Package ‘flowWorkspace’

March 17, 2013

Type Package

Title Import flowJo Workspaces into BioConductor and replicate flowJo gating with flowCore

Version 1.4.0

Date 2011-06-10

Author Greg Finak, Mike Jiang, Mose Andre

Maintainer Greg Finak <gfinak@fhcrc.org>

Description This package is designed to facilitate comparison of automated gating methods against manual gating done in flowJo. This package allows you to import basic flowJo workspaces into BioConductor and replicate the gating from flowJo using the flowCore functionality. Gating hierarchies, groups of samples, compensation, and transformation are performed so that the output matches the flowJo analysis.

License Artistic-2.0

LazyLoad yes

Imports Biobase, XML, flowCore, graph, graphics, lattice, methods,stats, stats4, utils

Collate AllClasses.R AllGenerics.R AllMethods.R AllExportMethods.R flowWorkspace2flowCore-methods.R flowWorkspaceEdit-methods.R bitVector.R

biocViews FlowCytometry, DataImport, Preprocessing, DataRepresentation

Depends Cairo, BiocGenerics, methods, RBGL, graph, XML, flowCore,flowViz, Biobase, IDPmisc, tools,hexbin

Enhances multicore,Rmpi,ncdfFlow

Suggests testthat, flowWorkspaceData, Rgraphviz

R topics documented:

flowWorkspace-package 2
closeWorkspace 3
ellipsoidGate2FlowJoVertices 4
execute 5
exportAsFlowJoXML 6
ExportTSVAnalysis 8

flowJoWorkspace-class . . . 9
flowWorkspace2flowCore . . . 10
GatingHierarchy-class . . . 11
GatingSet-class . . . 14
getBoundaries . . . 15
getChildren . . . 16
getCompensationMatrices . . . 17
getData . . . 18
getDimensions . . . 19
getFJWSubsetIndices . . . 20
getGate . . . 21
getIndiceFile . . . 22
getIndices . . . 23
getKeywords . . . 24
getNcdf . . . 25
getNodes . . . 26
getParent . . . 27
getPopStats . . . 28
getProp . . . 29
getSample . . . 30
getSampleGroups . . . 31
getSamples . . . 32
getTotal . . . 33
getTransformations . . . 34
haveSameGatingHierarchy . . . 35
includedChannel2ExcludedChannel . . . 36
includedGate2ExcludedGate . . . 36
keyword-methods . . . 37
lapply-methods . . . 37
length-methods . . . 38
openWorkspace . . . 38
parseWorkspace . . . 39
plot . . . 40
plotGate . . . 41
plotPopCV . . . 43
plotWf . . . 44
recomputeGatingSet . . . 44
saveNcdf . . . 45
summary-methods . . . 45

Index **46**

---

flowWorkspace-package *Import and replicate flowJo workspaces and gating schemes using flowCore.*

Description

Import flowJo workspaces into R. Generate the flowJo gating hierarchy and gates using flowCore functionality. Transform and compensate data in accordance with flowJo settings. Plot gates, gating hierarchies, population statistics, and compare flowJo vs flowCore population summaries.

Details

Package:	flowWorkspace
Type:	Package
Version:	0.5.40
Date:	2011-03-04
License:	Artistic 2.0
LazyLoad:	yes
Depends:	methods, RBGL, graph, XML, flowCore, flowViz, Rgraphviz, Biobase

Author(s)

Greg Finak, Mike Jiang, Mose Andre

Maintainer: Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

---

closeWorkspace *Close a flowJoWorkspace*

Description

Close a flowJoWorkspace, destroying the internal representation of the XML document, and freeing the associated memory.

Usage

```
## S4 method for signature 'flowJoWorkspace'
closeWorkspace(workspace)
```

Arguments

workspace	A flowJoWorkspace

Details

* Close a flowJoWorkpsace after finishing with it. This is necessary to explicitly clean up the C-based representation of the XML tree. (See the XML package).

Value

This function doesn't return anything.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

xmlTreeParse

Examples

```
## Not run:
ws<-openWorkspace("myworkspace.xml");
G<-parseWorkspace(ws,path=".") #path to fcs files, will search recursively
closeWorkspace(ws);

## End(Not run)
```

---

ellipsoidGate2FlowJoVertices

Get the vertices of an ellipsoid gate needed to export it to flowJo

---

Description

Fetches the top, bottom, right, and leftmost points of an ellipsoid gate (represented by a covariance matrix), for export to flowJo.

Usage

```
## S4 method for signature 'ellipsoidGate'
ellipsoidGate2FlowJoVertices(gate, level = 0.95,...)
```

Arguments

gate	ellipsoidGate object.
level	numeric. The quantile of the ellipse to be retrieved. Defaults to 0.95 (95%)
...	Additional arguments. Currently not used.

Details

flowJo's XML representation of ellipsoid gates uses the top, bottom,right, and leftmost points of the ellipse. This function retrieves those vertices given an ellipsoidGate flowCore object.

Value

A matrix with two columns. Rows are the top, bottom, right, and leftmost points on the ellipse (rotated).

Note

The implementation of this function is based on the ellipse function in the ellipse package.

Author(s)

Greg Finak <gfinak@fhrcr.org>

References

http://www.rglab.org/

See Also

ellipse

Examples

```
require(flowCore)
e<-ellipsoidGate(.gate={d<-diag(2);colnames(d)<-c("A","B");d},mean=c(2,2))
ellipsoidGate2FlowJoVertices(e);
```

---

execute — *Apply the GatingHierarchy to data, computing population statistics along the way.*

Description

A GatingHierarchy is associated with an fcs file. Calling execute on the GatingHierarchy will load the fcs file, perform compensation and transformation, and calculate the gates as described in the flowJoWorkspace. The method is not meant to be called by the user, but is used internally by flowWorkspace.

Usage

```
## S4 method for signature 'GatingHierarchy'
execute(hierarchy, cleanup=FALSE,keep.indices=TRUE,isNcdf=FALSE,ncfs=NULL,dataEnvironment=NULL
```

Arguments

hierarchy	A GatingHierarchy object
cleanup	cleanup=TRUE|FALSE When not using netcdf, this logical flag indicates whether the data should be retained in memory after processing or whether it should be scrapped to save RAM, keeping only population statistics. If you are loading a large data set, you may want to consider using netcdf, or setting this to TRUE. However, you will not be able to visualize the results.
keep.indices	keep.indices=TRUE|FALSE Logical indicating whether the indices calculated from gating should be stored, or deleted, leaving just their counts.
isNcdf	TRUE|FALSE a logical flag indicating whether the data (FlowFrame) is saved on the disc in netCDF format.
ncfs	ncdfFlowSet a ncdfFlowSet object created by parseWorkspace(when isNcdf is set as TRUE) which contains information of fcs metaData and netCDF file that stores the real data .
dataEnvironment	environment for storing ncdfFlowSet common to all GatingHierarchies in a GatingSet.

...	Additional arguments.
	path="character" A file path to the fcs file or files.

Details

This method is not meant to be called by the user. Rather, passing execute=TRUE to parseWorkspace will execute the gating scheme after the flowJo workspace has been loaded. Cleanup is FALSE by default. This may lead to memory issues when you have lots of data, but it is necessary to visualize the analysis. Netcdf is strongly recommended.

Value

Returns a GatingHierarchy with calculated population statistics and gate indices.

Note

This function is not meant to be called by the user. Gating of samples in a flowJo workspace can be invoked by passing execute=TRUE to parseWorkspace.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

parseWorkspace

Examples

```
## Not run:
ws<-openWorkspace("myworkspace.xml");
G<-parseWorkspace(ws,execute=TRUE,isNcdf=FALSE);

## End(Not run)
```

---

exportAsFlowJoXML *Export a workFlow to FlowJo XML*

Description

Converts a workFlow and associated metadata to a FlowJo 9.2 OSX compatible workspace.

