this class is returned by `demuxmix`. Users should not directly initialize this class. There are various methods to extract or plot data from a `Demuxmix` object. Please see the package's vignette for how to work with an object of this class.

## Usage

```
## S4 method for signature 'Demuxmix'
show(object)

## S4 method for signature 'Demuxmix'
pAcpt(object)

## S4 replacement method for signature 'Demuxmix,numeric'
pAcpt(object) <- value

## S4 method for signature 'Demuxmix'
summary(object, ...)
```

## Arguments

<code>object</code>	A <code>Demuxmix</code> object.
<code>value</code>	Value between 0 and 1 specifying the acceptance probability, i.e., the minimum posterior probability required to assign a droplet to a hashtag.
<code>...</code>	Additional arguments forwarded to <code>summary</code> (ignored).

## Details

All matrices stored by `Demuxmix` have the same dimension and the same row and column names as the original matrix `hto` passed to `demuxmix`. The mixture models in slot `models` are stored in an internal class format.

## Value

An object of class `Demuxmix`.

## Functions

- `show(Demuxmix)`: Displays the object on the command line.
- `pAcpt(Demuxmix)`: Returns the acceptance probability `pAcpt`.
- `pAcpt(object = Demuxmix) <- value`: Sets a new acceptance probability `pAcpt`.
- `summary(Demuxmix)`: Summarizes the classification results and estimates error rates.

**Slots**

- models** A list of mixture models. One model per HTO.
- outliers** A logical matrix of size HTOs x droplets identifying outlier values excluded from model fitting.
- clusterInit** A numeric matrix of size HTOs x droplets with the class memberships used to initialize model fitting. A value of 1 corresponds to the negative component and a value of 2 to the positive component.
- posteriorProb** A numeric matrix of size HTO x droplets with the posterior probabilities that a droplet is positive for an HTO.
- tailException** A logical matrix of size HTO x droplets identifying posterior probabilities that would be adjusted based on the exception rules defined when calling `demuxmix` to correct inaccuracies at the extreme tails of the mixture distributions. See `demuxmix` for details.
- modelSelection** A data.frame with information about the model selection process if parameter `model` was set to 'auto'. Empty data.frame if model was specified manually.
- parameters** A list with the `demuxmix` parameters used to generate the model represented by this class.

**See Also**

`dmmClassify` to obtain classification results. `plotDmmHistogram`, `plotDmmScatter`, `plotDmmPosteriorP`, and `dmmOverlap` to assess the model fit.

**Examples**

```
set.seed(2642)
simdata <- dmmSimulateHto(class=rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                     c(rep(FALSE, 200), rep(TRUE, 220))))

dmm <- demuxmix(simdata$hto, rna=simdata$rna, pAcpt=0.9)
pAcpt(dmm)
dmm
head(dmmClassify(dmm))
```

---

`dmmClassify`

*Return classification results from a Demuxmix object*

---

**Description**

This method uses the posterior probabilities from the given `demuxmix` model to assign each droplet to the most likely class, either a single HTO, a combination of HTOs (multiplet) or the negative class (non-labeled cells, empty droplets, cell debris). If the assignment cannot be made with certainty above a defined threshold, the droplet is labeled as "uncertain".

**Usage**

```
dmmClassify(object)
```

**Arguments**

object            An object of class [Demuxmix](#).

**Details**

A droplet is labeled as "uncertain" if the posterior probability of the most likely class is smaller than the threshold `pAcpt`, which is stored in the given [Demuxmix](#) object. The acceptance probability `pAcpt` can be inspected and set to a different value by applying the getter/setter method `pAcpt` to the [Demuxmix](#) object before calling this method. The method `summary` is useful to inspect classification results and to estimate error rates for different values of `pAcpt`.

**Value**

A data.frame with 3 columns and one row for each droplet in the dataset. The first column gives the class (HTO) the droplet has been assigned to. The second column contains the posterior probability. And the third column specifies the type of the assigned class, i.e., "singlet", "multiplet", "negative" or "uncertain".

