Package 'FlowSOM'

October 9, 2015

Version 1.0.0

Date 2014-10-16

Title Using self-organizing maps for visualization and interpretation of cytometry data

Author Sofie Van Gassen, Britt Callebaut and Yvan Saeys

Maintainer Sofie Van Gassen <sofie.vangassen@ugent.be>

Depends R (>= 2.11), flowCore, igraph, ConsensusClusterPlus, BiocGenerics, tsne

Suggests flowUtils, BiocStyle

Description FlowSOM offers visualization options for cytometry data, by using Self-Organizing Map clustering and Minimal Spanning Trees

License GPL (>= 2)

URL http://www.r-project.org, http://dambi.ugent.be

biocViews CellBiology, FlowCytometry, Clustering, Visualization, Software, CellBasedAssays

NeedsCompilation yes

R topics documented:

`AggregateFlowFrames` *Aggregate multiple fcs files together*

Description

Aggregate multiple fcs files to analyze them simultaneously. A new fcs file is written, which contains about `cTotal` cells, with `ceiling(cTotal/nFiles)` cells from each file. Two new columns are added: a column indicating the original file by index, and a noisy version of this for better plotting opportunities (index plus or minus a value between 0 and 0.1.

Usage

```
AggregateFlowFrames(fileNames, outputFile, cTotal,writeMeta=FALSE)
```

Arguments

`fileNames`	Character vector containing full paths to the fcs files to aggregate
`outputFile`	Full path to output file
`cTotal`	Total number of cells to write to the output file
`writeMeta`	If TRUE, files with the indices of the selected cells are generated

Value

This function does not return anything, but will write a file with about `cTotal` cells to `outputFile`

See Also

`ceiling`

Examples

```
# Define filename
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
# This example will sample 2 times 500 cells.
AggregateFlowFrames(c(fileName,fileName),"tmp.fcs",1000)
```

`BuildMST` *Build Minimal Spanning Tree*

Description

Add minimal spanning tree description to the FlowSOM object

Usage

```
BuildMST(fsom, silent=FALSE, tSNE=FALSE)
```

Arguments

`fsom`	FlowSOM object, as generated by `BuildSOM`
`silent`	if `TRUE`, no progress updates will be printed
`tSNE`	If `TRUE`, an alternative tSNE layout is computed as well

Value

FlowSOM object containing MST description

See Also

`BuildSOM`

Examples

```
# Read from file, build self-organizing map
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))

# Build the Minimal Spanning Tree
flowSOM.res <- BuildMST(flowSOM.res)
```

`BuildSOM` *Build a self-organizing map*

Description

Build a SOM based on the data contained in the FlowSOM object

Usage

```
BuildSOM(fsom, colsToUse=NULL, silent=FALSE, ...)
```

Arguments

fsom	FlowSOM object containing the data, as constructed by the ReadInput function
colsToUse	column names or indices to use for building the SOM
silent	if TRUE, no progress updates will be printed
...	options to pass on to the SOM function (xdim, ydim, rlen, mst, alpha, radius, init, distf)

Value

FlowSOM object containing the SOM result, which can be used as input for the BuildMST function

References

This code is strongly based on the kohonen package. R. Wehrens and L.M.C. Buydens, Self- and Super-organising Maps in R: the kohonen package J. Stat. Softw., 21(5), 2007

See Also

ReadInput,BuildMST

Examples

```
# Read from file
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)

# Build the Self-Organizing Map
# E.g. with gridsize 5x5, presenting the dataset 20 times,
# no use of MST in neighbourhood calculations in between
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18),
                         xdim=5,ydim=5,rlen=20)

# Build the minimal spanning tree and apply metaclustering
flowSOM.res <- BuildMST(flowSOM.res)
metacl <- MetaClustering(flowSOM.res$map$codes,
                         "metaClustering_consensus",max=10)
```

CountGroups *Compute differences between groups*

Description

Map new data to the flowSOM grid and average over all files from one group

Usage

```
CountGroups(fsom, groups, plot=TRUE, silent=FALSE)
```

Arguments

fsom	FlowSOM object, as generated by BuildMST
groups	List with for every group a vector with filepaths
plot	Logical: if TRUE, the star plot is shown for each individual file, with adapted node size
silent	Logical: if FALSE, progress is printed