Usage

```
## S4 method for signature 'workFlow'
exportAsFlowJoXML(obj, file=NULL,...)
## S4 method for signature 'list'
exportAsFlowJoXML(obj, file=NULL,...)
## S4 method for signature 'ellipsoidGate'
exportAsFlowJoXML(obj,transforms,...)
## S4 method for signature 'polygonGate'
exportAsFlowJoXML(obj,transforms,...)
## S4 method for signature 'rectangleGate'
exportAsFlowJoXML(obj,transforms,...)
## S4 method for signature 'intersectFilter'
exportAsFlowJoXML(obj,transforms,gate_view,workflow)
```

Arguments

obj	workFlow workFlow to be exported.
list	list list of workFlows to be exported.
file	character the name of the XML output file.
transforms	function the transform from raw scale to channel space
gate_view	gate_view the view from the flowCore workflow referencing the gates to be combined
workflow	workflow the flowCore workFlow this filter is used in
...	Additional arguments used by some S4 methods.

Details

Exports a flowCore workFlow object to an XML workspace readable by FlowJo.

Value

If file is NULL it will return the string for the XML output. Otherwise it returns the file name.

Author(s)

Mose Andre <mandre@fhcrc.org>

References

http://www.rglab.org/

See Also

workFlow-class

Examples

```
## Not run:
#Assume w is a workFlow object
exportAsFlowJoXML(w, "export.xml")

## End(Not run)
```

ExportTSVAnalysis *Export a GatingSet as a set of TSV (tab separated value) files for statistics, gates, and plots.*

Description

This function exports a GatingSet as a set of TSV files for import by other tools such as LabKey.

Usage

```
ExportTSVAnalysis(x = NULL, Keywords = NULL, EXPORT = "export")
```

Arguments

x	A GatingSet to be exported.
Keywords	A character vector with a set of keywords used to annotate the GatingSet. (currently not used)
EXPORT	A character vector. The directory into which the exported files will be placed.

Details

The function generates two tsv files. The first statistics.tsv has the sample names, population names, sample counts and fraction of parent population for all samples and populations in the GatingSet. The second gates.tsv has the population name, sample name, and path to the plots for each gate/population of each sample in the GatingSet. Inside the EXPORT directory it creates a subdirectory for each sample, named after that sample. The subdirectories hold png plots of the gates / populations defined for those samples. The names of the png files are based on the md5 hash of the png file itself. File names are in the graphs.tsv file.

Value

The function doesn't return anything.

Author(s)

Greg Finak <gfinak@fhcrc.org>

See Also

GatingSet-class

flowJoWorkspace-class *Class* "flowJoWorkspace"

Description

An R representation of a flowJo workspace.

Objects from the Class

Objects can be created by calls of the form new("flowJoWorkspace.xml", ...).

Slots

version: Object of class "character". The version of the XML workspace.

file: Object of class "character". The file name.

.cache: Object of class "environment". An environment for internal use.

path: Object of class "character". The path to the file.

doc: Object of class "XMLInternalDocument". The XML document object.

Methods

closeWorkspace signature(workspace = "flowJoWorkspace"): Close the workspace file and delete the C representation of the XML document, freeing memory.

flowWorkspace2flowCore signature(obj = "flowJoWorkspace"): Convert a flowJo workspace to a flowCore workflow

getCompensationMatrices signature(x = "flowJoWorkspace"): Retrieve the compensation matrices in the flowJo workspace.

getKeywords signature(obj = "flowJoWorkspace", y = "character"): Get the keywords for sample y from the flowJo workspace

getSampleGroups signature(x = "flowJoWorkspace"): Get the sample groups defined in the flowJo workspace.

getSamples signature(x = "flowJoWorkspace"): Get the samples listed in the flowJo workspace.

getTransformations signature(x = "flowJoWorkspace"): Get the data transformations listed in the flowJo workspace

parseWorkspace signature(obj = "flowJoWorkspace"): Parse a workspace, creating a GatingSet.

show signature(object = "flowJoWorkspace"): Print information about a workspace

summary signature(object = "flowJoWorkspace"): Summarize

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

GatingSet GatingHierarchy

Examples

```
require(flowWorkspaceData)
d<-system.file("extdata",package="flowWorkspaceData")
wsfile<-list.files(d,pattern="A2004Analysis.xml",full=TRUE)
ws <- openWorkspace(wsfile);
summary(ws)
getSamples(ws)
```

---

flowWorkspace2flowCore *Convert the GatingHierarchies in a GatingSet to a flowCore workflow.*

Description

Extract the compensation matrices,transformation functions and all the gates from GatingHierarchies in a GatingSet generated by the flowWorkspace package, and convert them to the respective views and actionItems of workFlows defined by flowCore package.

Usage

```
## S4 method for signature 'GatingSet'
flowWorkspace2flowCore(obj, ...)
## S4 method for signature 'GatingHierarchy'
flowWorkspace2flowCore(obj, ...)
## S4 method for signature 'flowJoWorkspace'
flowWorkspace2flowCore(obj, ...)
```

Arguments

obj	can be a flowJoWorkspace,GatingSet or a GatingHierarchy
...	Additional arugments. path="character" a file path to the fcs file or files. groupId="integer" a number indicating which group of the data (FlowFrame) should be processed when obj is a flowJoWorkspace. isCompare a logical flag indicating whether the gatingHierarchies should be compared and merged when they have the same structure if a flowJoWorkspace or GatingSet is provided as the input,default is TRUE

Details

When the function is applied to a flowJoWorkspace or GatingSet, it compares gating hierarchies and generate one workflow object for multiple samples if they have the same gating hierarchy structure. When obj is a flowJoWorkspace it first calls parseWorkspace function to parse the Workspace and generate GatingSet object and then convert the GatingSet to workflows.

Value

Returns a workflow if obj is a GatingHierarchy. Returns a list of workflows if obj is a flowJoWorkspace or a GatingSet.

Author(s)

Mike Jiang <wjiang2@fhcrc.org>

References

http://www.rglab.org/

See Also

GatingSet-class GatingHierarchy-class flowJoWorkspace-class parseWorkspace

Examples

```
##locate workspace xml file and fcs files
dataDir <- system.file("extdata", package = "flowWorkspace")
wsfile<-list.files(dataDir,pattern="xml",full=TRUE)[1]

##open workspace xml file
## Not run:
ws<-openWorkspace(wsfile)

##Convert a flowWorkspace to workFlows
wfs<-flowWorkspace2flowCore(ws,groupId=1,path=dataDir)
plotWf(wfs[[1]])

##parse workspace and convert a GatingSet to workFlows
G <- parseWorkspace(ws,execute=TRUE,name=1,path=dataDir)
wfs<-flowWorkspace2flowCore(G,isCompare=TRUE,path=dataDir)
plotWf(wfs[[1]])

##Convert a GatingHierarchy to workFlow
wf <- flowWorkspace2flowCore(G[[1]],path=dataDir)
plotWf(wf)

## End(Not run)
```

GatingHierarchy-class *Class* "GatingHierarchy"

Description

GatingHierarchy is a class for representing the gating hierarchy imported from a flowJo workspace.

Details

There is a one-to-one correspondence between GatingHierarchy objects and FCS files in the flowJo workspace. Each sample (FCS file) is associated with it's own GatingHierarchy. This is different from the workflow representation used in flowCore.

A GatingHierarchy can have two "states". After a call to parseWorkspace(...,execute=FALSE), the workspace is imported but the data is not. A call to execute() is needed in order to load, transform, compensate, and gate the associated data. Alternately, one may call parseWorkspace(...,execute=TRUE).

Whether or not a GatingHierarchy has been applied to data is encoded in the flag slot. Some methods will warn the user, or may not function correctly if the GatingHierarchy has not been execute()d.This mechanism is in place, largely for the purpose of memory efficiency when working with larger workspaces. It allows the use to load a workspace and subset desired samples before proceeding to load the data. If one has netCDF 4 library installed, then memory is no longer an issue.

Given a GatingHierarchy, one can extract the data associated with any subpopulation, extract gates, plot gates, and extract population proportions. This facilitates the comparison of manual gating methods with automated gating algorithms.

