

# Package ‘cytometree’

July 3, 2017

**Type** Package

**Title** Automated Cytometry Gating

**Version** 1.0.1

**Date** 2017-06-25

**Author** Alkhassim Chariff

**Maintainer** Alkhassim Chariff <chariff.alkhassim@gmail.com>

**LinkingTo** Rcpp, RcppArmadillo

**Description** Given the hypothesis of a bimodal distribution of cells for each marker, the algorithm constructs a binary tree, the nodes of which are subpopulations of cells. At each node, observed cells and markers are modeled by both a family of normal distributions and a family of bimodal normal mixture distributions. Splitting is done according to a normalized difference of AIC between the two families.

**License** LGPL-3 | file LICENSE

**LazyData** true

**Depends** R ( $\geq 3.1.0$ ), Rcpp ( $\geq 0.12.4$ )

**Imports** ggplot2, graphics, igraph, mclust, stats

**RoxygenNote** 6.0.1

**BugReports** <https://github.com/chariff/Cytometree/issues>

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2017-07-03 13:30:40 UTC

## R topics documented:

cytometree-package . . . . .	2
Annotation . . . . .	2
bootstrapCI . . . . .	3
CytomeTree . . . . .	4
DLBCL . . . . .	6

FmeasureC . . . . .	6
FmeasureC_no0 . . . . .	7
plot_graph . . . . .	7
plot_nodes . . . . .	8
RetrievePops . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

cytometree-package	<i>Binary tree algorithm for cytometry data analysis.</i>
--------------------	---

---

**Description**

The algorithm is based on the construction of a binary tree, the nodes of which are subpopulations of cells. At each node, observed cells and markers are modeled by both a family of normal distributions and a family of bimodal normal mixture distributions. Splitting is done according to a normalized difference of AIC between the two families. Given the unsupervised nature of the binary tree, some of the available markers may not be used to find the different cell populations present in a given sample. To recover a complete annotation, we defined, as a post processing procedure, an annotation method which allows the user to distinguish two or three expression levels per marker.

**Details**

Package: cytometree  
Type: Package  
Version: 1.0.0  
Date: 2017-05-23  
License: **LGPL-3**

The main function in this package is [CytomeTree](#).

**Author(s)**

Chariff Alkhassim — Maintainer: Chariff Alkhassim

**References**

D. Commenges, C. Alkhassim, B.P. Hejblum, R. Thiebaut. A Binary tree algorithm for automatic gating in cytometry analysis. Submitted, 2017.

---

Annotation	<i>Annotates cell populations found using CytomeTree.</i>
------------	---

---

**Description**

Annotates cell populations found using CytomeTree.

**Usage**

```
Annotation(CytomeTreeObj, K3markers = NULL, plot = TRUE)
```

**Arguments**

CytomeTreeObj	An object of class CytomeTree.
K3markers	A vector of class character where the names of the markers for which 3 levels of expression are sought can be specified. Default is NULL i.e. 2 levels of expression per marker.
plot	A logical value indicating whether or not to plot the partitioning in 2 or 3 groups for each marker. Default is TRUE.

**Details**

The algorithm is set to find the partitioning in 2 or 3 groups of cell populations found using CytomeTree. It minimizes the within-leaves sum of squares of the observed values on each marker.

**Value**

A data.frame containing the annotation of each cell population.

**Author(s)**

Chariff Alkhassim

---

bootstrapCI	<i>Bootstrapped Confidence Interval.</i>
-------------	--

---

**Description**

Bootstrapped Confidence Interval.

**Usage**

```
bootstrapCI(stat, n, alpha)
```

**Arguments**

stat	A numeric vector of statistics for which to compute a confidence interval.
n	An integer giving the number of bootstrap samples.
alpha	A real number comprised in ]0, 1[ : 1 - desired confidence level.

**Author(s)**

Chariff Alkhassim

---

CytomeTree

*Binary tree algorithm for cytometry data analysis.*


---

## Description

Binary tree algorithm for cytometry data analysis.