Value

A list including the counts, percentages, means and medians

See Also

PlotStars,PlotGroups,codeNewData

Examples

```
## Use the wrapper function to build a flowSOM object (saved in fsom[[1]])
## and a metaclustering (saved in fsom[[2]])
# fsom <- FlowSOM(ff,compensate = FALSE, transform = FALSE,scale = TRUE,
#               colsToUse = colsToUse, nClus = 10, silent = FALSE,
#               xdim=7, ydim=7)

## Make a list with for each group a list of files
## The reference group should be the first
#groups <- list("C"=file.path(workDir,grep("C",files,value = TRUE)),
#          "D"=file.path(workDir,grep("D",files,value=TRUE)))

## Compute the percentages for all groups
# groups_res <- CountGroups(fsom[[1]],groups)

## Plot the groups. For all groups except the first, differences with the
## first group are indicated
# annotation <- PlotGroups(fsom[[1]],groups_res)
```

FlowSOM *Run the FlowSOM algorithm*

Description

Method to run general FlowSOM workflow. Will scale the data and uses consensus meta-clustering by default.

Usage

```
FlowSOM(input, pattern=".fcs", compensate=FALSE, spillover=NULL,
                transform=FALSE, toTransform=NULL, scale=TRUE,
                scaled.center=TRUE, scaled.scale=TRUE, silent=TRUE, colsToUse,
                nClus=NULL, maxMeta,...)
```

Arguments

input	a flowFrame, a flowSet or an array of paths to files or directories
pattern	if input is an array of file- or directorynames, select only files containing pattern
compensate	logical, does the data need to be compensated
spillover	spillover matrix to compensate with If NULL and compensate=TRUE, we will look for $SPILL description in fcs file.
transform	logical, does the data need to be transformed with a logicle transform
toTransform	column names or indices that need to be transformed. If NULL and transform = TRUE, column names of $SPILL description in fcs file will be used.
scale	logical, does the data needs to be rescaled
scaled.center	see scale
scaled.scale	see scale
silent	if TRUE, no progress updates will be printed
colsToUse	column names or indices to use for building the SOM
nClus	Exact number of clusters for meta-clustering. If NULL, several options will be tried (1:maxMeta)
maxMeta	Maximum number of clusters to try out for meta-clustering. Ignored if nClus is specified
...	options to pass on to the SOM function (xdim, ydim, rlen, mst, alpha, radius, init, distf)

Value

A list with two items: the first is the flowSOM object containing all information (see the vignette for more detailed information about this object), the second is the metaclustering of the nodes of the grid. This is a wrapper function for ReadInput, BuildSOM, BuildMST and MetaClustering. Executing them separately may provide more options.

See Also

scale,ReadInput,BuildSOM, BuildMST,MetaClustering

Examples

```
# Read from file
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- FlowSOM(fileName, compensate=TRUE,transform=TRUE,
                scale=TRUE,colsToUse=c(9,12,14:18),maxMeta=10)

# Or read from flowFrame object
ff <- read.FCS(fileName)
ff <- compensate(ff,ff@description$SPILL)
ff <- transform(ff,transformList(colnames(ff@description$SPILL),
                logicleTransform()))
flowSOM.res <- FlowSOM(ff,scale=TRUE,colsToUse=c(9,12,14:18),maxMeta=10)
```

```
# Get metaclustering per cell
flowSOM.clustering <- flowSOM.res[[2]][flowSOM.res[[1]]$map$mapping[,1]]
```

`FlowSOMSubset` *FlowSOM subset*

Description

Take a subset from a FlowSOM object

Usage

```
FlowSOMSubset(fsom,ids)
```

Arguments

`fsom`	FlowSOM object, as generated by `BuildMST`
`ids`	Array containing the ids to keep