GatingHierarchy objects can be converted to workflows.

Objects from the Class

GatingHierarchy objects are elements of a GatingSet, which is returned by a call to parseWorkspace().

Slots

tree**:** Object of class "graphNEL" representing the tree-structured gating hierarchy.

nodes**:** Object of class "character". A vector of node names representing the populations/gates in the tree.

name**:** Object of class "character". The name of the sample. Usually the FCS filename, but it depends on how it was defined in the flowJo workspace.

flag**:** Object of class "logical". A flag indicating whether the gates, transformations, and compensation matrices have been applied to data, or simply imported.

transformations**:** Object of class "list". The list of transformations applied to each dimension of the data.

compensation**:** Object of class "matrix". The compensation matrix applied to the data

dataPath**:** Object of class "character". A path to the fcs file associated with this GatingHierarchy

isNcdf**:** Flag indicating whether ncdf is used to store data and gating indices

Methods

[[<- signature(x = "GatingSet", i = "ANY", j = "ANY", value = "GatingHierarchy"): replacement method for GatingHierarchy objects within a GatingSet

execute signature(hierarchy = "GatingHierarchy"): Apply the compensation, transformation, and gating of a GatingHierarchy to its associated fcs file.

flowWorkspace2flowCore signature(obj = "GatingHierarchy"): convert a GatingHierarchy to a flowCore workflow.

getBoundaries signature(obj = "GatingHierarchy", y = "character"): Get the vertices of gate y in GatingHierarchy obj.

getChildren signature(obj = "GatingHierarchy", y = "character"): Get the child nodes of population y in obj.

getData signature(obj = "GatingHierarchy"): Return a flowFrame for population y in GatingHierarchy obj.

getDimensions signature(obj = "GatingHierarchy", y = "character"): Get the dimensions for the gate of population y

getGate signature(obj = "GatingHierarchy", y = "character"): Return the gate for population y.

getGate signature(obj = "GatingHierarchy", y = "numeric"): REturn the gate for population y, by index rather than name.

getIndices signature(obj = "GatingHierarchy", y = "character"): Return the event membership indices for population y.

getKeywords signature(obj = "GatingHierarchy", y = "missing"): Return the keywords for the GatingHierarchy

getNodes signature(x = "GatingHierarchy"): Return the node list for the gating hierarchy.

getParent signature(obj = "GatingHierarchy", y = "character"): Get the parent node of a population.

getParent signature(obj = "GatingHierarchy", y = "numeric"): Get the parent node of a population, by index.

getPopStats signature(x = "GatingHierarchy"): Return a table of population statistics (proportions and counts) for a gating hierarchy

getProp signature(x = "GatingHierarchy", y = "character"): return the population proportion for population y.

getSample signature(x = "GatingHierarchy"): Return the sample name of the gating hierarchy

getTotal signature(x = "GatingHierarchy", y = "character"): get the total number of events in population y

keyword signature(object = "GatingHierarchy", keyword = "character"): get a specific keyword from the gating hierarchy

plot signature(x = "GatingHierarchy", y = "missing"): plot a gating hierarchy graph

plotGate signature(x = "GatingHierarchy", y = "character"): plot a manual gate for the population over the parent data.

plotGate signature(x = "GatingHierarchy", y = "numeric"): plot a manual gate for a population referenced by index, over the parent data

plotPopCV signature(x = "GatingHierarchy"): plot the coefficient of variation for all populations, between flowCore and flowJo counts.

show signature(object = "GatingHierarchy"): Summarize a Gating Hierarchy

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/import-flowjo-workspaces-into-r-bioconductor/

See Also

parseWorkspace GatingSet

Examples

```
require(flowWorkspaceData)
d<-system.file("extdata",package="flowWorkspaceData")
wsfile<-list.files(d,pattern="A2004Analysis.xml",full=TRUE)
ws <- openWorkspace(wsfile);
G<-try(parseWorkspace(ws,execute=TRUE,path=d,name=1));
getPopStats(G[[1]]);
```

```
plotPopCV(G[[1]])
plotGate(G[[1]],getNodes(G[[1]])[4]);
getGate(G[[1]],getNodes(G[[1]])[4]);
getBoundaries(G[[1]],getNodes(G[[1]])[4])
getData(G[[1]],getNodes(G[[1]])[4])
```

GatingSet-class *Class* "GatingSet"

Description

Class that holds a set of GatingHierarchy objects, representing a set of samples and the gating scheme associated with each.

Objects from the Class

Objects can be created by a call to parseWorkspace(). The annotated data frame can be populated with the keywords from each sample.

Description

Objects store a collection of GatingHierarchies and represent a group in a flowJo workspace.

Slots

set**:** Object of class "list". A list of GatingHierarchy objects

metadata**:** Object of class "AnnotatedDataFrame". Stores the metadata associated with this set of FCS samples.

Methods

[signature(x = "GatingSet", i = "ANY"): Subset a GatingSet using the familiar bracket notation

[<- signature(x = "GatingSet", i = "ANY", j = "ANY", value = "GatingSet"): Replace elements of a GatingSet.

[[signature(x = "GatingSet", i = "ANY"): Extract a GatingHierarchy from a GatingSet

[[<- signature(x = "GatingSet", i = "ANY", j = "ANY", value = "GatingHierarchy"): Replace a GatingHierarchy in a GatingSet

flowWorkspace2flowCore signature(obj = "GatingSet"): Convert a GatingSet to a single workflow (if they have a common set of gates) or list of workflows if the GatingHierarchies differ.

getData signature(obj = "GatingSet"): Return a flowSet for the GatingSet

getGate signature(obj = "GatingSet", y = "numeric"): Return a flowSet for a subpopulation of each GatingHierarchy, numerically indexed.

getKeywords signature(obj = "GatingSet", y = "character"): Get the keywords associated with sample y

getKeywords signature(obj = "GatingSet", y = "numeric"): Get the keywords associated with sample y, numerical index.

getSamples signature(x = "GatingSet"): Get the sample names of the GatingHierarchies in this GatingSet

keyword signature(object = "GatingSet", keyword = "character"): Get the specific keyword for all samples in this GatingSet.

lapply signature(X = "GatingSet"): lapply method for GatingSet

length signature(x = "GatingSet"): Return the length of the GatingSet, number of GatingHierarhcy objects

plotPopCV signature(x = "GatingSet"): plot the population coefficients of variation between flowJo and flowCore for all populations and all samples

show signature(object = "GatingSet"): Print information about the GatingSet.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/import-flowjo-workspaces-into-r-bioconductor/

See Also

AnnotatedDataFrame GatingHierarchy flowJoWorkspace

Examples

```
require(flowWorkspaceData)
d<-system.file("extdata",package="flowWorkspaceData")
wsfile<-list.files(d,pattern="A2004Analysis.xml",full=TRUE)
ws <- openWorkspace(wsfile);
G<-try(parseWorkspace(ws,execute=TRUE,path=d,name=1));
plotPopCV(G);
```

---

getBoundaries *Get the boundaries of a flowJo gate*

Description

Get the boundaries (vertices) of a flowJo gate, on the transformed scale.

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getBoundaries(obj, y)
```

Arguments

obj	A GatingHierarchy
y	A character, the name of the node / gate / population of interest whose gate boundaries you wish to return.

Details

Each node in a GatingHierarchy represents a population. That population is defined by a gate. getBoundaries will return the vertices of the gate.

Value

A matrix with column names corresponding to channels / dimensions, and rows to x,y tuples of vertices for polygon gates in these dimensions.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org

See Also

getGate

Examples

```
## Not run:
file<-"myworkspace.xml"
ws<-openWorkspace(file)
G<-parseWorkspace(ws,execute=TRUE,path=".")
n<-getNodes(G[[1]],tsort=TRUE)[3] #get the third node in the first gating hierarchy (topological sort order)
getGate(G[[1]],n); #return the gate for that node.

## End(Not run)
```

---

getChildren — *Returns a list of child populations of the current node in the* GatingHierarchy

---

Description

Returns a character vector of all the children of the current node in the GatingHierarchy

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getChildren(obj, y)
```

Arguments

obj	a GatingHierarchy
y	a character name of the node / population.