**See Also**

[demuxmix](#)

**Examples**

```
set.seed(2642)
simdata <- dmmSimulateHto(class = rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                         c(rep(FALSE, 200), rep(TRUE, 220))))

dmm <- demuxmix(simdata$hto, rna = simdata$rna)
head(dmmClassify(dmm))
table(dmmClassify(dmm)$HTO, simdata$groundTruth)

pAcpt(dmm) <- 0.5
sum(dmmClassify(dmm)$HTO == "uncertain")
pAcpt(dmm) <- 0.9999
sum(dmmClassify(dmm)$HTO == "uncertain")
```

---

dmmOverlap

*Calculate the intersection of two components of a mixture model*


---

**Description**

`dmmOverlap` sums over the probability mass intersected by the two components of the given mixture model. The sum should be close to 0 if the HTO labeling experiment was successful.

**Usage**

```
dmmOverlap(object, hto, tol = 0.001)
```

## Arguments

object	An object of class <a href="#">Demuxmix</a> .
hto	Optional vector specifying a subset of HTOs in object which should be used by this function.
tol	The maximum acceptable error when calculating the area.

## Details

The probability mass shared between the negative and positive component is an informative quality metric for the labeling efficiency of the HTO. Values under 0.03 can be considered as good, values larger than 0.1 are problematic.

The probability mass functions of the negative and positive component are not scaled by the estimated proportions of negative and positive droplets. Therefore, the result does not depend on the proportion of cells stained with the HTO and the returned value lies between 0 and 1.

The definition of the shared probability mass is not obvious for a regression mixture model since the distributions' means depend on the covariate, i.e., the number of detected genes in the RNA library. If a regression mixture model is given, this method calculates for each of the two components the weighted mean number of detected genes and uses these numbers to calculate the expectation value for the negative and positive component respectively.

## Value

A numeric vector with the shared probability mass for each HTO in the given object.

## See Also

[demuxmix](#)

## Examples

```
set.seed(2642)
simdata <- dmmSimulateHto(class = rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                       c(rep(FALSE, 200), rep(TRUE, 220))))

dmm <- demuxmix(simdata$hto, model = "naive")
dmmOverlap(dmm)

dmmreg <- demuxmix(simdata$hto, rna = simdata$rna)
dmmOverlap(dmmreg)
dmmOverlap(dmmreg, hto = "HTO_1")
dmmOverlap(dmmreg, hto = 2)
```

---

dmmSimulateHto      *Simulate HTO sequencing data*

---

### Description

This method simulates HTO count data and corresponding numbers of detected RNA features using the negative binomial distribution. The purpose of this method is to provide simple example datasets for testing and documentation.

### Usage

```
dmmSimulateHto(
  class,
  mu = 180,
  theta = 15,
  muAmbient = 30,
  thetaAmbient = 10,
  muRna = 3000,
  thetaRna = 30
)
```

### Arguments

class	A matrix of type logical defining the number of HTOs, the number of droplets, and the droplets' class memberships, i.e., which droplets contain cells that have been tagged with a certain HTO. Each row corresponds to one HTO and each column to a droplet. Negative droplets (all entries in the column are FALSE) and multiplets (more than one entry are TRUE) are allowed. If the matrix has row names, the names must be unique and are used as HTO names.
mu	Vector of expectation values of the HTO counts if a droplet is positive for the HTO. Values are recycled if mu is shorter than number of HTOs defined by class.
theta	Vector of dispersion parameters of the HTO counts if a droplet is positive for the HTO. Values are recycled if theta is shorter than number of HTOs defined by class.
muAmbient	Vector of expectation values of the HTO counts if a droplet is negative for the HTO. Values are recycled if mu is shorter than number of HTOs defined by class.
thetaAmbient	Vector of dispersion parameters of the HTO counts if a droplet is negative for the HTO. Values are recycled if theta is shorter than number of HTOs defined by class.
muRna	Single expectation value for the number of detected RNA features.
thetaRna	Single dispersion parameter for the number of detected RNA features.