## Usage

```
CytomeTree(M, minleaf = 1, t = 0.1)
```

## Arguments

M	A matrix of size $n \times p$ containing cytometry measures of $n$ cells on $p$ markers.
minleaf	An integer indicating the minimum number of cells per population. Default is 1.
t	A real positive-or-null number used for comparison with the normalized AIC computed at each node of the tree. A higher values limits the height of the tree.

## Details

The algorithm is based on the construction of a binary tree, the nodes of which are subpopulations of cells. At each node, observed cells and markers are modeled by both a family of normal distributions and a family of bimodal normal mixture distributions. Splitting is done according to a normalized difference of AIC between the two families.

## Value

An object of class 'cytomeTree' providing a partitioning of the set of  $n$  cells.

- `annotation` A data.frame containing the annotation of each cell population underlying the tree pattern.
- `labels` The partitioning of the set of  $n$  cells.
- `M` The input matrix.
- `mark_tree` A two level list containing markers used for node splitting.

## Author(s)

Chariff Alkhassim

## Examples

```
head(DLBCL)
# number of cell event.
N <- nrow(DLBCL)
# Cell events.
cellevents <- DLBCL[,c("FL1", "FL2", "FL4")]
# Manual partitioning of the set N (from FlowCAP-I).
```

```

manual_labels <- DLBCL[, "label"]
# Build the binary tree.
Tree <- CytomeTree(cellevents, minleaf = 1, t=.1)
# Retrieve the resulting partition of the set N.
Tree_Partition <- Tree$labels
# Plot node distributions.
par(mfrow=c(1,2))
plot_nodes(Tree)
# Choose a node to plot.
plot_nodes(Tree, "FL4.1")
# Plot a graph of the tree.
par(mfrow=c(1,1))
plot_graph(Tree, edge.arrow.size=.3, Vcex=.5, vertex.size=30)
# Run the annotation algorithm.
Annot <- Annotation(Tree, plot=FALSE)
Annot$combinations
# Compare to the annotation gotten from the tree.
Tree$annotation
# Example of seeked phenotypes.
# Variable in which seeked phenotypes can be entered in the form
# of matrices.
phenotypes <- list()
## Seeked phenotypes.
# FL1- FL2+ FL4-.
phenotypes[[1]] <- rbind(c("FL1", 0), c("FL2", 1), c("FL4", 0))
# FL1+ FL2- FL4+.
phenotypes[[2]] <- rbind(c("FL1", 1), c("FL2", 0), c("FL4", 1))
# FL1+ FL2+ FL4+.
phenotypes[[3]] <- rbind(c("FL1", 1), c("FL2", 1), c("FL4", 1))
# Retrieve cell populations found using Annotation.
PhenoInfos <- RetrievePops(Annot, phenotypes)
PhenoInfos$phenotypesinfo
# F-measure ignoring cells labeled 0 as in FlowCAP-I.
# Use FmeasureC() in any other case.
FmeasureC_no0(ref=manual_labels, pred=Tree_Partition)
# Scatterplots.
library(ggplot2)
# Ignoring cells labeled 0 as in FlowCAP-I.
rm_zeros <- which(!manual_labels)
# Building the data frame to scatter plot the data.
FL1 <- cellevents[-c(rm_zeros), "FL1"]
FL2 <- cellevents[-c(rm_zeros), "FL2"]
FL4 <- cellevents[-c(rm_zeros), "FL4"]
n <- length(FL1)
Labels <- c(manual_labels[-c(rm_zeros)]%2+1, Tree_Partition[-c(rm_zeros)])
Labels <- as.factor(Labels)
method <- as.factor(c(rep("FlowCap-I", n), rep("CytomeTree", n)))
scatter_df <- data.frame("FL2"=FL2, "FL4"=FL4, "labels"=Labels, "method"=method)
p <- ggplot2::ggplot(scatter_df, ggplot2::aes_string(x="FL2", y="FL4", colour="labels"))+
  ggplot2::geom_point(alpha=1, cex=1)+
  ggplot2::scale_colour_manual(values=c("green", "red", "blue"))+
  ggplot2::facet_wrap(~ method)+
  ggplot2::theme_bw()+

```

```
ggplot2::theme(legend.position="bottom")
p
```

DLBCL

*Diffuse large B-cell lymphoma data set from the flowCAP-I challenge.***Description**

Diffuse large B-cell lymphoma data set from the flowCAP-I challenge.