Details

Get the child nodes / populations of the given node / population, y in the GatingHierarchy obj

Value

A character vector of the names of the child nodes of the current node. An empty vector if the node has no children.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

Examples

```
## Not run:
#G is a GatingHierarchy
n<-getNodes(G,tsort=T)[4];
getChildren(G,n);#Get the names of the child nodes of the 4th node in this gating hierarchy.

## End(Not run)
```

---

getCompensationMatrices

Retrieve the compensation matrices from a flowJo Workspace

Description

Retrieve all the compensation matrices from a flowJo workspace

Usage

```
## S4 method for signature 'flowJoWorkspace'
getCompensationMatrices(x)
```

Arguments

x	A flowJoWorkspace object.

Details

Return all the compensation matrices in a flowJoWorkspace object.

Value

A list of matrix representing the spillover matrices in the flowJoWorkspace

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

openWorkspace

Examples

```
## Not run:
#ws is a flowJoWorkspace
file<-"myworkspace.xml"
ws<-openWorkspace(file)
getCompensationMatrices(ws);

## End(Not run)
```

---

getData *Return the flowFrame associated with a GatingHierarchy*

Description

Return the flowFrame associated with a GatingHierarchy

Usage

```
## S4 method for signature 'GatingHierarchy'
getData(obj,y=NULL,tsort=FALSE)
## S4 method for signature 'GatingSet'
getData(obj,y=NULL,tsort=FALSE)
## S4 method for signature 'graphNEL'
getData(obj,y=NULL,tsort=FALSE)
```

Arguments

obj	A GatingHierarchy, GatingSet, or graphNEL object extracted from a GatingHierarchy @tree slot.
y	character node name or numeric node index. Default is NULL. If obj is a GatingHierarchy or graphNEL, y is the name of the node in obj for which you wish to extract the data or a numeric index into getNodes(obj). If obj is a GatingSet, y is a numeric index into getNodes(obj[[i]]), where i is any GatingHierarchy in the GatingSet. The trees represented by the GatingHierarchies are ASSUMED to be the same. Defaults to NULL, will return the complete flowFrame at the root node.
tsort	TRUE|FALSE logical indicating whether to retrieve the node list in topological sort order. Defaults to FALSE for internal compatibility.

Details

Returns a flowFrame containing the events in the gate defined at node y. Subset membership can be obtained using getIndices. Population statistics can be obtained using getPop and getPopStats. When calling getData on a GatingSet, y="numeric" only. Furthermore, the trees representing the GatingHierarchy for each sample in the GaingSet are presumed to have the same structure, facilitating identical node ordering with tsort=TRUE, and numeric indexing of the nodes.

Value

A flowFrame object if obj is a GatingHierarchy or graphNEL. A flowSet if

Note

The argument tsort ensures that the nodes are ordered in topological sort order. This is useful if you are using numeric node indices to access data across two or more identical trees (GatingHierarchies) with different node names (population names).

WARNING The trees in a GatingSet are assumed to have the same structure, such that a topological sort of the nodes in any tree will return the populations in the same order.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

getIndices getProp getPopStats

Examples

```
## Not run:
#G is a GatingSet
geData(G,3) #get a flowSet constructed from the third node / population in the tree.

#G is a GatingHierarchy
getData(G,)


## End(Not run)
```

---

getDimensions — *Return the dimensions on which a gate is applied within a GatingHierarchy*

Description

Return the dimension names on which a gate in a GatingHierarchy is applied.

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getDimensions(obj, y, index=FALSE)
```

Arguments

obj	A GatingHierarchy
y	The name of the population (node) in the GatingHierarchy for which you want the dimension names
index	TRUE | FALSE a logical indicating whether we should return the names of the dimensions (FALSE, default) or the indices of the dimensions (TRUE)

Value

A character vector of dimension names on which the gate is applied (when index=FALSE), or a numeric vector of the indices of the dimensions on which the gate is applied (when index=TRUE).

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org

See Also

getNodes

Examples

```
## Not run:
#G is a GatingHierarchy
#Fetch the dimensions for the fifth population in the hierarchy.
getDimensions(G,getNodes(G)[5],index=FALSE)

## End(Not run)
```

---

getFJWSubsetIndices — *Fetch the indices for a subset of samples in a flowJo workspace, based on a keyword value pair*

Description

Returns an index vector into the samples in a flowJo workspace for use with parseWorkspace(subset=), based on a keyword/value filter in a specific group of samples.

Usage

```
getFJWSubsetIndices(ws, key=NULL, value=NULL, group,requiregates=TRUE)
```

Arguments

ws	The flowJoWorkspace object
key	The name of the keyword. Type "character"
value	The value of the keyword. Type "character"
group	The group of samples to subset. Type numeric.
requiregates	TRUE or FALSE, specifying whether we include only samples that have gates attached or whether we include any sample in the workspace.

Details

This function will calculate the indices of a subset of samples in a flowJoWorkspace, based on a keyword/value filter. It is applied to a specific group of samples in the workspace. The output is meant to be passed to the subset= argument of parseWorkspace.

Value

A numeric vector of indices.

Author(s)

Greg Finak <gfinak@fhcrc.org>

See Also

parseWorkspace

---

getGate — *Return the flowCore gate definition associated with a node in a GatingHierarchy.*

Description

Return the flowCore gate definition object associated with a node in a GatingHierarchy object.

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getGate(obj, y)
## S4 method for signature 'GatingHierarchy,numeric'
getGate(obj, y,tsort=FALSE)
## S4 method for signature 'GatingSet,numeric'
getGate(obj, y,tsort=FALSE)
```

Arguments

obj	A GatingHierrarchy or GatingSet
y	A character when obj is a GatingHierarchy: the name of the node of interest. Or, a numeric when obj is either a GatingHierarchy or GatingSet. An index into the node list of nodes in the GatingHierarchy or GatingSet.
tsort	TRUE|FALSE if true, return the index y will access the nodes in topological sort order.

Value

A gate object from flowCore. Usually a polygonGate, but may be a rectangleGate. Boolean gates are represented by a "BooleanGate" S3 class. This is a list boolean gate definition that references populations in the GatingHierarchy and how they are to be combined logically. If obj is a GatingSet, assuming the trees associated with each GatingHierarchy are identical, then this method will return a list of gates, one for each sample in the GatingSet corresponding to the same population indexed by y.

Note

You should not have to deal with boolean gates. It is sufficient to retrieve the contents of a boolean gate node with getData.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org

See Also

getData getNodes

Examples

```
## Not run:  #G is a GatingHierarchy
getGate(G,5,tsort=TRUE) #return the gate for the fifth node in the tree.
getGate(G,getNodes(G,tsort=TRUE)[5]) #return the gate for the fifth node in the tree, but fetch it by name.
#G is a GatingSet
getGate(G,5,tsort=TRUE) #return a list of gates for the fifth node in each tree, assuming topological sort ordering.

## End(Not run)
```

---

getIndiceFile — *Returns the .nc file used to store the gating indices for the GatingHierarchy*

Description

If netCDF is being used to store the data, then this function returns the .nc file used to store the gating indices for the GatingHierarchy.

Usage

```
getIndiceFile(obj)
```

Arguments

obj	A GatingHierarchy

Details

Returns the .nc file used to store the gate indices for this GatingHierarchy. Used internally, not meant for user consumption at this time.

Value

A character vector, the name of the .nc file.

Author(s)

Greg Finak <gfinak@fhcrc.org>

getIndices *Get the membership indices for each event with respect to a particular gate in a GatingHierarchy*

Description

Returns a logical vector that describes whether each event in a sample is included or excluded by this gate.

