Package ‘flowType’

March 22, 2017

Type Package
Title Phenotyping Flow Cytometry Assays
Version 2.12.0
Date 2014-11-26
Author Nima Aghaeepour, Kieran O’Neill, Adrin Jalali
Maintainer Nima Aghaeepour <naghaeep@gmail.com>
Description Phenotyping Flow Cytometry Assays using multidimensional expansion of single dimensional partitions.
Imports Biobase, graphics, grDevices, methods, flowCore, flowMeans, sfsmisc, rrcov, flowClust, flowMerge, stats
Depends R (>= 2.10), Rcpp (>= 0.10.4), BH (>= 1.51.0-3)
LinkingTo Rcpp, BH
Suggests xtable
biocViews FlowCytometry
License Artistic-2.0
LazyLoad yes
NeedsCompilation yes

R topics documented:

flowType-package ........................................... 2
calcMemUse ................................................. 3
calcNumPops ............................................... 4
decodePhenotype ........................................... 5
DLBCLExample ............................................. 6
encodePhenotype ........................................... 6
flowType .................................................... 7
getLabels .................................................... 9
HIVData .................................................... 10
HIVMetaData ............................................... 10
Phenotypes-class ......................................... 11
plot ......................................................... 12
summary-methods .......................................... 12

Index 14
flowType-package

flowType: Phenotyping Flow Cytometry Assays

Description

flowType uses a simple threshold, Kmeans, flowMeans, or flowClust to partition every channel to a positive and a negative cell population. These partitions are then combined to generate a set of multi-dimensional phenotypes.

Details

Package: flowType
Type: Package
Version: 0.0.1
Date: 2011-04-27
License: Artistic-2.0
LazyLoad: yes
Depends: methods

For a given FCS file, the flowType function extracts a the phenotypes and reports their cell frequencies (number of cells) and mean fluorescence intensity (MFI)s.

Author(s)

Nima Aghaeepour, Kieran O’Neill, Adrin Jalali

References

Please cite the following for the current version of flowType:


The original paper and description can be found at:


Examples

#Load the library
library(flowType)
data(DLBCLExample)

#These markers will be analyzed
PropMarkers <- 3:5
MFIMarkers <- PropMarkers
MarkerNames <- c("FS", "SS", "CD3", "CD5", "CD19")
# Run flowType
Res <- flowType(DLBCLExample, PropMarkers, MFIMarkers, 'kmeans', MarkerNames);

MFIs=Res@MFIs;
Proportions=Res@CellFreqs;
Proportions <- Proportions / max(Proportions)
names(Proportions) <- unlist(lapply(Res@PhenoCodes,
  function(x){return(decodePhenotype(x, Res@MarkerNames[PropMarkers],
   Res@PartitionsPerMarker))}))

# Select the 30 largest phenotypes
index=order(Proportions,decreasing=TRUE)[1:30]
bp=barplot(Proportions[index], axes=FALSE, names.arg=FALSE)
text(bp+0.2, par("usr")[3]+0.02, srt = 90, adj = 0, labels = names(Proportions[index]), xpd = TRUE, cex=0.8)
axis(2);
axis(1, at=bp, labels=FALSE);
title(xlab='Phenotype Names', ylab='Cell Proportion')

# These phenotype can be analyzed using a predictive model (e.g., classification or regression)

---

**calcMemUse**

*Function calcMemUse in Package flowType*

**Description**

Estimates the memory usage in bytes for running flowType with a given set of parameters.

**Usage**

`calcMemUse(NumPops, NumPropMarkers, NumMFIMarkers, NumCells, MaxMarkersPerPop, PartitionsPerChannel)`

**Arguments**

- `NumPops`: Number of cell types which will be returned. Can be computed using `calcNumPops`
- `NumPropMarkers`: Number of markers to use for combinatorial gating
- `NumMFIMarkers`: Number of markers to determine the MFIs of for every cell type
- `NumCells`: Number of cells in the flowFrame passed to flowType
- `MaxMarkersPerPop`: Maximum number of markers to use at once in combinatorial gating (ie all cell types over 1:MaxMarkersPerPop will be counted)
- `PartitionsPerChannel`: Number of partitions per channel.

**Details**

If you use different numbers of partitions for different channels, try providing the highest number as `PartitionsPerChannel`, and expect an over-estimate.

