

Package ‘SureTypeSCR’

November 15, 2019

Title Interface to python based package SureTypeSC via reticulate

Description SureTypeSCR is the implementation of algorithm for regenotyping of single cell data coming from Illumina BeadArrays without genome studio.

Version 0.99.0

Author Ivan Vogel, Lishan Cai

Suggests testthat

Depends R (>= 3.5.0), reticulate, knitr, BiocStyle

Maintainer Lishan Cai <Lishan@sund.dk.com>

License Artistic-2.0

biocViews Software, GenotypingArray,SingleCell

VignetteBuilder knitr

SystemRequirements python (>= 2.7), sklearn, numpy, pandas,
SureTypeSC, IlluminaBeadArrayFiles

NeedsCompilation no

R topics documented:

| | |
|-----------------------------|-----------|
| allele_freq | 2 |
| apply_thresh | 2 |
| calculate_ma | 3 |
| callrate | 4 |
| callrate_chr | 5 |
| create_from_frame | 6 |
| locus_cluster | 6 |
| locus_ma | 7 |
| pca_chr | 8 |
| pca_samples | 9 |
| restrict_chrom | 9 |
| sample_ma | 10 |
| scbasic | 11 |
| scEls | 12 |
| scload | 13 |
| scpredict | 13 |
| scsave | 15 |
| scTrain | 16 |
| Index | 19 |

| | |
|-------------|---|
| allele_freq | <i>The frequency function is to calculate the allele frequency over all the samples</i> |
|-------------|---|

Description

The frequency function is to calculate the allele frequency over all the samples

Usage

```
allele_freq(df, th=0)
```

Arguments

| | |
|----|--|
| df | the pandas dataframe from GenomeStudio or scbasic function |
| th | the threshold |

Value

The frequency values of all samples

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs", package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt', package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm', package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv', package='SureTypeSCR')

df <- scbasic(manifest_path, cluster_path, samplesheet)

# The Random Forest classifier
call <- allele_freq(df, th=0.2)
```

| | |
|--------------|---|
| apply_thresh | <i>To apply threshold over all samples on GenCall score</i> |
|--------------|---|

Description

To apply threshold over all samples on GenCall score

Usage

```
apply_thresh(df, th)
```

Arguments

df the Data object from create_from_frame function
th the threshold

Value

Data object only with applied threshold

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)

df <- create_from_frame(df)

df <- apply_thresh(df,0.01)
```

| | |
|--------------|--------------------------------------|
| calculate_ma | <i>To calculate m and a features</i> |
|--------------|--------------------------------------|

Description

To calculate m and a features

Usage

```
calculate_ma(df)
```

Arguments

df the Data object from create_from_frame function

Value

Data object with adding m and a features

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)

df <- create_from_frame(df)

dfs <- calculate_ma(df)
```

| | |
|----------|--|
| callrate | <i>The callrate function is to calculate the allele frequency over all the samples</i> |
|----------|--|

Description

The callrate function is to calculate the allele frequency over all the samples

Usage

```
callrate(df, th=0)
```

Arguments

| | |
|----|--|
| df | the pandas dataframe from GenomeStudio or scbasic function |
| th | the threshold |

Value

The callrate values of all samples

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)

# The Random Forest classifier
```

```
call <- callrate(df,th=0.2)
```

| | |
|--------------|---|
| callrate_chr | <i>The callrate function is to calculate the allele frequency over all the samples of one specific chromosome</i> |
|--------------|---|

Description

The callrate function is to calculate the allele frequency over all the samples of one specific chromosome

Usage

```
callrate_chr(df,chr_name,th=0)
```

Arguments

| | |
|----------|--|
| df | the pandas dataframe from GenomeStudio or scbasic function |
| chr_name | the name of the selected chromosome |
| th | the threshold |

Value

The callrate values of all samples of one specific chromosome

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)

# The Random Forest classifier
call <- callrate_chr(df,'1',th=0.2)
```

| | |
|-------------------|--|
| create_from_frame | <i>convert pandas dataframe to Data object</i> |
|-------------------|--|

Description

Convert pandas dataframe to Data object and rearrange the index level

Usage