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getIndices(obj, y)
```

Arguments

obj	A GatingHierarchy representing a sample.
y	A character giving the name of the population / node of interest.

Details

Returns a logical vector that describes whether each event in the data file is included in the given gate of this GatingHierarchy. The indices are for all events in the file, and do not reflect the population counts relative to the parent but relative to the root. To get population frequencies relative to the parent one cross-tabulate the indices of y with the indices of its parent.

Value

A logical vector of length equal to the number of events in the FCS file that determines whether each event is or is not included in the current gate.

Warning

The indices returned reference all events in the file and are not directly suitable for computing population statistics, unless subsets are taken with respect to the parent populations.

Note

Generally you should not need to use getIndices but the more convenient methods getProp and getPopStats which return population frequencies relative to the parent node.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org

See Also

See also getProp, getPopStats.

Examples

```
## Not run:
#G is a gating hierarchy
#Return the indices for population 5 (topological sort)
getIndices(G,getNodes(G,tsort=TRUE)[5]);

## End(Not run)
```

---

getKeywords *Get List of Keywords for a Flow Sample*

Description

Retrieve the list of keywords associated with a sample

Usage

```
## S4 method for signature 'GatingHierarchy,missing'
getKeywords(obj, y)
## S4 method for signature 'GatingSet,character'
getKeywords(obj, y)
## S4 method for signature 'GatingSet,numeric'
getKeywords(obj, y)
## S4 method for signature 'flowJoWorkspace,character'
getKeywords(obj, y)
```

Arguments

obj	A flowJoWorkspace, GatingSet, or GatingHierarchy
y	can be omitted if obj is a GatingHierarchy. A character, or numeric if obj is a GatingSet. A character if obj is a flowJoWorkspace

Details

Retrieve a list of keywords from a flowJoWorkspace, GatingSet, or GatingHierarchy for a particular sample. The sample is specified via y, either a numeric index into a GatingSet, or a sample name (character) for all other types of obj.

Value

A list of keyword - value pairs.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

Examples

```
## Not run:
#G is a GatingHierarchy
getKeywords(G);
#G is a GatingSet
getKeywords(G[[1]])
getKeywords(G,1)

## End(Not run)
```

---

getNcdf *Return the ncdfFlow object storing the data*

Description

Returns the netcdfFlowSet object that stores the data

Usage

```
## S4 method for signature 'GatingHierarchy'
getNcdf(obj)
## S4 method for signature 'GatingSet'
getNcdf(obj)
```

Arguments

obj	A GatingHierarchy.

Details

If a netcdfFlow object is storing the data in this GatingHierarchy, the function will return it.

Value

Returns a netcdfFlowSet if one is availble. Null otherwise

Author(s)

Greg Finak <gfinak@fhcrc.org>

See Also

See also getData

getNodes *Get the names of all nodes in a gating hierarchy*

Description

Returns a character vector of names of the nodes (populations) in the GatingHierarchy.

Usage

```
## S4 method for signature 'GatingHierarchy'
getNodes(x,tsort=FALSE,...)
```

Arguments

x	A GatingHierarchy
tsort	tsort=TRUE|FALSE returns the nodes in topological sort (TRUE) order.
...	Additional arguments.

Value

a character vector of node/population names, ordered appropriately.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

tsort

Examples

```
## Not run:
#G is a gating hierarchy
getNodes(G,tsort=TRUE)#return nodes in topological sort order.

## End(Not run)
```

getParent *Return the name of the parent population of the current population in the GatingHierarchy*

Description

Returns the name of the parent population of the current population in the given GatingHierarchy

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getParent(obj, y)
## S4 method for signature 'GatingHierarchy,numeric'
getParent(obj, y, tsort=FALSE)
```

Arguments

obj	A GatingHierarchy
y	The population whose parent you want to retrieve, either character or numeric
tsort	logical If TRUE, nodes are ordered by topological sort. Important when comparing across identical trees and using numeric indices. Default FALSE.

Value

Returns a character vector, the name of the parent population.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

getChildren

Examples

```
## Not run:
#G is a gatinghierarchy
#return the name of the parent of the fifth node in the hierarchy.
getParent(G,getNodes(G,tsort=TRUE)[5])

## End(Not run)
```

getPopStats *Return a table of population statistics for all populations in a GatingHierarchy*

Description

More useful than getPop. Returns a table of population statistics for all populations in a GatingHierarchy. Includes the flowJo counts, flowCore counts and frequencies.

Usage

```
## S4 method for signature 'GatingHierarchy'
getPopStats(x,...)
```

Arguments

x	A GatingHierarchy or GatingSet
...	Additional arguments

Details

Returns a table population statistics for all populations in the gating hierarchy. The output is useful for verifying that the import was successful, if the flowJo and flowCore derived counts don't differ much (i.e. if they have a small coefficient of variation.) for a GatingSet, returns a matrix of proportions for all populations and all samples

Value

A data.frame with columns for the population name, flowJo derived counts, flowCore derived counts, and the population proportions (relative to their parent pouplation).

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

getProp

Examples

```
## Not run:
#If G is a GatingHierarchy
getPopStats(G);

## End(Not run)
```

getProp — *Get the population proportions of a node (population) in a GatingHierarchy*

Description

Calculate the population proportion (events in the gate / events in the parent population) associated with a node in the GatingHierarchy.

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getProp(x, y)
```

Arguments

x	A GatingHierarchy object.
y	character The name of the node. A list of nodes is accessible via getNodes(x).

Details

Returns the proportion of cells in the gate, relative to its parent.

Value

Returns a population frequency numeric.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

getPopStats

Examples

```
## Not run:
#G is a GatingHierarchy
#proportion for the fifth population
getProp(G,getNodes(G)[5])

## End(Not run)
```

getSample *Get the sample name associated with a GatingHierarchy*

Description

Return the sample name

Usage

```
## S4 method for signature 'GatingHierarchy'
getSample(x,isFullPath=FALSE)
## S4 method for signature 'graphNEL'
getSample(x)
```

Arguments

x	A GatingHierarchy or a graphNEL object from the @tree slot of a GatingHierarchy
isFullPath	isFullPath is a logical value indicating whether the full path of the sample FCS file is returned.Default is FALSE.

Details

Returns the name of the sample, or the path to the FCS file.

Value

A "character" vector of length 1. Either the sample name or the path to the FCS file.

Author(s)

Mike Jiang <wjiang2@fhcrc.org>

References

http://www.rglab.org/

Examples

```
## Not run:
#G is  a GatingHierarhcy
getSample(G)
getSample(G@tree);

## End(Not run)
```

getSampleGroups *Get a table of sample groups from a flowJo workspace*

Description

Return a data frame of sample group information from a flowJo workspace

Usage

```
## S4 method for signature 'flowJoWorkspace'
getSampleGroups(x)
```

Arguments

x	A flowJoWorkspace object.

Details

Returns a table of samples and groups defined in the flowJo workspace

Value

A data.frame containing the groupName, groupID, and sampleID for each sample in the workspace. Each sample may be associated with multiple groups.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

flowJoWorkspace-class openWorkspace

Examples

```
## Not run:
#ws is a flowJoWorkspace
getSampleGroups(ws);

## End(Not run)
```

getSamples *Get a list of samples in a flowJo workspace or a GatingSet*

Description

Return a data frame of samples contained in a flowJo workspace or a GatingSet

Usage

```
## S4 method for signature 'GatingSet'
getSamples(x,isFullPath=FALSE)
## S4 method for signature 'flowJoWorkspace'
getSamples(x)
```

Arguments

x	A flowJoWorkspace or a GatingSet
isFullPath	isFullPath is a logical value indicating whether the full path of the sample file is returned.Default is FALSE.

Details

Returns a data.frame of samples in the flowJoWorkspace, including their sampleID, name, and compID (compensation matrix ID). If x is a GatingSet, returns a character vector of sample names.

Value

A data.frame with columns sampleID, name, and compID if x is a flowJoWorkspace. A character vector of sample names if x is a GatingSet.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

Examples

```
## Not run:
#G is  a GatingSet
getSamples(G)
#f is a flowJoWorkspace
getSamples(f);

## End(Not run)
```

getTotal *Fetch the total number of events in a gate defined in a GatingHierarchy*

Description

Returns the total number of events in the gate defined in the GatingHierarchy object

Usage