**Details**

A vector  $r$  of detected RNA features (same length as columns in `class`) is simulated using `rnbinom` with `muRna` and `thetaRna` as parameters. HTO counts of positive droplets are then simulated using `rnbinom` with  $r \mu/\mu_{\text{Rna}}$  as expectation value and `theta` as dispersion. If a droplet is negative for the HTO,  $r \mu_{\text{Ambient}}/\mu_{\text{Rna}}$  and `thetaAmbient` are used respectively.

**Value**

A list with three elements: `hto` is a matrix of the same dimension as the given `class` matrix and contains the simulated HTO counts. `rna` is a vector of simulated detected number of genes (same length as `hto` has columns). `groundTruth` is a character vector encoding the class labels given by `class` as character strings for convenience.

**See Also**

[demuxmix](#)

**Examples**

```
set.seed(2642)
class <- rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
              c(rep(FALSE, 200), rep(TRUE, 220)))
simdata <- dmmSimulateHto(class = class, mu = c(150, 300), theta = c(15, 20),
                        muAmbient = c(30, 30), thetaAmbient = c(10, 10),
                        muRna = 3000, thetaRna = 30)

dim(simdata$hto)
table(simdata$groundTruth)

mean(simdata$rna) # muRna
var(simdata$rna) # muRna + muRna^2/thetaRna

mean(simdata$hto[1, class[1, ]]) # mu[1]
mean(simdata$hto[1, !class[1, ]]) # muAmbient[1]
var(simdata$hto[1, class[1, ]]) # > mu[1] + mu[1]^2/theta[1]

cor(simdata$rna[class[1, ]], simdata$hto[1, class[1, ]])
```

---

plotDmmHistogram

*Plotting a histogram with mixture probability mass function*

---

**Description**

This methods plots the mixture probability mass function with the negative and positive component on top of a histogram of the HTO counts used to fit the mixture model. The mixture model must be generated by [demuxmix](#).

**Usage**

```
plotDmmHistogram(object, hto, quantile = 0.95, binwidth = 5)
```

**Arguments**

object	An object of class <a href="#">Demuxmix</a> .
hto	Optional vector specifying a subset of HTOs in object which should be used by this function.
quantile	Quantile of the mixture distribution which is used as right limit of the plot's x axis.
binwidth	Width of the bins of the histogram.

**Details**

A histogram overlaid with the pmf is a standard tool to assess the fit of a the mixture model and trivial for a naive mixture model. However, if a regression mixture model is given, the expectation values of the components are different for each droplet depending on the covariates (here the number of genes detected in the droplet). This method calculates the weighted mean number of detected genes in droplets in the positive and negative component, and then uses these numbers to calculate expectation values for an average droplet of the positive and negative component. The HTO counts shown in the histogram are adjusted to account for different numbers of detected genes by replacing the original HTO counts with the expected counts given the mean number of detected genes plus the residuals from the regression model. In other words, the effect of the number of detected genes was regressed out before plotting the HTO counts in the histogram.

It may be useful to zoom into the plot to obtain a better view of the fit. To restrict the plot to a certain range on the x or y axis, the method [coord\\_cartesian](#) from the `ggplot2` package should be used (see examples).

**Value**

An object of class `ggplot` is returned, if only one HTO is plotted. If several HTOs are plotted simultaneously, a grid of plots is returned.

**See Also**

[demuxmix](#)

**Examples**

```
set.seed(2642)
simdata <- dmmSimulateHto(class = rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                         c(rep(FALSE, 200), rep(TRUE, 220))))

dmm <- demuxmix(simdata$hto, simdata$rna)
plotDmmHistogram(dmm)
p <- plotDmmHistogram(dmm, hto = 1)
p + ggplot2::coord_cartesian(xlim = c(25, 100), ylim = c(0, 0.01))
```

---

plotDmmPosteriorP      *Plotting a histogram of posterior probabilities*

---

### Description

This methods plots a histogram of posterior probabilities obtained from the given mixture model. The posterior probabilities indicate whether the droplet likely contains a cell labeled by the respective HTO. The mixture model passed to this function must be generated by [demuxmix](#).

