Package 'SureTypeSCR'

May 29, 2021

Title Library for (single-cell) SNP array processing

Description
SureTypeSCR is R package for QC of (single cell) SNP arrays and single cell genotype scoring

Version 0.99.0

Author Ivan Vogel, Lishan Cai

Suggests testthat,rmarkdown,markdown,knitr

Depends R (>= 4.0.0), reticulate, tidyverse, magrittr

Maintainer Ivan Vogel <ivogel@sund.ku.dk>

License GPL-3

Imports ggrepel, RColorBrewer, BiocStyle

biocViews Software, GenotypingArray,SingleCell

VignetteBuilder knitr

SystemRequirements python (>= 3.6), sklearn, numpy, pandas,
SureTypeSC, IlluminaBeadArrayFiles

NeedsCompilation no

R topics documented:

`calculate_ma` *Calculate M (logarithmic difference) and A (logarithmic average) of allelic intensities*

Description

Function applies Logartihmic transformation on signal intensities which is a preliminary step for classification using SureTypeSC.

Usage

```
calculate_ma(df)
```

Arguments

`df`	data frame from `scbasic(.)` or compatible

Value

`df` with extra columns representing the results of MA transformation (m, a, m_raw and a_raw)

Examples

```
setwd(system.file(package='SureTypeSCR'))
samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')

#Load data
df=scbasic(manifest,cluster,samplesheet)

#calculate MA transform and store in the original dataframe
#df %<>% calculate_ma()
```

`callrate` *Calculates call rate as proportion of called SNPs in the input dataframe*

Description

Calculate call rate as proportion of called SNPs in the input dataframe.

Usage

```
callrate(.data)
```

Arguments

.data	data frame from scbasic or compatible

Value

Table with call rate(s) of the input data frame.

Examples

```
setwd(system.file(package='SureTypeSCR'))

samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')

##Load data
df=scbasic(manifest,cluster,samplesheet)

##Get overall callrate
#df %>% callrate()

##Get callrate per individual
#df %>%
# group_by(individual) %>%
# callrate()

##Get callrate as allelic fractions and pivot to columns
#df %>%
# group_by(individual,gtype) %>%
# callrate() %>%
# pivot_wider(names_from=gtype,values_from=Callrate)
```

---

configure_iaap *Procedure for setting path to IAAP-cli executable*

Description

This procedure first checks whether the package home folder contains IAAP-cli executable. If not, then it guides the user to download the IAAP-cli archive and configure path to the archive. The procedure then decompresses the archive and sets up the path.

Usage

```
configure_iaap()
```

Value

returns path to IAAP-cli executable

`create_dataobject_from_frame` *Internal function for converting basic data frame from SureTypeSCR to SuretypeSC's python dataobject*

Description

The function returns SureTypeSC's dataobject that contains multindexed pandas data frame and various metadata. This structre is used in some of the SureTypeSC's methods that are called from SureTypeSCR.

Usage

```
create_dataobject_from_frame(df_single)
```

Arguments

`df_single`	data frame used in SureTypeSCR

Value

SuretypeSC's data object (instance of class Data) containing multiindexed data frame and metadata.

`create_from_frame` *Internal function for converting between python structures*

Description

This functions is used internally in `scbasic` and converts python data frame in to SureTypeSC's data object.

Usage

```
create_from_frame(df)
```

Arguments

`df`	genotyping pandas dataframe from scbasic function

Value

Instance of class Data from Python SureTypeSC library with multiindex.

getGEO_and_folder_in *Download item from the GEO database*

Description

Downloads metadata (and data) for GEO records specified in x.

Usage

```
getGEO_and_folder_in(x,download=TRUE)
```

Arguments

x	GEO record ID
download	boolean flag indicating whether to download the data (TRUE) or just the metadata (FALSE)

Value

Data frame with metadata

Examples

```
#library(GEOquery)

#gse <- getGEO("GSE19247",GSEMatrix = TRUE)

#samplelist_sperm=as.data.frame(gse$`GSE19247-GPL6985_series_matrix.txt.gz`) %>%
# filter((str_detect(cell.type.ch1,'sperm')) & str_detect(cell.amplication.ch1,'MDA')) %>%
#  rownames()

#metadf_sperm=map_if(samplelist_sperm,function(x) !is.null(x),
#                    function(x) getGEO_and_folder_in(x,download=TRUE))
#metadf_sperm_merged = Reduce(function(...) merge(..., all=T), metadf_sperm)
```

get_multiind_df *Internal function that creates data frame with multiindex*

Description

Internal function for index conversion. Function converts R data frame to a Python pandas dataframe with multiindex, that is compatible with Python methods implemented in SureTypeSC.