Value

FlowSOM object containg updated data and meanvalues, but with the same grid

See Also

`BuildMST`

Examples

```
# Read two files (Artificially, as we just split 1 file in 2 subsets)
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
ff1 <- read.FCS(fileName)[1:1000,]
ff1@description$FIL <- "File1"
ff2 <- read.FCS(fileName)[1001:2000,]
ff2@description$FIL <- "File2"

flowSOM.res <- FlowSOM(flowSet(c(ff1,ff2)), compensate=TRUE,transform=TRUE,
                       scale=TRUE,colsToUse=c(9,12,14:18),maxMeta=10)
fSOM <- flowSOM.res[[1]]

# see $metadata for subsets:
fSOM$metaData

# Use only the second file, without changing the map
fSOM2 <- FlowSOMSubset(fSOM,
                    (fSOM$metaData[[2]][1]):(fSOM$metaData[[2]][2]))
```

FMeasure *FMeasure of a clustering result*

Description

Compute mean weighted F-Measure of a clustering result in comparison with true cluster labels

Usage

```
FMeasure(realClusters, predictedClusters, silent=FALSE)
```

Arguments

realClusters	array with real cluster values
predictedClusters	array with predicted cluster values
silent	logical. Should precision and recall values be printed or not

Value

FMeasure of the predicted clustering result

See Also

Purity

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Apply metaclustering
metacl <- MetaClustering(flowSOM.res$map$codes,
                         "metaClustering_consensus",max=10)

# Get the manually gated labels using a gatingML file
# and the flowUtils library
ff <- read.FCS(fileName)
ff_c <- compensate(ff,ff@description$SPILL)
colnames(ff_c)[8:18] <- paste("Comp-",colnames(ff_c)[8:18],sep="")
library(flowUtils)
flowEnv <- new.env()
gatingFile <- system.file("extdata","manualGating.xml",
                         package="FlowSOM")
read.gatingML(gatingFile, flowEnv)
```

```
filterList <- list( "B cells" = flowEnv$ID52300206,
                "ab T cells" = flowEnv$ID785879196,
                "yd T cells" = flowEnv$ID188379411,
                "NK cells" = flowEnv$ID1229333490,
                "NKT cells" = flowEnv$ID275096433
                )
results <- list()
for(cellType in names(filterList)){
    results[[cellType]] <- filter(ff_c,filterList[[cellType]])@subSet
}
manual <- rep("Unknown",nrow(ff))
for(celltype in names(results)){
    manual[results[[celltype]]] <- celltype
}

# Test the fmeasure of the result
FMeasure(manual,metacl[flowSOM.res$map$mapping[,1]])
```

MetaClustering *MetaClustering*

Description

Cluster data with automatic number of cluster determination for several algorithms

Usage

```
MetaClustering(data,method,max=20,...)
```

Arguments

data	Matrix containing the data to cluster
method	Clustering method to use
max	Maximum number of clusters to try out
...	Extra parameters to pass along

Value

Numeric array indicating cluster for each datapoint

See Also

metaClustering_consensus

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Apply metaclustering
metacl <- MetaClustering(flowSOM.res$map$codes,"metaClustering_consensus",
                         max=10)

# Get metaclustering per cell
flowSOM.clustering <- metacl[flowSOM.res$map$mapping[,1]]
```

---

metaClustering_consensus

MetaClustering

Description

Cluster data using hierarchical consensus clustering with k clusters

Usage

```
metaClustering_consensus(data, k=7)
```

Arguments

data	Matrix containing the data to cluster
k	Number of clusters

Value

Numeric array indicating cluster for each datapoint

See Also

MetaClustering

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)
```

```
# Apply consensus metaclustering
metacl <- metaClustering_consensus(flowSOM.res$map$codes,k=10)
```

NewData *Map new data to a FlowSOM grid*

Description

New data from a flowframe is mapped to an existing FlowSOM object. A new FlowSOM object is created, with the same grid, but a new mapping, node sizes and mean values. We assume the data is already compensated and transformed, but not scaled yet. The same scaling parameters as from the original grid will be used.

Usage

```
NewData(fsom,ff)
```

Arguments

fsom	FlowSOM object
ff	Flow frame with the data to map

Value

A new FlowSOM object

See Also

FlowSOMSubset if you want to get a subset of the current data instead of a new dataset

Examples

```
# Build FlowSom result
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
ff <- read.FCS(fileName)
ff <- compensate(ff,ff@description$SPILL)
ff <- transform(ff,transformList(colnames(ff@description$SPILL),
                logicleTransform()))
flowSOM.res <- FlowSOM(ff[1:1000,],scale=TRUE,colsToUse=c(9,12,14:18),
maxMeta=10)