```
## S4 method for signature 'GatingHierarchy,character'
getTotal(x, y)
```

Arguments

x	The GatingHierarchy
y	A character name of the gate / population

Details

Will return the total number of events included in this gate. The contents of "thisTot" variable in the "metadata" environment of the nodeData element associated with the gating tree and gate / population.

Value

A numeric value of the total number of elements in the population.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

getProp getPopStats

Examples

```
## Not run:
#G is a gating hierarchy
#Fifth node total.
getTotal(G,getNodes(G,tsort=T)[5])

## End(Not run)
```

getTransformations *Return a list of transformations in a flowJo workspace*

Description

Return a list of all the transformations in a flowJo workspace

Usage

```
## S4 method for signature 'flowJoWorkspace'
getTransformations(x)
```

Arguments

x	A flowJoWorkspace object

Details

Returns a list of the transformations in the flowJo workspace. The list is of length L, where L is the number of distinct transformations applied to samples in the flowJoWorkspace. Each element of L is itself a list of length M, where M is the number of parameters that were transformed for a sample or group of samples in a flowJoWorkspace. For example, if a sample has 10 parameters, and 5 are transformed during analysis, using two different sets of transformations, then L will be of length 2, and each element of L will be of length 5. The elements of L represent channel- or parameter-specific transformation functions that map from raw intensity values to channel-space used by flowJo.

Value

comp 1	The first transformaton in the workspace.
comp 2	The second transformation in the workspace.

Comp 1 .. Comp L are themselves lists of functions, with each element of the list representing a transformation applied to a specific channel/parameter of a sample.

Note

This representation will likely be changed in the future to use the flowCore internal transformation classes.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

linknames

Examples

```
## Not run:
#Assume f is a flowJoWorkspace
getTransformations(f);

## End(Not run)
```

---

haveSameGatingHierarchy

Tests whether two GatingHierarchy or GatingSet objects have the same gating structure.

Description

Tests whether GatingHierarchy, or GatingSet objects have the same gating structure.

Usage

```
    ## S4 method for signature 'GatingHierarchy,GatingHierarchy'
haveSameGatingHierarchy(object1, object2)
    ## S4 method for signature 'GatingHierarchy,GatingSet'
haveSameGatingHierarchy(object1, object2)
    ## S4 method for signature 'GatingSet,GatingHierarchy'
haveSameGatingHierarchy(object1, object2)
    ## S4 method for signature 'GatingSet,GatingSet'
haveSameGatingHierarchy(object1, object2)
    ## S4 method for signature 'GatingSet,missing'
haveSameGatingHierarchy(object1, object2)
    ## S4 method for signature 'list,missing'
haveSameGatingHierarchy(object1, object2)
```

Arguments

object1	A GatingHierarchy, GatingSet, or list of GatingHierarchy objects.
object2	A GatingHierarchy, GatingSet, or missing.

Details

Compares the gating tree of the two objects, or of the GatingHierarchies in a single GatingSet.

Value

A logical, TRUE if the gating structures are the same, or FALSE if they differ.

Author(s)

Greg Finak <gfinak@fhcrc.org>

---

includedChannel2ExcludedChannel

Convert a list of parameter names in a GatingSet *to its complement.*

Description

Convert a list of parameter names in a GatingSet to its complement.

Usage

```
includedChannel2ExcludedChannel(gs, includedims = NULL)
```

Arguments

gs	A GatingSet.
includedims	A character vector of channel / parameter names in the gating set.

Value

A character vector of channel names. The complement of includedims.

Author(s)

Greg Finak <gfinak@fhcrc.org>

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
```

---

includedGate2ExcludedGate

Convert a set of gates given by their flowJo gate paths to its complement (and excluding Boolean gates), referenced by gate indices recognized by normalizeGatingSet.

Description

Convert a set of gates given by their flowJo gate paths to its complement (and excluding Boolean gates), referenced by gate indices recognized by normalizeGatingSet.

Usage

```
includedGate2ExcludedGate(gs, includegates)
```

Arguments

gs	A GatingSet
includegates	A character vector of flowJo gating paths for the gates in gs

Value

A numeric vector of gate indices to be used with normalizeGatingSet.

Author(s)

Greg Finak <gfinak@fhcrc.org>

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
```

---

keyword-methods — *Methods to retrieve keywords associated with an FCS sample*

Description

See Methods for function keyword in Package 'flowCore'

Methods

Retrieve a specific keyword for a specific sample in a GatingHierarchy or or set of samples in a GatingSet

Retrieve a keyword for the GatingHierarchy.

signature(object = "GatingHierarchy", keyword = "character")signature(object="GatingSet",keyword="characte

Retrieve a specific keyword for each sample in the GatingSet

See Also

keyword-methods

---

lapply-methods — *Methods for iterating over a gating set*

Description

~~ Methods for function lapply ~~

Methods

signature(X = "ANY")

signature(X = "GatingSet")

length-methods *Methods to get the length of a GatingSet*

Description

Return the length of a GatingSet object (number of samples).

Methods

signature(x = "GatingSet")

openWorkspace *Open a flowJo workspace*

Description

Open a flowJo workspace and return a flowJoWorkspace object.

Usage

```
## S4 method for signature 'character'
openWorkspace(file)
```

Arguments

file	Full path to the XML flowJo workspace file.

Details

Open an XML flowJo workspace file and return a flowJoWorkspace object. The workspace is represented using a XMLInternalDocument object.

Value

Returns a flowJoWorkspace object.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

See Also as xmlTreeParse

Examples

```
## Not run:
file<-"myworkspace.xml"
ws<-openWorkspace(file);
class(ws); #flowJoWorkspace

## End(Not run)
```

---

parseWorkspace *Parse a flowJo Workspace*

Description

Function to parse a flowJo Workspace, generate a GatingHierarchy or GatingSet object, and associated flowCore gates. The data are not loaded or acted upon until an explicit call to execute() is made on the GatingHierarchy objects in the GatingSet.

Usage

```
## S4 method for signature 'flowJoWorkspace'
parseWorkspace(obj,name=NULL,execute=TRUE,isNcdf=FALSE,subset=NULL,nslaves=4,requiregates=TR
```

Arguments

obj	A flowJoWorkspace to be parsed.
name	numeric or character. The name or index of the group of samples to be imported. If NULL, the groups are printed to the screen and one can be selected interactively. Usually, multiple groups are defined in the flowJo workspace file.
execute	TRUE|FALSE a logical specifying if the gates, transformations, and compensation should be immediately calculated after the flowJo workspace have been imported. TRUE by default.
isNcdf	TRUE|FALSE logical specifying if you would like to use netcdf to store the data, or if you would like to keep all the flowFrames in memory. For a small data set, you can safely set this to FALSE, but for larger data, we suggest using netcdf. You will need the netcdf C library installed.
subset	numeric vector specifying the subset of samples in a group to import.
nslaves	numeric number of slave processes for executing the gating under Rmpi
requiregates	logical Should samples that have no gates be included?
includeGates	logical Should gates be imported, or just the data with compensation and transformation?
...	Additional arguments. path="character" The path to the fcs files that are to be imported. The code will search recursively, so you can point it to a location above the files. This argument is mandatory.

Details

A flowJoWorkspace is generated with a call to openWorkspace(), passing the name of the xml workspace file. This returns a flowJoWorkspace, which can be parsed using the parseWorkspace() method. The function can be called non-interactively by passing the index or name of the group of samples to be imported via parseWorkspace(obj,name=x), where x is either the numeric index, or the name.

Value

Returns a GatingSet, which is a wrapper around a list of GatingHierarchy objects, each representing a single sample in the workspace. The GatingHierarchy objects contain graphNEL trees that represent the gating hierarchy of each sample. Each node in the GatingHierarchy has associated data, including the population counts from flowJo, the parent population counts, the flowCore gates generated from the flowJo workspace gate definitions. Data are not yet loaded or acted upon at this stage. To execute the gating of each data file, a call to execute() must be made on each GatingHierarchy object in the GatingSet. This is done automatically by default, and there is no more reason to set this argument to FALSE.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

getSampleGroups

Examples

```
## Not run:
#f is a xml file name of a flowJo workspace
ws<-openWorkspace(f)
G<-parseWorkspace(ws,execute=TRUE,isNcdf=FALSE,path="."); #assume that the fcs files are below the current dire
#G is a GatingSet.