Usage

```
get_multiind_df(df)
```

Arguments

df	R data frame as returned by function scbasic(.) or compatible

Value

returns data frame with multiindex

`get_simpleind_df` *Internal function that creates data frame with simple column index*

Description

Internal function for index conversion. Function converts data frame's column multiindex into a simple form.

Usage

```
get_simpleind_df(df)
```

Arguments

`df`	data frame with column multiindex

Value

returns data frame with simple index

`get_threshold` *Determine threshold used on the data in data frame*

Description

Determine score cutoff that was applied on the data - that is all genotypes below the returned value are no-calls.

Usage

```
get_threshold(.data,col='score')
```

Arguments

`.data`	basic data frame compatible with output from `scbasic(.)`
`col`	column name with score - effective columns GenCall's score (`scode`), SureTypeSC's score (`rfgda_score`) or single layer SureTypeSC's score (`rf_score`)

Value

returns float

idat_to_gtc *Convert raw itensity IDAT files to GTC using IAAP-cli*

Description

The raw itensity data (red and green channel) are processed into GTC using Illumina's IAAP-cli. Function runs `configure_iaap()` before instantiating IAAP-cli to make sure IAAP-cli is properly configured.

Usage

```
idat_to_gtc(idat_inputfolder,gtc_output,manifest,cluster)
```

Arguments

idat_inputfolder	input folder with stored idat files
gtc_output	output folder for the GTC files
manifest	path to manifest file
cluster	path to cluster file

Examples

```
#library(GEOquery)
#gse <- getGEO("GSE19247",GSEMatrix = TRUE)

#samplelist_sperm=as.data.frame(gse$`GSE19247-GPL6985_series_matrix.txt.gz`) %>%
# filter((str_detect(cell.type.ch1,'sperm')) & str_detect(cell.amplication.ch1,'MDA')) %>%
#  rownames()

#metadf_sperm=map_if(samplelist_sperm,function(x) !is.null(x),
#                    function(x) getGEO_and_folder_in(x,download=TRUE))
#metadf_sperm_merged = Reduce(function(...) merge(..., all=T), metadf_sperm)

#manifest=system.file("files/HumanCytoSNP-12v2_H.bpm",package="SureTypeSCR")
#cluster=system.file("files/HumanCytoSNP-12v2_H.egt",package="SureTypeSCR")

#for (ar in unique(metadf_sperm_merged$arrayid))
#{
#  idat_to_gtc(ar,'GTC',manifest,cluster)
#}
```

plot_ma *Create MA plot per individual.*

Description

Plot XY plot of logarithmic difference (M) and logarithmic average (A) of intensity signals-

Usage

```
plot_ma(.data,norm=TRUE,smooth=FALSE,nocalls=FALSE,n=1)
```

Arguments

.data	data frame returned by scbasic or a compatible function
norm	boolean flag indicating whether to use normalized (TRUE) or raw (FALSE) intensities
smooth	boolean flag indicating whether to apply smoothing spline per genotype (AA/BB/AB) cluster (TRUE or FALSE, set to FALSE by default.
nocalls	boolean flag indicating whether to plot SNPs that did not call (TRUE or FALSE, set to FALSE by default.
n	fraction of data to use from .data (real number in range 0..1), set to 1 by default meaning all points will by plotted.

Value

ggplot object with MA plots facetted by individual

Examples

```
setwd(system.file(package='SureTypeSCR'))

samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')

#Load data
df=scbasic(manifest,cluster,samplesheet)

#Visualise MA plot
#df %>% plot_ma(norm=TRUE,n=0.1)
```

---

plot_pca *Run and visualise PCA*

Description

Function creates feature matrix from non-zero SNPs and feature types indicated in features. The feature matrix used for calculating PCA has n rows and k columns, while n corresponds to number of individuals in the data frame and k is defined as length of features x number of non-zero SNPs across all samples.