# Map new data
print(colnames(ff[1001:2000,]))
fSOM2 <- NewData(flowSOM.res[[1]], ff[1001:2000,])
```

PlotCenters *Plot cluster centers on a 2D plot*

Description

Plot FlowSOM nodes on a 2D scatter plot of the data

Usage

```
PlotCenters(fsom, marker1, marker2, MST=TRUE)
```

Arguments

fsom	FlowSOM object, as generated by BuildMST
marker1	Marker to show on the x-axis
marker2	Marker to show on the y-axis
MST	logical. If TRUE, plot tree, else plot grid

Value

Nothing is returned. A 2D scatter plot is drawn on which the nodes of the grid are indicated

See Also

PlotStars,PlotPies,PlotMarker, BuildMST

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Plot centers
PlotCenters(flowSOM.res,"FSC-A","SSC-A")
PlotCenters(flowSOM.res,2,5)
```

PlotClusters2D *Plot nodes on scatter plot*

Description

Plot a 2D scatter plot. All cells of fsom$data are plotted in black, and those of the selected nodes are plotted in red. The nodes in the grid are indexed starting from the left bottom, first going right, then up. E.g. In a 10x10 grid, the node at top left will have index 91.

Usage

```
PlotClusters2D(fsom, marker1, marker2, nodes,main="")
```

Arguments

fsom	FlowSOM object, as generated by BuildMST
marker1	Marker to plot on the x-axis
marker2	Marker to plot on the y-axis
nodes	Nodes of which the cells should be plotted in red
main	Title of the plot

Value

Nothing is returned. A plot is drawn in which all cells are plotted in black and the cells of the selected nodes in red.

See Also

PlotCenters,BuildMST

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Plot cells
PlotClusters2D(flowSOM.res,1,2,91)
```

PlotGroups *Plot differences between groups*

Description

Plot FlowSOM trees, where each node is represented by a star chart indicating mean marker values, the size of the node is relative to the mean percentage of cells present in each

Usage

```
PlotGroups(fsom, groups, tresh=0.5, p_tresh=NULL,...)
```

Arguments

fsom	FlowSOM object, as generated by BuildMST or the first list item of FlowSOM
groups	groups result as generated by CountGroups
tresh	Relative difference in groups before the node is coloured
p_tresh	Threshold on p-value from t-test before the node is coloured. If this is not NULL, tresh will be ignored.
...	Other parameters to pass to PlotStars

Value

A vector containing the labels assigned to the nodes for all groups except the first

See Also

PlotStars,CountGroups

Examples

```
## Use the wrapper function to build a flowSOM object (saved in fsom[[1]])
## and a metaclustering (saved in fsom[[2]])
# fsom <- FlowSOM(ff,compensate = FALSE, transform = FALSE,scale = TRUE,
#               colsToUse = colsToUse, nClus = 10, silent = FALSE,
#               xdim=7, ydim=7)

## Make a list with for each group a list of files
## The reference group should be the first
#groups <- list("C"=file.path(workDir,grep("C",files,value = TRUE)),
#          "D"=file.path(workDir,grep("D",files,value=TRUE)))

## Compute the percentages for all groups
# groups_res <- CountGroups(fsom[[1]],groups)

## Plot the groups. For all groups except the first, differences with the
## first group are indicated
# annotation <- PlotGroups(fsom[[1]],groups_res)
```

PlotMarker *Plot marker values*

Description

Plot FlowSOM grid or tree, coloured by node values for a specific marker

Usage

```
PlotMarker(fsom, marker=NULL, MST=TRUE,main=NULL,
                colorPalette=colorRampPalette(c("#00007F", "blue", "#007FFF",
                "cyan","#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000")))
```

Arguments

fsom	FlowSOM object, as generated by BuildMST
marker	Name or index of marker to plot
MST	logical. If TRUE, plot tree, else plot grid
main	Title of the plot
colorPalette	Color palette to use

Value

Nothing is returned. A plot is drawn in which each node is coloured depending on its mean value for the given marker

References

This visualization technique resembles SPADE results. M. Linderman, P. Qiu, E. Simonds and Z. Bjornson (). spade: SPADE – An analysis and visualization tool for Flow Cytometry. R package version 1.12.2. http://cytospade.org

See Also

PlotStars,PlotPies,PlotCenters, BuildMST

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Plot one marker
PlotMarker(flowSOM.res,"FSC-A")
```