## End(Not run)
```

---

plot *Plot a GatingHierarchy*

Description

Plot a tree representing the GatingHierarchy

Usage

```
## S4 method for signature 'GatingHierarchy,missing'
plot(x,y,layout="dot",width=3,height=2,fontsize=14,labelfontsize=14,fixedsize=FALSE,boolean=FALSE,...)
```

Arguments

x	The GatingHierarchy to be plotted
y	missing.
layout	See layoutGraph in package Rgraphviz
width	See layoutGraph in package Rgraphviz
height	See layoutGraph in package Rgraphviz

fontsize	See layoutGraph in package Rgraphviz
labelfontsize	See layoutGraph in package Rgraphviz
fixedsize	See layoutGraph in package Rgraphviz
boolean	TRUE|FALSE logical specifying whether to plot boolean gate nodes. Defaults to FALSE.
...	Additional arguments passed to plot in Rgraphviz

Details

Plot a GatingHierarchy object using the Rgraphviz plot function.

Value

Nothing to return

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

See Also

layoutGraph

Examples

```
## Not run:
#G is a GatingHierarchy
plot(G);

## End(Not run)
```

---

plotGate *Plot a flowJo Gate and Cell Population*

Description

Plots a flowJo gate and associated cell population using it's flowCore definition contained in a GatingHierarchy

Usage

```
## S4 method for signature 'GatingHierarchy,character'
plotGate(x, y, add=FALSE,border="red",tsort=FALSE,smooth=FALSE,fast=FALSE,...)
```

Arguments

x	A GatingHierarchy
y	A character or numeric representing the node in the GatingHierarchy. Nodes can be accessed with getNodes(GatingHierarchy).
add	TRUE|FALSE logical specifying whether to add the gate to the current plot.
border	character, The color to plot the border of the gate. Default is "red".
tsort	TRUE|FALSE logical indicating if nodes should be referenced in topological sort order when y="numeric";
smooth	TRUE|FALSE logical indicating whether a smoothed 2D scatterplot should be drawn;
fast	TRUE|FALSE logical indicating whether or not to use hexbin plotting routines, which are a bit faster.
...	Additional arguments to the plot function.

Details

The function will plot the gate if the gating hierarchy represented by x has been execute()'d. That is to say, the associated data has been loaded, compensated, transformed, and had the gates applied to it. If the data has not been gated, plotGate will print a message, and return without plotting anything.

Value

If the data has been gated, the function will plot the gate. If it has not been gated, the function will print a message and return nothing.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org/

Examples

```
## Not run:
#G is a GatingHierarchy
plotGate(G,getNodes(G)[5]);#plot the gate for the  fifth node

## End(Not run)
```

plotPopCV *Plot the coefficient of variation between flowJo and flowCore population statistics for each population in a gating hierarchy.*

Description

This function plots the coefficient of variation calculated between the flowJo population statistics and the flowCore population statistics for each population in a gating hierarchy extracted from a flowJoWorkspace.

Usage

```
## S4 method for signature 'GatingHierarchy'
plotPopCV(x,m=2,n=2,...)
## S4 method for signature 'GatingSet'
plotPopCV(x,...)
```

Arguments

x	A GatingHierarchy from a flowJoWorkspace, or a GatingSet.
m	numeric The number of rows in the panel plot. Now deprecated, uses lattice.
n	numeric The number of columns in the panel plot. Now deprecated, uses lattice.
...	Additional arguments to the barplot methods.

Details

The CVs are plotted as barplots across panels on a grid of size m by n.

Value

Nothing is returned.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

http://www.rglab.org

See Also

getPopStats

Examples

```
## Not run:
#G is a GatingHierarchy
plotPopCV(G,4,4);

## End(Not run)
```

plotWf — *plot a workflow*

Description

Plots a workflow that contains one GatingHierarchy

Usage

```
plotWf(x,...)
```

Arguments

x	A workflow
...	Additional arguments to the plot function.

Author(s)

Mike Jiang

Examples

```
## Not run: dataDir <- system.file("extdata", package = "flowWorkspace")
wsfile<-list.files(dataDir,pattern="xml",full=TRUE)[1]

##open workspace xml file
ws<-openWorkspace(wsfile)

##Convert a flowWorkspace to workFlows
wfs<-flowWorkspace2flowCore(ws,groupId=1,path=dataDir)
plotWf(wfs[[1]])
## End(Not run)
```

recomputeGatingSet — *Recalculates all the gating in the GatingSet if netCDF storage is used.*

Description

Re–runs all the gating in a GatingSet object if netCDF storage is used.

Usage

```
recomputeGatingSet(x)
```

Arguments

x	A GatingSet object.

Details

This function wil recalculate all the gates in a gating set object. Useful if you've run some normalization and need to recalcualte the gates based on the adjusted data. Only if netCDF storage is used.

Value

Returns NULL. The gates are written directly to the netCDF file and reflected in the parent frame.

Author(s)

Greg Finak <gfinak@fhcrc.org>

---

saveNcdf — *Save the netcdf file of a GatingSet or GatingHierarchy to a permanent location.*

Description

When creating a new GatingSet or GatingHierarchy using netcdf backed storage, the nc files are written to the TEMP directory. This will move those files to a more permanent location, allowing the object to be saved and persist after restarting R and relaoding the workspace.

Usage