Usage

```
plot_pca(.data,by_chrom=TRUE,features=c('x','y'),importances=TRUE,metadata=NULL,labels=TRUE)
```

Arguments

.data	dataframe from scbasic or compatible
by_chrom	boolean flag indicating whether to run PCA by chromosome (TRUE or FALSE)
features	vector of feature types to use for PCA (columns from .data , i.e. c('x','y'))
importances	boolean flag indicating whether to render dimension importances (TRUE or FALSE)
metadata	data frame with metadata - currently grouping by family is supported, given that metadata contains familyid and individual columns and metadata will be merged to internal data frame by matching individual
labels	boolean flag indicating whether to render sample names using package ggrepel to prevent overplotting (TRUE or FALSE)

Value

ggplot object with PCA plot

Examples

```
setwd(system.file(package='SureTypeSCR'))

samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')

#Load data
df=scbasic(manifest,cluster,samplesheet)

#create ggplot object with PCA
#df %>% plot_pca(by_chrom=TRUE,features=c('x','y'),labels=TRUE)
```

---

predict_suretype *Run classification model using combination of Random Forest and Gaussian Discriminant Analysis*

Description

Cascade classification model with Random Forest in the first layer and Gaussian Discriminant Analysis in the second layer. This function analyzes the whole dataset comprised in .data in one batch. To minimize bias in the model, it is recommended to run suretype_model(.) and build a model per sample basic instead of the whole dataset.

Usage

```
predict_suretype(.data, rf_clf)
```

Arguments

.data	Input dataframe coming from scbasic() or compatible
rf_clf	Instance of classifier loaded using scload currently embodied in RF

Value

data frame decorated with SureTypeSC genotyping score

Examples

```
setwd(system.file(package='SureTypeSCR'))
samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')
clf=system.file('files/rf.clf',package='SureTypeSCR')


#Load data
df=scbasic(manifest,cluster,samplesheet)

# The Random Forest classifier
clf_instance <- scload(clf)

#assign prediction results back to the original dataframe using margittr %<>% operator
#df %<>% predict_suretype(clf_instance)
```

---

preprocess_pca *Internal function that calculates matrix for principal component analysis*

Description

Internal function that preprocesses feature matrix for PCA. Output of this function is used in `plot_pca(.)`

Usage

```
preprocess_pca(.data,.group,features=c('x','y'))
```

Arguments

`.data`	output data frame from `scbasic(.)` or compatible
`.group`	formal parameter effective when run under `group_by(.)` and `nest(.)` from `tidyverse`
`features`	list of columns from the `.data` data frame that will be used for creation of the feature matrix

Value

matrix formatted for PCA analysis using `stats::prcomp(.)`

scbasic *Load data in GTC format into data frame*

Description

Function instantiates Illumina BeadArray library to load data from GTC files into data frame using information from manifest and cluster file.

Usage

```
scbasic(bpm,egt,samplesheet)
```

Arguments

bpm	pathname to manifest file
egt	pathname to cluster file
samplesheet	pathname to samplesheet

Value

data frame stacked by individual

Examples

```
setwd(system.file(package='SureTypeSCR'))
samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')

#Load data
df=scbasic(manifest,cluster,samplesheet)
```

scEls *mediate access to python modules*

Description

mediate access to python modules

Usage

```
scEls()
```

Value

list of (S3) "python.builtin.module"

Note

Returns a list containing objects sc and pd that refer to Python modules SureTypeSC and pandas, respectively.

Examples

```
els = scEls()
els
##$sc
##Module(SureTypeSC)

##$pd
##Module(pandas)
```

`scload` *Load classifier from file*

Description

Load pre-trained classifier from file.

Usage

```
scload(filename)
```

Arguments

`filename`	path to a file with classifier

Value

instance of a classifier

Examples

```
setwd(system.file(package='SureTypeSCR'))

samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')
clf=system.file('files/rf.clf',package='SureTypeSCR')


#Load data
df=scbasic(manifest,cluster,samplesheet)

rf=scload(clf)
#df %>% calculate_ma() %>% predict_suretype(rf)
```

scpredict *Internal function used for prediction with* suretype_model(.)

Description

Internal function that evaluates score given an instance of classifier. The function is used in suretype_model(.)