```
create_from_frame(df)
```

Arguments

df genotyping pandas dataframe from scbasic function

Value

Data object with index rearrangement (multi-level index)

Examples

```
gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

# get genotyping data from gtc files and meta file
df <- scbasic(manifest_path,cluster_path,samplesheet)

# create Data object and rearrange the index
df <- create_from_frame(df)
```

| | |
|---------------|--|
| locus_cluster | <i>To do intensity aggregation at a specific locus</i> |
|---------------|--|

Description

To do intensity aggregation at a specific locus

Usage

```
locus_cluster(df,locus)
```

Arguments

df the pandas dataframe from GenomeStudio or scbasic function
locus the name of the locus

Value

the intensity of one locus

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)

# The Random Forest classifier
call <- locus_cluster(df,'rs3128117')
```

| | |
|----------|--|
| locus_ma | <i>To do m and a aggregation at a specific locus</i> |
|----------|--|

Description

To do m and a aggregation at a specific locus

Usage

```
locus_ma(df, locus)
```

Arguments

| | |
|-------|--|
| df | the pandas dataframe from GenomeStudio or scbasic function |
| locus | the name of the locus |

Value

the m and a of one locus

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)
```

```
# The Random Forest classifier
call <- locus_ma(df, 'rs3128117')
```

| | |
|---------|---|
| pca_chr | <i>To apply principle component annalysis on frequency dataframe of samples of one chromosome</i> |
|---------|---|

Description

To apply principle component annalysis on frequency dataframe of samples

Usage

```
pca_chr(df, chr_name, th=0, n=2)
```

Arguments

| | |
|----------|--|
| df | the pandas dataframe from GenomeStudio or scbasic function |
| chr_name | the name of the specific chromosome |
| th | the threshold |
| n | n is the number of components |

Value

Component values

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs", package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt', package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm', package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv', package='SureTypeSCR')

df <- scbasic(manifest_path, cluster_path, samplesheet)

# The Random Forest classifier
call <- pca_chr(df, 'X')
```

| | |
|-------------|---|
| pca_samples | <i>To apply principle component annalysis on frequency dataframe of samples</i> |
|-------------|---|

Description

To apply principle component annalysis on frequency dataframe of samples

Usage

```
pca_samples(df, th=0)
```

Arguments

| | |
|----|--|
| df | the pandas dataframe from GenomeStudio or scbasic function |
| th | the threshold |

Value

Component values

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs", package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt', package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm', package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv', package='SureTypeSCR')

df <- scbasic(manifest_path, cluster_path, samplesheet)

# The Random Forest classifier
call <- pca_samples(df, th=0.2)
```

| | |
|----------------|---|
| restrict_chrom | <i>To choose certain chromosomes with Data object</i> |
|----------------|---|

Description

To choose certain chromosomes with Data object

Usage

```
restrict_chrom(df, chrom)
```

Arguments

df the Data object from create_from_frame function
 chrom the list of selected chromosomes

Value

Data object only with certain chromosomes

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)

df <- create_from_frame(df)

df <- restrict_chrom(df,c('1','2'))
```

| | |
|-----------|--|
| sample_ma | <i>To do m and a aggregation at a specific chromosome of a specific sample</i> |
|-----------|--|

Description

To do m and a at a specific chromosome of a specific sample

Usage

```
sample_ma(df,sample_name,chr_name)
```

Arguments

df the pandas dataframe from GenomeStudio or scbasic function
 sample_name the name of the sample
 chr_name the name of the chromosome

Value

the m and a of specific chromosome of a specific sample

Examples

```
# parsing file from gtc raw files

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

df <- scbasic(manifest_path,cluster_path,samplesheet)

# The Random Forest classifier
call <- sample_ma(df,'Kit4_4mos_SC21','1')
```

scbasic

Function to process raw gtc data and meta data without genomestudio

Description

Function to process raw gtc data and meta data without genomestudio

Usage

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

Arguments

| | |
|-------------|--------------------------------|
| bpm | a pathname to manifest file |
| egt | a pathname to cluster file |
| samplesheet | a pathname to samplesheet file |