### Usage

```
plotDmmPosteriorP(object, hto, bins = 50)
```

### Arguments

object	An object of class <a href="#">Demuxmix</a> .
hto	Optional vector specifying a subset of HTOs in object which should be used by this function.
bins	The number of bins of the histogram.

### Details

The histogram visualizes how well the positive droplets can be separated from the negative droplets. Ideally, the histogram shows many droplets with a posterior probability very close to 0 and many droplets close to 1, but no or very few droplets with probabilities somewhere in between. The histogram can be useful for guiding the selection of the acceptance probability  $p_{Acpt}$ .

### Value

An object of class `ggplot` is returned, if only one HTO is plotted. If several HTOs are plotted simultaneously, a grid of plots is returned.

### See Also

[demuxmix](#)

### Examples

```
set.seed(2642)
simdata <- dmmSimulateHto(class = rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                       c(rep(FALSE, 200), rep(TRUE, 220))))

dmm <- demuxmix(simdata$hto, model = "naive")
plotDmmPosteriorP(dmm)

dmmreg <- demuxmix(simdata$hto, rna = simdata$rna, model = "auto")
plotDmmPosteriorP(dmmreg)
plotDmmPosteriorP(dmmreg, hto = 1)
```

---

`plotDmmScatter`*Plotting RNA features versus HTO counts*

---

## Description

This methods plots the number of genes detected in a droplet versus the number of sequenced HTOs. The posterior probability that the droplet is positive for the HTO is indicated by a color gradient. Optionally, the decision boundary with posterior probability 0.5 can be plotted. The mixture model passed to this function must be a regression mixture model generated by [demuxmix](#).

## Usage

```
plotDmmScatter(  
  object,  
  hto,  
  log = TRUE,  
  pointsize = 1.2,  
  plotDecBoundary = TRUE,  
  tol = 0.01  
)
```

## Arguments

<code>object</code>	An object of class <a href="#">Demuxmix</a> .
<code>hto</code>	Optional vector specifying a subset of HTOs in <code>object</code> which should be used by this function.
<code>log</code>	Logical value indicating whether both HTO counts and number of detected genes should be log transformed.
<code>pointsize</code>	Numeric value specifying the size of the points.
<code>plotDecBoundary</code>	Logical value indicating whether the decision boundary should be added to the plot.
<code>tol</code>	Numeric value between 0 and 1 specifying the error tolerance of the decision boundary, i.e., a point on the plotted line has a posterior probability within 0.5 +/- <code>tol</code> . Only used if <code>plotDecBoundary</code> is true.

## Details

The scatterplot produced by this method is helpful to assess the relation between the number of detected genes and the number of HTO counts obtained for a droplet. A positive association is usually visible for the positive cells (i.e., droplets with cells treated with the oligo-labeled antibodies). The association is often weak/absent in the droplets negative for the HTO. This method can only be applied to regression mixture models and not to naive mixture models. To see whether a [Demuxmix](#) object contains regression mixture models, type `show(object)` to display the type of model used for each HTO.

**Value**

An object of class `ggplot` is returned, if only one HTO is plotted. If several HTOs are plotted simultaneously, a grid of plots is returned.

**See Also**

[demuxmix](#)

**Examples**

```
set.seed(2642)
simdata <- dmmSimulateHto(class = rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                       c(rep(FALSE, 200), rep(TRUE, 220))))

dmmreg <- demuxmix(simdata$hto, rna = simdata$rna, model = "reg")
plotDmmScatter(dmmreg)
plotDmmScatter(dmmreg, hto = 1, log = FALSE)
```

---

summary

*Summarize classification results of a Demuxmix model*

---

**Description**

This method takes the demultiplexing results from an HTO experiment returned by [demuxmix](#) and returns a `data.frame` summarizing the classification results and expected error rates.