**Value**

Estimated memory use in bytes.
calcNumPops

Author(s)

Kieran O’Neill

See Also

calcNumPops, flowType

Description

Compute the number of populations that will be produced by running flowType with a given set of parameters. This is especially useful for estimating memory but is also used internally to determine the size of return objects to pass down to C++. This may also be useful for determining the cutoff for number of markers to use to make phenotypes (in terms of statistical power for later testing).

Usage

calcNumPops(PartitionsPerMarker, MaxMarkersPerPop)

Arguments

PartitionsPerMarker
  Integer vector specifying the number of partitions for each marker, in order.

MaxMarkersPerPop
  Integer specifying the threshold chosen

Value

Integer specifying the number of populations the given parameters would produce.

Author(s)

Kieran O’Neill

See Also

calcMemUse, flowType

Examples

calcNumPops(c(2,2,3,2,2,4), 5)
Method to decode phenotypes back to a human-readable string.

Details

FlowType’s encoding is as follows:

0 – marker not considered in phenotype (don’t care about its value) 1 – marker is negative (e.g. CD4-) 2 – marker is positive (e.g. CD4+) 3 – marker is positive, but brighter than 2 (CD4++) 4 – marker is even brighter (CD4+++ etc

Note that this encoding system does not allow for "dim" markers – dim positivity is denoted by the first level of positivity.

Also note that the encoding is performed from the dimmest to the brightest partition, but the location of thresholds will dictate the interpretation of the code. (e.g. if you only set one threshold, but you place it between the positive and the bright population, then both positive and negative events will be considered negative.)

Methods

signature(pheno.code = "character", marker.names = "character", partitions.per.marker = "numeric")

In flowType, phenotypes themselves are represented by codes (e.g. 012) rather than full strings (CD4+CD8-), in order to save memory when a very large number of phenotypes are considered. decodePhenotype serves to translate the codes back to a human-readable string.

Author(s)

Kieran O’Neill

See Also

encodePhenotype, flowType

Examples

decodePhenotype(’1034’,c(’CD34’,’CD3’), 4)
decodePhenotype(’20013’,c(’CD34’,’CD3’,’CD45’,’CD19’), 4)
**Description**

A flow cytometry sample from a patient with DLBC lymphoma. The full dataset is available through the FlowCAP project (http://flowcap.flowsite.org).

**Usage**

data(DLBCLExample)

**Format**

A flowFrame describing expression values of 3 markers and 3796 cells. Each column represents a marker and each row represents a cell.

**Examples**

data(DLBCLExample)

---

**encodePhenotype**

*Function encodePhenotype in Package flowType*

**Description**

In flowType, phenotypes themselves are represented by codes (e.g. 012) rather than full strings (CD4+CD8-), in order to save memory when a very large number of phenotypes are considered. encodePhenotype serves to translate a human-readable string down to flowType's internal coded representation.

**Usage**

encodePhenotype(pheno.string, marker.names)

**Arguments**

pheno.string character vector containing containing the phenotype string to be encoded
marker.names vector of character vectors each specifying the name of a channel, in order

**Details**

FlowType's encoding is as follows:

- 0 – marker not considered in phenotype (don’t care about its value)
- 1 – marker is negative (e.g. CD4-)
- 2 – marker is positive (e.g. CD4+)
- 3 – marker is positive, but brighter than 2 (CD4++)
- 4 – marker is even brighter (CD4+++)

Note that this encoding system does not allow for "dim" markers – dim positivity is denoted by the first level of positivity.
flowType

Also note that the encoding is performed from the dimmest to the brightest partition, but the location of thresholds will dictate the interpretation of the code. (e.g. if you only set one threshold, but you place it between the positive and the bright population, then both positive and negative events will be considered negative.)

Value

Character vector containing the encoded phenotype, with one character per channel.

Author(s)

Kieran O’Neill

See Also

decodePhenotype, flowType

Examples

encodePhenotype(CD34++CD3-CD45+++),c('CD34','CD3','CD45','CD19'))

flowType

flowType: Phenotyping Flow Cytometry Assays

Description

flowType uses a simple threshold, Kmeans, flowMeans or flowClust to partition every channel to a positive and a negative cell population. These partitions are then combined to generate a set of multi-dimensional phenotypes.

Usage

flowType(Frame, PropMarkers=NULL, MFIMarkers=NULL, Methods='kmeans', MarkerNames=NULL, MaxMarkersPerPop=NULL, PartitionsPerMarker=2, Thresholds=NULL, MemLimit=4,verbose=FALSE)

#If upgrading from flowType 1.x to 2.x, please check documentation as some arguments have changed slightly.