Value

pandas data frame of genotyping data

Examples

```
gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')

# get genotyping data from gtc files and meta file
df <- scbasic(manifest_path,cluster_path,samplesheet)

# /Users/apple/anaconda3/envs/gtc2/lib/python2.7/site-packages/sklearn/ensemble/weight_boosting.py:29:
# DeprecationWarning: numpy.core.umath_tests is an internal NumPy module and should not be imported.
# It will be removed in a future NumPy release.
# from numpy.core.umath_tests import inner1d
# Reading cluster file
```

```
#Reading sample file
#Number of samples: 2
#Reading manifest file
#Initializing genotype data
#Generating
#9968648019_R06C01
#9968648019_R06C02
#Finish parsing
```

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 with elements sc (SureTypeSC), pd (pandas) each referring to python modules.

Examples

```
els = scEls()

els

##$sc
##Module(SureTypeSC)

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

| | |
|--------|--|
| scload | <i>Load Random Forest classifier or Gaussian Discriminate Analysis</i> |
|--------|--|

Description

Load Random Forest classifier or Gaussian Discriminate Analysis

Usage

```
scload(filename)
```

Arguments

filename a pathname to an classifier

Value

instance of a classifier

Examples

```
clf_rf_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')
clf_gda_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')

# The Random Forest classifier
clf_rf <- scload(clf_rf_path)

# The Gaussian Discriminate Analysis classifier
clf_gda <- scload(clf_gda_path)
```

| | |
|-----------|--|
| scpredict | <i>Predictions from Random Forest classifier or Gaussian Discriminant Analysis</i> |
|-----------|--|

Description

Predictions from Random Forest classifier or Gaussian Discriminant Analysis

Usage

```
scpredict(clf,test,clftype='rf')
```

Arguments

| | |
|----------------------|--|
| <code>clf</code> | classifier load by using <code>scload</code> |
| <code>test</code> | Data object including <code>m</code> and <code>a</code> feature |
| <code>clftype</code> | The type of classifier (rf: Random Forest; gda: Gaussian Discriminant Analysis; rf-gda: the cascade of Random Forest and Gaussian Discriminant Analysis) |

Value

The prediction Data object.

The predicted items might include:

`rf_ratio:l_pred`: Random Forest prediction (binary)

`rf_ratio:l_prob`: Random Forest Score for the positive class

`gda_ratio:l_prob`: Gaussian Discriminant Analysis score for the positive class

`gda_ratio:l_pred`: Gaussian Discriminant Analysis prediction (binary)

`rf-gda_ratio:l_prob`: combined 2-layer RF and GDA - probability score for the positive class

`rf-gda_ratio:l_pred`: binary prediction of RF-GDA

Examples

```
gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')
clf_rf_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')
clf_gda_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')

# The Random Forest classifier
clf_rf <- scload(clf_rf_path)

# The Gaussian Discriminant Analysis
clf_gda <- scload(clf_gda_path)

# get genotyping data from gtc files and meta file
df <- scbasic(manifest_path,cluster_path,samplesheet)

# create Data object and rearrange the index
dfs <- create_from_frame(df)

# extract the chromosomes 1 and 2
dfs <- restrict_chrom(dfs,c('1','2'))

# mask the Gencall score lower than 0.01
dfs <- apply_thresh(dfs,0.01)

# calculate the m and a feature
dfs <- calculate_ma(dfs)

# prediction by Random Forest
```

```

result_rf <- scpredict(clf_rf,dfs,clftype='rf')

# prediction by Guassian Discriminate Analysis
result_gda <- scpredict(clf_gda,dfs,clftype='gda')

```

scsave

Save the predictions from different classifiers

Description

Save the predictions from different classifiers

Usage

```

# save different mdoes based on the full prediction table
scsave(result,filename,header=TRUE,clftype='rf',threshold=0.15,all=FALSE)

```

Arguments

| | |
|-----------|---|
| result | The predicted result |
| filename | The path where the result will be saved |
| header | the index |
| clftype | classifier type |
| threshold | the threshold of gencall score |
| all | if the users want to save the full table or not |

Value

txt file of the results

Examples