Usage

```
scpredict(.data,clf,clftype='rf')
```

Arguments

.data	data frame containing m and a features
clf	instance of a classifier
clftype	The type of classifier

Value

data frame with predicted score using given classifier in clf

scTrain *Internal function that fits parameters of GDA to given data*

Description

Internal function that creates GDA classifier based on the results of first layer (RF).

Usage

```
scTrain(trainingdata,clfname='gda')
```

Arguments

trainingdata	intermediate results of RF classification
clfname	name of the classifier to be fitted parameters for, currently 'Gaussian discriminant analysis' (GDA) is supported and change of this parameter will only have effect on column name in the resulting data frame.

Value

instance of a GDA classifier

set_threshold *Set threshold on a classification method.*

Description

Function truncates the genotypes (column gtype) given a threshold (cutoff) an classification method. This causes that all genotypes below threshold defined in threshol will be called as NC (no-call)

Usage

```
set_threshold(.data,clfcol,threshold)
```

Arguments

.data	dataframe from scbasic or compatible
clfcol	column with classification score
threshold	threshold from 0 to 1

Value

data frame with gtype set to NC for classification score below chosen threshold

Examples

```
setwd(system.file(package='SureTypeSCR'))
samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')

##Load data
#df=scbasic(manifest,cluster,samplesheet)
##changing Gencall score threshold
#df %>% set_threshold(clfcol='score',threshold=0.5) %>% callrate()
##with higher threshold we expect lower call rate
#df %>% set_threshold(clfcol='score',threshold=0.75) %>% callrate()
```

suretype_model *Create model and perform classification with RF-GDA in one individual*

Description

Function is almost always used with in connection with group_by(invidiual) and nest() to create a classification model per group (typically one individual)

Usage

```
suretype_model(df,individual, clf,.sclist=NULL)))
```

Arguments

df	Basic dataframe with precalculated M and A features
individual	sample name to be classified
clf	path to the classifier - currently support for Random Forest
.sclist	samples to be processed stored in list(), default is NULLmeaning all samples in df will be classified

Value

returns data frame with two columns corresponding to the classification results: rfgda_score and rf_score, while rfgda_score is score calculated by cascade RF-GDA algorithm and rf_score is score calculated by RF.

Examples

```
setwd(system.file(package='SureTypeSCR'))
samplesheet=system.file('files/GSE19247_example.csv',package='SureTypeSCR')
manifest=system.file('files/HumanCytoSNP-12v2_H.bpm',package='SureTypeSCR')
cluster=system.file('files/HumanCytoSNP-12v2_H.egt',package='SureTypeSCR')
clf=system.file("files/rf.clf",package="SureTypeSCR")

#df_model = df %>%
#calculate_ma() %>%
#group_by(individual) %>%
#nest() %>%
#mutate(model=map(data , function(df) suretype_model(df,individual, clf,.sclist=list('gsm477563')))
```

---

write_samplesheet *Function creates samplesheet compatible with SureTypeSCR*

Description

Function is used in case data was converted from idat and sample sheet was not available with the data.

Usage

```
write_samplesheet(filename,array_positions,gtc_output,manifest,experiment_name='Experiment1',\\{
project_name='Project1',investigator_name='Investigator1')
```

Arguments

filename	name/path of the newly created sample sheet
array_positions	data frame storing metadata about sample id, array id and position, following columns are required: sampleid,arrayid,position
gtc_output	path where GTC files are stored
manifest	name of the manifest file
experiment_name	experiment name

`project_name` project name

`investigator_name`
investigator name

Value

writes external file to `filename`

Examples

```
#library(GEOquery)

#gse <- getGEO("GSE19247",GSEMatrix = TRUE)
#manifest=system.file("files/HumanCytoSNP-12v2_H.bpm",package="SureTypeSCR")

#samplelist_sperm=as.data.frame(gse$`GSE19247-GPL6985_series_matrix.txt.gz`) %>%
# filter((str_detect(cell.type.ch1,'sperm')) & str_detect(cell.amplication.ch1,'MDA')) %>%
#  rownames()

#metadf_sperm=map_if(samplelist_sperm,function(x) !is.null(x),
#                    function(x) getGEO_and_folder_in(x,download=TRUE))
#metadf_sperm_merged = Reduce(function(...) merge(..., all=T), metadf_sperm)

#write_samplesheet('samplesheet.csv',metadf_sperm_merged,'GTC',manifest)
```

Index