Arguments

Frame A flowFrame (after transformation) that is going to be phenotyped.
PropMarkers A vector of the indexes or names of the markers to partition to specify phenotypes. If NULL, all markers in the frame will be used.
MFIMarkers A vector of the indexes or names of the markers for which MFIs must be measured. If NULL, no markers will be used.
Methods A single string specifying the method to use to determine thresholds for partitioning of markers. Values can be "kmeans", "flowMeans", "flowClust", or "Thresholds". If "Thresholds" is specified, user-specified thresholds must be provided via the Thresholds parameter.
MarkerNames A vector of names for the channels. If NULL, the names in Frame will be used.
MaxMarkersPerPop An integer specifying the maximum number of markers to use to define populations (how "deep" to phenotype). This should be less than or equal to PropMarkers. If NULL, will default to the length of PropMarkers.
PartitionsPerMarker

An integer or vector of integers specifying the number of partitions per marker. If a single integer, this number will be used for all markers. If a vector, the numbers will be matched with PropMarkers in order.

Thresholds

A list of vectors specifying per-channel thresholds. Each list item corresponds to one marker, and contains the threshold(s) for that marker. If only one vector is provided in the list, then those thresholds will be used for all markers. Otherwise, the list must be of the same length as PropMarkers. Note: if Methods == 'thresholds', then Thresholds must be specified. If not, it is ignored.

MemLimit

Memory limit in GB. flowType will do a sanity check before executing, and if the total size of counts plus MFI values for all populations would exceed MemLimit, will not run.

verbose

Boolean variable. If TRUE, information about different processing tasks will be printed into the standard output.

Value

CellFreqs: Object of class "numeric" containing the cell frequencies measured for each phenotype. Phenotype names are assigned as labels.

MFIs: Object of class "matrix" containing the measured MFIs for each phenotype. Phenotype names are assigned as column labels and marker names as row labels.

PropMarkers A vector of the indexes or names of the markers for which cell proportions must be measured.

MFIMarkers A vector of the indexes or names of the markers for which MFIs must be measured.

MarkerNames A vector of names for the channels. If NULL, the names provided in Frame will be used.

Partitions A matrix where each column shows the partitioning of the respective channel. 1 and 2 correspond to negative and positive, respectively.

PhenoCodes A vector of strings of length N (the number of markers) for each phenotype measured. For every phenotype, the character corresponding to a given marker can be 0, 1, 2, etc for neutral, negative, positive, bright, etc. See the provided vignette for more details and examples.

Author(s)

Nima Aghaeepour, Kieran O’Neill

References

Please cite the following for the current version of flowType:


The original paper and description can be found at:

**getLabels**

**Examples**

```r
#Load the library
library(flowType)
data(DLBCLExample)

#These markers will be analyzed
PropMarkers <- 3:5
MFIMarkers <- PropMarkers
MarkerNames <- c('FS', 'SS', 'CD3', 'CD5', 'CD19')

#Run flowType
Res <- flowType(DLBCLExample, PropMarkers, MFIMarkers, 'kmeans', MarkerNames);

MFIs=Res@MFIs;
Proportions=Res@CellFreqs;
Proportions <- Proportions / max(Proportions)
names(Proportions) <- unlist(lapply(Res@PhenoCodes, 
    function(x){return(decodePhenotype(x,Res@MarkerNames[PropMarkers],
        Res@PartitionsPerMarker))}))

#Select the 30 largest phenotypes
index=order(Proportions,decreasing=TRUE)[1:30]
bp=barplot(Proportions[index], axes=FALSE, names.arg=FALSE)
    text(bp+0.2, par("usr")[3]+0.02, srt = 90, adj = 0, labels = names(Proportions[index]), xpd = TRUE, cex=0.8)
axis(2);
axis(1, at=bp, labels=FALSE);
title(xlab="Phenotype Names", ylab='Cell Proportion')

#These phenotype can be analyzed using a predictive model (e.g., classification or regression)
```

**getLabels**

*getLabels: Returns the labels of the cells in a given phenotype.*

**Description**

Returns the labels of the cells in a given phenotype in a Phenotypes object.

**Usage**

```r
getLabels(Phenotypes, PhenotypeNumber)
```

**Arguments**

- **Phenotypes**: An object of class Phenotypes as produced by the `flowType` function.
- **PhenotypeNumber**: A numeric or character value representing the phenotypes number of name, respectively.