```

gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')
clf_rf_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')
clf_gda_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')

# The Random Forest classifier
clf_rf = scload(clf_rf_path)

# The Gaussian Discriminant Analysis
clf_gda = scload(clf_gda_path)

# get genotyping data from gtc files and meta file

```

```

df <- scbasic(manifest_path,cluster_path,samplesheet)

# create Data object and rearrange the index
dfs <- create_from_frame(df)

#original shape (294602, 15)
#shape after operation (294602, 12)

# extract the chromosomes 1 and 2
dfs <- restrict_chrom(dfs,c('1','2'))

# mask the Gencall score lower than 0.01
dfs <- apply_thresh(dfs,0.01)

# calculate the m and a feature
dfs <- calculate_ma(dfs)

# prediction by Random Forest
result_rf <- scpredict(clf_rf,dfs,clftype='rf')

# prediction by Guassian Discriminate Analysis
result_gda <- scpredict(clf_gda,dfs,clftype='gda')

# Train the rf-gda classifier
trainer <- scTrain(result_gda,clfname='gda')

# The prediction from the cascade of Random Forest and Gaussian Discriminate Analysis
result_end <- scpredict(trainer,result_gda,clftype='rf-gda')

# Save the complete prediction table
scsave(result_end,'fulltable.txt',clftype='rf',header=TRUE,threshold=0.15,all=TRUE)

# recall mode
scsave(result_end,'fulltable.txt',clftype='rf',threshold=0.15,header=TRUE,all=FALSE)

```

scTrain

Train Gaussian Discriminate Analysis by using the output of predictions of Random Forest

Description

Train Gaussian Discriminate Analysis by using the output of predictions of Random Forest

Usage

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


Arguments

trainingdata Data object, the training data of scTrain
clfname The classifier type

Value

instance of a classifier

Examples

```
gtc_path = system.file("files/GTCs",package='SureTypeSCR')
cluster_path = system.file('files/HumanKaryomap-12v1_A.egt',package='SureTypeSCR')
manifest_path = system.file('files/HumanKaryomap-12v1_A.bpm',package='SureTypeSCR')
samplesheet = system.file('files/Samplesheetr.csv',package='SureTypeSCR')
clf_rf_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')
clf_gda_path = system.file('files/clf_30trees_7228_ratio1_lightweight.clf',package='SureTypeSCR')

# The Random Forest classifier
clf_rf <- scload(clf_rf_path)

# The Gaussian Discriminant Analysis
clf_gda <- scload(clf_gda_path)

# get genotyping data from gtc files and meta file
df <- scbasic(manifest_path,cluster_path,samplesheet)

# create Data object and rearrange the index
dfs <- create_from_frame(df)

# extract the chromosomes 1 and 2
dfs <- restrict_chrom(dfs,c('1','2'))

# mask the Gencall score lower than 0.01
dfs <- apply_thresh(dfs,0.01)

# calculate the m and a feature
dfs <- calculate_ma(dfs)

# prediction by Random Forest
result_rf <- scpredict(clf_rf,dfs,clftype='rf')

# prediction by Guassian Discriminate Analysis
result_gda <- scpredict(clf_gda,dfs,clftype='gda')

# Train the rf-gda classifier
trainer <- scTrain(result_rf,clfname='gda')

# The prediction from the cascade of Random Forest and Gaussian Discriminate Analysis
result_end <- scpredict(trainer,result_gda,clftype='rf-gda')
```


Index

allele_freq, [2](#)
apply_thresh, [2](#)

calculate_ma, [3](#)
callrate, [4](#)
callrate_chr, [5](#)
create_from_frame, [6](#)

locus_cluster, [6](#)
locus_ma, [7](#)

pca_chr, [8](#)
pca_samples, [9](#)

restrict_chrom, [9](#)

sample_ma, [10](#)
scbasic, [11](#)
scEls, [12](#)
scload, [13](#)
scpredict, [13](#)
scsave, [15](#)
scTrain, [16](#)