PlotPies *Plot comparison with other clustering*

Description

Plot FlowSOM grid or tree, with pies indicating another clustering or manual gating result

Usage

```
PlotPies(fsom, cellTypes, MST=TRUE, legend=TRUE, clusters=NULL,main="",
                colorPalette=colorRampPalette(c("white","#00007F", "blue",
                "#007FFF","cyan", "#7FFF7F", "yellow", "#FF7F00", "red")))
```

Arguments

fsom	FlowSOM object, as generated by BuildMST
cellTypes	Array of factors indicating the celltypes
MST	logical. If TRUE, plot tree, else plot grid
legend	logical. Sometimes the position of the legend is not great, so it might be easier to plot without
clusters	optional, clustering of the SOM nodes
main	Title of the plot
colorPalette	Color palette to use

Value

Nothing is returned. A plot is drawn in which each node is represented by a pie chart indicating the percentage of cells present of each cell type

See Also

PlotStars,PlotMarker,PlotCenters, BuildMST

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Get the manually gated labels using a gatingML file
# and the flowUtils library
ff <- read.FCS(fileName)
ff_c <- compensate(ff,ff@description$SPILL)
colnames(ff_c)[8:18] <- paste("Comp-",colnames(ff_c)[8:18],sep="")
```

```
library(flowUtils)
flowEnv <- new.env()
gatingFile <- system.file("extdata","manualGating.xml",
                          package="FlowSOM")
read.gatingML(gatingFile, flowEnv)
filterList <- list( "B cells" = flowEnv$ID52300206,
                "ab T cells" = flowEnv$ID785879196,
                "yd T cells" = flowEnv$ID188379411,
                "NK cells" = flowEnv$ID1229333490,
                "NKT cells" = flowEnv$ID275096433
                )
results <- list()
for(cellType in names(filterList)){
    results[[cellType]] <- filter(ff_c,filterList[[cellType]])@subSet
}
manual <- rep("Unknown",nrow(ff))
for(celltype in names(results)){
    manual[results[[celltype]]] <- celltype
}

# Plot pies indicating the percentage of cell types present in the nodes
PlotPies(flowSOM.res,manual)
```

PlotStars *Plot star charts*

Description

Plot FlowSOM grid or tree, where each node is represented by a star chart indicating mean marker values

Usage

```
PlotStars(fsom, markers=fsom$map$colsUsed, MST=1, legend=TRUE,
                clusters=NULL,main="",colorPalette=colorRampPalette(
                c("#00007F", "blue", "#007FFF", "cyan", "#7FFF7F", "yellow",
                "#FF7F00", "red", "#7F0000")),clusterColorPalette=
                function(n){rainbow(n,alpha=0.3)})
```

Arguments

fsom	FlowSOM object, as generated by BuildMST
markers	Array of markers to use. Default: the markers used to build the tree
MST	Numeric. If 1 plot tree, if 2 plot grid, if 3 plot tSNE
legend	logical. Sometimes the position of the legend is not great, so it might be easier to plot without
clusters	optional, clustering of the SOM nodes
main	Title of the plot

colorPalette Colorpalette to be used

clusterColorPalette
Colorpalette to be used for the metaclusters

Value

Nothing is returned. A plot is drawn in which each node is represented by a star chart indicating the mean fluorescence intensities

See Also

PlotPies,PlotMarker,PlotCenters, BuildMST

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Plot stars indicating the MFI of the cells present in the nodes
PlotStars(flowSOM.res)
```

Purity *Purity of a clustering result*

Description

Compute mean weighted purity of a clustering result in comparison with true cluster labels

Usage

```
Purity(realClusters, predictedClusters, weighted=TRUE)
```

Arguments

realClusters array with real cluster values

predictedClusters
array with predicted cluster values

weighted logical. Should the mean be weighted depending on the number of points in the predicted clusters

Value

Weighted mean purity value

See Also

FMeasure

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Apply metaclustering
metacl <- MetaClustering(flowSOM.res$map$codes,"metaClustering_consensus",
                         max=10)