**Value**

- **Membership Labels**: A vector of length of the number of events. 1 and 2 represent the cells that are not-included and included in the phenotype respectively.
Author(s)
Nima Aghaeepour

References

Examples
#See the vignette
**Phenotypes-class**

**Examples**

```r
data(HIVMetaData)
```

---

**Phenotypes-class**  
*Class "Phenotypes"*

**Description**

The return data from running `flowType`, containing counts

**Objects from the Class**

Objects can be created by calls of the form `new("Phenotypes", ...)`.  

**Slots**

- **CellFreqs**: Numeric vector containing counts of the number of cells belonging to each cell type.  
- **MFIs**: Matrix of MFIs, with rows for cell types and columns for markers.  
- **PhenoCodes**: Vector of character strings representing the codes of each cell type (phenotype).  
- **PropMarkers**: Numeric vector specifying which markers were used for combinatorial gating.  
- **MFIMarkers**: Numeric vector specifying for which markers MFIs were computed for each cell type.  
- **MarkerNames**: A character vector of the names of all markers in the flowFrame given
- **Partitions**: The first level partitions that each cell in the flowFrame belong to in each channel.  
- **MaxPopSize**: `MaxMarkersPerPop`  
- **PartitionsPerMarker**: Vector of number of partitions used for each marker  
- **Thresholds**: A list of vectors with the calculated thresholds for each marker (if a clustering algorithm was used) or the thresholds provided by the user.

**See Also**

`flowType`

**Examples**

```r
displayClass("Phenotypes")
```
Methods for Function `plot`

**Description**
Methods for function `plot`

**Usage**
```r
## S4 method for signature 'Phenotypes,flowFrame'
plot(x, y, ...)  
## S4 method for signature 'Phenotypes,numeric'
plot(x, y, Frame, ...)
## S4 method for signature 'Phenotypes,character'
plot(x, y, Frame, ...)
```

**Arguments**
- `x`: An object of class `Phenotypes` as generated by the `flowType` package.
- `y`: A `flowFrame` or a numeric/character value representing the phenotype that needs to be plotted depending on the signature of the function.
- `Frame`: A `flowFrame` (might be optional depending on the signature of the function).
- `...`: Extra parameters that will be passed to the generic plot function.

**Author(s)**
Nima Aghaeepour <naghaeep@gmail.com>

**See Also**
- `flowType`

**Examples**
```
# See the vignette
```

---

Methods for Function `summary`

**Description**
~~ Methods for function `summary` ~~

**Methods**
- `signature(object = "Phenotypes")` Prints basic characteristics of a `Phenotypes` object.

**See Also**
- `flowType`
Examples

#See the vignette
Index

*Topic FlowCytData
  flowType, 7
  flowType-package, 2
  getLabels, 9
*Topic HIV
  flowType, 7
  flowType-package, 2
  getLabels, 9
*Topic classes
  Phenotypes-class, 11
*Topic classification
  flowType, 7
  flowType-package, 2
  getLabels, 9
*Topic clustering
  flowType, 7
  flowType-package, 2
  getLabels, 9
*Topic datasets
  DLBCLExample, 6
  HIVData, 10
  HIVMetaData, 10
*Topic print
  plot, 12
  summary-methods, 12
*Topic utilities
  calcMemUse, 3
  calcNumPops, 4
  decodePhenotype, 5
  encodePhenotype, 6

  calcMemUse, 3, 4
  calcNumPops, 3, 4, 4
  decodePhenotype, 5, 7
  decodePhenotype, character, character, numeric-method
    (decodePhenotype), 5
  decodePhenotype, character, character, numeric-method
    (decodePhenotype), 5
  decodePhenotype-methods
    (decodePhenotype), 5
  DLBCLExample, 6

  encodePhenotype, 5, 6
  encodePhenotype, character, character-method
    (encodePhenotype), 6
  encodePhenotype-methods
    (encodePhenotype), 6
  flowType, 4, 5, 7, 7, 11, 12
  flowType-package, 2
  getLabels, 9
  HIVData, 10
  HIVMetaData, 10
  Phenotypes (Phenotypes-class), 11
  Phenotypes-class, 11
  plot, 12
  plot, Phenotypes, character (plot), 12
  plot, Phenotypes, character-method
    (plot), 12
  plot, Phenotypes, flowFrame (plot), 12
  plot, Phenotypes, flowFrame-method
    (plot), 12
  plot, Phenotypes, numeric (plot), 12
  plot, Phenotypes, numeric-method (plot), 12
  summary, Phenotypes-method
    (summary-methods), 12
  summary-methods, 12