**Usage**

```
data(DLBCL)
```

**Format**

A data frame with 5524 cell events and 3 markers.

**Source**

<http://flowcap.flowsite.org/>

FmeasureC

*C++ implementation of the F-measure computation***Description**

C++ implementation of the F-measure computation

**Usage**

```
FmeasureC(pred, ref)
```

**Arguments**

pred	vector of a predicted partition
ref	vector of a reference partition

---

FmeasureC_no0	<i>C++ implementation of the F-measure computation without the ref classe 0</i>
---------------	---

---

**Description**

Aghaeepour in FlowCAP 1 ignore the reference class labeled "0"

**Usage**

```
FmeasureC_no0(pred, ref)
```

**Arguments**

pred	vector of a predicted partition
ref	vector of a reference partition

---

plot_graph	<i>Plot the binary tree built using CytomeTree.</i>
------------	---

---

**Description**

Plot the binary tree built using CytomeTree.

**Usage**

```
plot_graph(CytomeTreeObj, Ecex = 1, Ecolor = 8, Vcex = 0.8, Vcolor = 0,
...)
```

**Arguments**

CytomeTreeObj	An object of class CytomeTree.
Ecex	Number indicating the amount by which text on the edges should be scaled. Default is 1.
Ecolor	An integer or a string of character to color edges of the graph. Default is 8.
Vcex	Number indicating the amount by which text in the vertices should be scaled. Default is .8.
Vcolor	A vector of class numeric or character to color vertices of the graph. Default is 0.
...	additional arguments to be passed to <a href="#">plot_graph</a>

**Author(s)**

Chariff Alkhassim

---

plot_nodes	<i>Plot the distribution of the observed cells at each node of the binary tree built using CytomeTree.</i>
------------	--

---

### Description

Plot the distribution of the observed cells at each node of the binary tree built using CytomeTree.

### Usage

```
plot_nodes(CytomeTreeObj, nodes = NULL)
```

### Arguments

CytomeTreeObj	An object of class CytomeTree.
nodes	A vector of class character containing the name of the nodes for which the distribution is to be plotted. Default is NULL, and plots the distribution of each node.

### Author(s)

Chariff Alkhassim

---

RetrievePops	<i>Retrieve cell populations found using Annotation.</i>
--------------	--

---

### Description

Retrieve cell populations found using Annotation.

### Usage

```
RetrievePops(AnnotationObj, phenotypes)
```

### Arguments

AnnotationObj	An object of class Annotation.
phenotypes	A list containing at least one element of class matrix describing a sought phenotype. Each matrix should have two columns where the name of a used marker is associated to a value chosen between 0, 1 and 2. 0 translates to -, 1 to + and 2 to ++.



**Value**

A list of two elements.

- `phenotypesinfo` A list containing informations about seeked populations.
- `Mergedleaves` The partitioning of the set of  $n$  cells with potentially merged leaves.

**Author(s)**

Chariff Alkhassim

# Index

\*Topic **data**

DLBCL, [6](#)

Annotation, [2](#)

bootstrapCI, [3](#)

CytomeTree, [2](#), [4](#)

cytometree (cytometree-package), [2](#)

cytometree-package, [2](#)

DLBCL, [6](#)

FmeasureC, [6](#)

FmeasureC\_no0, [7](#)

plot\_graph, [7](#), [7](#)

plot\_nodes, [8](#)

RetrievePops, [8](#)