**Usage**

```
summary(object, ...)
```

**Arguments**

<code>object</code>	An object of class <a href="#">Demuxmix</a> .
<code>...</code>	Additional parameters (ignored).

**Details**

Results are summarized for the individual HTOs, for all singlets combined, for all multiplets combined, and for the negative class. Relative frequencies are calculated after excluding the "uncertain" class. The estimated number of false positive droplets and the estimated FDR are based on several assumptions, one of which is the independence of the HTO counts from different hashtags. This assumption is unlikely for real data where all HTO counts are obtained from the same droplet. Usually, the positive correlation among HTOs causes an overestimation of multiplets and negative/empty droplets. Error rates are more accurate when regression mixture models are used since the number of detected genes explains some of the positive correlation between HTOs.

**Value**

A data.frame with one row per class showing the number of droplets in the class (NumObs), the relative frequency of the class (RelFreq), the median probability with which a droplet was assigned to the class (MedProb), the estimated number of droplets falsely assigned to the class (ExpFPs), and the corresponding estimated false discovery rate (FDR).

**See Also**

[demuxmix](#)

**Examples**

```
set.seed(2642)
simdata <- dmmSimulateHto(class = rbind(c(rep(TRUE, 220), rep(FALSE, 200)),
                                       c(rep(FALSE, 200), rep(TRUE, 220))))

dmm <- demuxmix(simdata$hto, rna = simdata$rna)
summary(dmm)
pAcpt(dmm) <- 0.05
summary(dmm)
```

# Index

- \* **datasets**
  - csf, [2](#)
  
- coord\_cartesian, [13](#)
- csf, [2](#)
  
- Demuxmix, [6, 9, 10, 13–16](#)
- Demuxmix (Demuxmix-class), [6](#)
- demuxmix, [3, 7–10, 12–17](#)
- demuxmix, Matrix, missing-method (demuxmix), [3](#)
- demuxmix, matrix, missing-method (demuxmix), [3](#)
- demuxmix, Matrix, numeric-method (demuxmix), [3](#)
- demuxmix, matrix, numeric-method (demuxmix), [3](#)
- Demuxmix-class, [6](#)
- dmmClassify, [6, 8, 8](#)
- dmmClassify, Demuxmix-method (dmmClassify), [8](#)
- dmmOverlap, [6, 8, 9](#)
- dmmOverlap, Demuxmix, ANY-method (dmmOverlap), [9](#)
- dmmOverlap, Demuxmix, missing-method (dmmOverlap), [9](#)
- dmmSimulateHto, [11](#)
- dmmSimulateHto, matrix-method (dmmSimulateHto), [11](#)
  
- pAcpt, [9](#)
- pAcpt (Demuxmix-class), [6](#)
- pAcpt, Demuxmix-method (Demuxmix-class), [6](#)
- pAcpt<- (Demuxmix-class), [6](#)
- pAcpt<- , Demuxmix, numeric-method (Demuxmix-class), [6](#)
- plotDmmHistogram, [6, 8, 12](#)
- plotDmmHistogram, Demuxmix, ANY-method (plotDmmHistogram), [12](#)
- plotDmmHistogram, Demuxmix, missing-method (plotDmmHistogram), [12](#)
- plotDmmPosteriorP, [6, 8, 14](#)
- plotDmmPosteriorP, Demuxmix, ANY-method (plotDmmPosteriorP), [14](#)
- plotDmmPosteriorP, Demuxmix, missing-method (plotDmmPosteriorP), [14](#)
- plotDmmScatter, [6, 8, 15](#)
- plotDmmScatter, Demuxmix, ANY-method (plotDmmScatter), [15](#)
- plotDmmScatter, Demuxmix, missing-method (plotDmmScatter), [15](#)
  
- rnbinom, [12](#)
  
- show, Demuxmix-method (Demuxmix-class), [6](#)
- summary, [6, 9, 16](#)
- summary, data.frame-method (summary), [16](#)
- summary, Demuxmix-method (Demuxmix-class), [6](#)