```
saveNcdf(objName, path = getwd())
```

Arguments

objName	The name of the GatingSet or GatingHierarchy. A character vector.
path	A character vector: the path where the ncdf files should be saved.

Details

The function will modify the GatingSet or GatingHierarchy to reflect the new location of the ncdf files. Do not pass the object directly via objName, but rather pass the name of the variable holding the GatingSet or GatingHierarchy as a character vector.

Value

Does not return anything. Modifes the object in the parent environment.

Author(s)

Greg Finak <gfinak@fhcrc.org>

---

summary-methods — *Summarize a flowJoWorkspace object*

Description

Summarize a flowJoWorkspace object.

Index

∗Topic **IO**
closeWorkspace, 3
∗Topic **classes**
flowJoWorkspace-class, 9
GatingHierarchy-class, 11
GatingSet-class, 14
∗Topic **hplot**
plot, 40
plotGate, 41
plotPopCV, 43
plotWf, 44
∗Topic **manip**
ellipsoidGate2FlowJoVertices, 4
execute, 5
exportAsFlowJoXML, 6
ExportTSVAnalysis, 8
flowWorkspace2flowCore, 10
getBoundaries, 15
getChildren, 16
getCompensationMatrices, 17
getData, 18
getDimensions, 19
getFJWSubsetIndices, 20
getGate, 21
getIndiceFile, 22
getIndices, 23
getNodes, 26
getParent, 27
getPopStats, 28
getProp, 29
getSample, 30
getSampleGroups, 31
getSamples, 32
getTotal, 33
getTransformations, 34
haveSameGatingHierarchy, 35
includedChannel2ExcludedChannel, 36
includedGate2ExcludedGate, 36
keyword-methods, 37
openWorkspace, 38
parseWorkspace, 39
saveNcdf, 45
∗Topic **methods**
keyword-methods, 37
lapply-methods, 37
length-methods, 38
summary-methods, 45
∗Topic **package**
flowWorkspace-package, 2
∗Topic **univar**
plotPopCV, 43
∗Topic **utilities**
flowWorkspace-package, 2
.includedChannel2ExcludedChannel *(includedChannel2ExcludedChannel)*, 36
.includedGate2ExcludedGate *(includedGate2ExcludedGate)*, 36
[,GatingSet,ANY-method *(GatingSet-class)*, 14
[[,GatingSet,ANY-method *(GatingSet-class)*, 14

AnnotatedDataFrame, *15*

closeWorkspace, 3
closeWorkspace,flowJoWorkspace-method *(closeWorkspace)*, 3
closeWorkspace-methods *(closeWorkspace)*, 3

ellipse, *5*
ellipsoidGate2FlowJoVertices, 4
ellipsoidGate2FlowJoVertices,ellipsoidGate-method *(ellipsoidGate2FlowJoVertices)*, 4
ellipsoidGate2FlowJoVertices-methods *(ellipsoidGate2FlowJoVertices)*, 4
execute, 5
execute,GatingHierarchy-method *(execute)*, 5
execute-methods *(execute)*, 5
exportAsFlowJoXML, 6
exportAsFlowJoXML,ellipsoidGate-method *(exportAsFlowJoXML)*, 6
exportAsFlowJoXML,intersectFilter-method *(exportAsFlowJoXML)*, 6

exportAsFlowJoXML,list-method (exportAsFlowJoXML), 6
exportAsFlowJoXML,polygonGate-method (exportAsFlowJoXML), 6
exportAsFlowJoXML,rectangleGate-method (exportAsFlowJoXML), 6
exportAsFlowJoXML,workFlow-method (exportAsFlowJoXML), 6
exportAsFlowJoXML-methods (exportAsFlowJoXML), 6
ExportTSVAnalysis, 8

flowJoWorkspace, *15*
flowJoWorkspace-class, 9
flowWorkspace (flowWorkspace-package), 2
flowWorkspace-package, 2
flowWorkspace2flowCore, 10
flowWorkspace2flowCore,flowJoWorkspace-method (flowWorkspace2flowCore), 10
flowWorkspace2flowCore,GatingHierarchy-method (flowWorkspace2flowCore), 10
flowWorkspace2flowCore,GatingSet-method (flowWorkspace2flowCore), 10
flowWorkspace2flowCore-methods (flowWorkspace2flowCore), 10

GatingHierarchy, *10*, *15*
GatingHierarchy-class, 11
GatingSet, *10*, *13*
GatingSet-class, 14
getBoundaries, 15
getBoundaries,GatingHierarchy,character-method (getBoundaries), 15
getBoundaries-methods (getBoundaries), 15
getChildren, 16, *27*
getChildren,GatingHierarchy,character-method (getChildren), 16
getChildren-methods (getChildren), 16
getCompensationMatrices, 17
getCompensationMatrices,flowJoWorkspace-method (getCompensationMatrices), 17
getCompensationMatrices-methods (getCompensationMatrices), 17
getData, 18, *22*, *25*
getData,GatingHierarchy-method (getData), 18
getData,GatingSet-method (getData), 18
getData,graphNEL-method (getData), 18
getData-methods (getData), 18
getDimensions, 19
getDimensions,GatingHierarchy,character-method (getDimensions), 19
getDimensions-methods (getDimensions), 19
getFJWSubsetIndices, 20
getGate, *16*, 21
getGate,GatingHierarchy,character-method (getGate), 21
getGate,GatingHierarchy,numeric-method (getGate), 21
getGate,GatingSet,numeric-method (getGate), 21
getIndiceFile, 22
getIndiceFile,GatingHierarchy-method (getIndiceFile), 22
getIndiceFile-methods (getIndiceFile), 22
getIndices, *19*, 23
getIndices,GatingHierarchy,character-method (getIndices), 23
getIndices-methods (getIndices), 23
getKeywords, 24
getKeywords,flowJoWorkspace,character-method (getKeywords), 24
getKeywords,GatingHierarchy,missing-method (getKeywords), 24
getKeywords,GatingSet,character-method (getKeywords), 24
getKeywords,GatingSet,numeric-method (getKeywords), 24
getKeywords-methods (getKeywords), 24
getNcdf, 25
getNcdf,GatingHierarchy-method (getNcdf), 25
getNcdf,GatingSet-method (getNcdf), 25
getNodes, *20*, *22*, 26
getNodes,GatingHierarchy-method (getNodes), 26
getNodes-methods (getNodes), 26
getParent, 27
getParent,GatingHierarchy,character-method (getParent), 27
getParent,GatingHierarchy,numeric-method (getParent), 27
getParent-methods (getParent), 27
getPopStats, *19*, *23*, 28, *29*, *33*, *43*
getPopStats,GatingHierarchy-method (getPopStats), 28
getPopStats,GatingSet-method (getPopStats), 28
getPopStats-methods (getPopStats), 28
getProp, *19*, *23*, *28*, 29, *33*
getProp,GatingHierarchy,character-method (getProp), 29
getProp-methods (getProp), 29

getSample, 30
getSample,GatingHierarchy-method (getSample), 30
getSample,graphNEL-method (getSample), 30
getSample-method (getSample), 30
getSampleGroups, 31, *40*
getSampleGroups,flowJoWorkspace-method (getSampleGroups), 31
getSampleGroups-methods (getSampleGroups), 31
getSamples, 32
getSamples,flowJoWorkspace-method (getSamples), 32
getSamples,GatingSet-method (getSamples), 32
getSamples-methods (getSamples), 32
getTotal, 33
getTotal,GatingHierarchy,character-method (getTotal), 33
getTotal-methods (getTotal), 33
getTransformations, 34
getTransformations,flowJoWorkspace-method (getTransformations), 34
getTransformations-methods (getTransformations), 34

haveSameGatingHierarchy, 35
haveSameGatingHierarchy,GatingHierarchy,GatingHierarchy-method (haveSameGatingHierarchy), 35
haveSameGatingHierarchy,GatingHierarchy,GatingSet-method (haveSameGatingHierarchy), 35
haveSameGatingHierarchy,GatingSet,GatingHierarchy-method (haveSameGatingHierarchy), 35
haveSameGatingHierarchy,GatingSet,GatingSet-method (haveSameGatingHierarchy), 35
haveSameGatingHierarchy,GatingSet,missing-method (haveSameGatingHierarchy), 35
haveSameGatingHierarchy,list,missing-method (haveSameGatingHierarchy), 35
haveSameGatingHierarchy-methods (haveSameGatingHierarchy), 35

includedChannel2ExcludedChannel, 36
includedGate2ExcludedGate, 36

keyword (keyword-methods), 37
keyword,GatingHierarchy,character (keyword-methods), 37
keyword,GatingHierarchy,character-method (keyword-methods), 37
keyword,GatingSet,character (keyword-methods), 37
keyword,GatingSet,character-method (keyword-methods), 37
keyword-methods, 37

lapply,ANY-method (lapply-methods), 37
lapply,GatingSet-method (lapply-methods), 37
lapply-methods, 37
layoutGraph, *40, 41*
length,GatingSet-method (length-methods), 38
length-methods, 38

normalizeGatingSet, *36, 37*

openWorkspace, *17*, *31*, 38
openWorkspace,character-method (openWorkspace), 38
openWorkspace-methods (openWorkspace), 38

parseWorkspace, *6*, *11*, *13*, *21*, 39
parseWorkspace,flowJoWorkspace-method (parseWorkspace), 39
parseWorkspace-methods (parseWorkspace), 39
plot, 40
plot,GatingHierarchy,missing-method (plot), 40
plot-methods (plot), 40
plotGate, 41
plotGate,GatingHierarchy,character-method (plotGate), 41
plotGate,GatingHierarchy,numeric-method (plotGate), 41
plotGate-methods (plotGate), 41
plotPopCV, 43
plotPopCV,GatingHierarchy-method (plotPopCV), 43
plotPopCV,GatingSet-method (plotPopCV), 43
plotPopCV-methods (plotPopCV), 43
plotWf, 44

recomputeGatingSet, 44

saveNcdf, 45
show,flowJoWorkspace-method (summary-methods), 45
show,GatingHierarchy-method (summary-methods), 45
show,GatingSet-method (summary-methods), 45
show-methods (summary-methods), 45

summary,ANY-method
(summary-methods), 45
summary,flowJoWorkspace-method
(summary-methods), 45
summary-methods, 45

tsort, *26*

xmlTreeParse, *4*, *38*