# Get the manually gated labels using a gatingML file
# and the flowUtils library
ff <- read.FCS(fileName)
ff_c <- compensate(ff,ff@description$SPILL)
colnames(ff_c)[8:18] <- paste("Comp-",colnames(ff_c)[8:18],sep="")
library(flowUtils)
flowEnv <- new.env()
gatingFile <- system.file("extdata","manualGating.xml",
                         package="FlowSOM")
read.gatingML(gatingFile, flowEnv)
filterList <- list( "B cells" = flowEnv$ID52300206,
                "ab T cells" = flowEnv$ID785879196,
                "yd T cells" = flowEnv$ID188379411,
                "NK cells" = flowEnv$ID1229333490,
                "NKT cells" = flowEnv$ID275096433
                )
results <- list()
for(cellType in names(filterList)){
    results[[cellType]] <- filter(ff_c,filterList[[cellType]])@subSet
}
manual <- rep("Unknown",nrow(ff))
for(celltype in names(results)){
    manual[results[[celltype]]] <- celltype
}

# Test the purity of the nodes
Purity(manual,metacl[flowSOM.res$map$mapping[,1]])
```

ReadInput *Read fcs-files or flowframes*

Description

Take some input and return a matrix with preprocessed data (compensated, transformed, scaled)

Usage

```
ReadInput(input, pattern=".fcs", compensate=FALSE, spillover=NULL,
                transform=FALSE, toTransform=NULL, scale=FALSE,
                scaled.center=TRUE, scaled.scale=TRUE, silent=FALSE)
```

Arguments

input	a flowFrame, a flowSet or an array of paths to files or directories
pattern	if input is an array of file- or directorynames, select only files containing pattern
compensate	logical, does the data need to be compensated
spillover	spillover matrix to compensate with If NULL and compensate=TRUE, we will look for $SPILL description in fcs file.
transform	logical, does the data need to be transformed with a logicle transform
toTransform	column names or indices that need to be transformed. If NULL and transform=TRUE, column names of $SPILL description in fcs file will be used.
scale	logical, does the data needs to be rescaled
scaled.center	see scale
scaled.scale	see scale
silent	if TRUE, no progress updates will be printed

Value

FlowSOM object containing the data, which can be used as input for the BuildSOM function

See Also

scale,BuildSOM

Examples

```
# Read from file
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)

# Or read from flowFrame object
ff <- read.FCS(fileName)
ff <- compensate(ff,ff@description$SPILL)
ff <- transform(ff,transformList(colnames(ff@description$SPILL),
                 logicleTransform()))
flowSOM.res <- ReadInput(ff,scale=TRUE)

# Build the self-organizing map and the minimal spanning tree
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Apply metaclustering
```

```
metacl <- MetaClustering(flowSOM.res$map$codes,
                         "metaClustering_consensus",max=10)

# Get metaclustering per cell
flowSOM.clustering <- metacl[flowSOM.res$map$mapping[,1]]
```

UpdateNodeSize *Update nodesize of FlowSOM object*

Description

Add size property to the graph based on cellcount for each node

Usage

```
UpdateNodeSize(fsom, reset=FALSE, logScale=FALSE)
```

Arguments

fsom	FlowSOM object, as generated by BuildMST
reset	logical. If TRUE, all nodes get the same size
logScale	Logical. Use a log scaling on the cell counts

Value

Updated FlowSOM object

See Also

BuildMST

Examples

```
# Read from file, build self-organizing map and minimal spanning tree
fileName <- system.file("extdata","lymphocytes.fcs",package="FlowSOM")
flowSOM.res <- ReadInput(fileName, compensate=TRUE,transform=TRUE,
                         scale=TRUE)
flowSOM.res <- BuildSOM(flowSOM.res,colsToUse=c(9,12,14:18))
flowSOM.res <- BuildMST(flowSOM.res)

# Give all nodes same size
flowSOM.res <- UpdateNodeSize(flowSOM.res,reset=TRUE)
PlotStars(flowSOM.res)

# Node sizes relative to amount of cells assigned to the node
flowSOM.res <- UpdateNodeSize(flowSOM.res)
PlotStars(flowSOM.res)
```

Index