Package 'SPIAssay'

August 19, 2016

Type Package

Title A Genetic-Based Assay for the Identification of Cell Lines

Version 1.1.0

Date 2016-08-19

Author Francesca Demichelis <francesca.demichelis@unitn.it>, Davide Prandi <davide.prandi@unitn.it>

Maintainer Davide Prandi <davide.prandi@unitn.it>

Depends R (>= 1.8.0)

Description The SNP Panel Identification Assay (SPIA) is a package that enables an accurate determination of cell line identity from the genotype of single nucleotide polymorphisms (SNPs). The SPIA test allows to discern when two cell lines are close enough to be called similar and when they are not. Details about the method are available at ``Demichelis et al. (2008) SNP panel identification assay (SPIA): a genetic-based assay for the identification of cell lines. Nucleic Acids Res., 3, 2446-2456".

License GPL (>= 2)

biocViews SNP, Bioinformatics, MultipleComparisons, QualityControl

URL http://www.r-project.org

Repository CRAN

NeedsCompilation no

Date/Publication 2016-08-19 14:21:01

R topics documented:

SPIAPlot *Function to visualize the result of the SPIA test*

Description

Function SPIAPlot allows the user to rapidly visualize the result of the SPIA test.

Usage

```
SPIAPlot(SPIAanalysis)
```

Arguments

SPIAanalysis	The result of the SPIATest function

Value

A plot of the SPIA test

Author(s)

Francesca Demichelis <francesca.demichelis@unitn.it>, Davide Prandi <davide.prandi@unitn.it>

Examples

```
# An example of genotype calls with:
# - four cell lines
# - for each cell line five SNP calls
GenotypeCalls <- rbind(
c("SNP1", "AA", "AA", "AB", "AB"),
c("SNP2", "NoCall", "AA", "AB", "AB"),
c("SNP3", "AB", "AB", "AA" , "AB"),
c("SNP4", "BB", "BB", "BB", "BB"),
c("SNP5", "AB", "BB", "AA", "AA"))
colnames(GenotypeCalls) <-
 c("SNP_ID","CellLine1","CellLine2","CellLine3","CellLine4")

# Encode the data into SPIA format.
# SPIA uses 0 for AA, 1 for BB, 2 for AB, and NA for NoCall
# therefore, GenotypeCalls has to be encoded by meand of
# toSPIAData
encoding <- c("AA","BB","AB","NoCall")
SPIAGenotypeCalls <-
 toSPIAData(GenotypeCalls,encoding)


# Perform SPIA analysis
SPIAParam <-
 list(Pmm=0.1, nsigma=1, Pmm_nonM=0.6, nsigma_nonM=1, PercValidCall=0.7)
```

```
SPIAanalysis <-
 SPIATest(SPIAGenotypeCalls, row.names=TRUE, test.prob=TRUE, SPIAParam)

# Plot summary
SPIAPlot(SPIAanalysis)
```

SPIAssay *A genetic-based assay for the identification of cell lines*

Description

The SNP Panel Identification Assay (SPIA) is a package that enables an accurate determination of cell line identity from the genotype of single nucleotide polymorphisms (SNPs). The SPIA test allows to discern when two cell lines are close enough to be called similar and when they are not.

Details

Package:	SPIAssay
Type:	Package
Version:	1.1.0
Date:	2011-07-26
License:	GPL Version 2 or later.

The most important function of the package is `SPIATest` that computes SPIA distance on a set of cell lines and perform the associated probabilistic test. Another important function is `SPIAPlot` that creates a summary graph of the SPIA test. Finally, `toSPIAData` function encodes generic representations of genotype calls into a format compatible with SPIA.

Author(s)

Francesca Demichelis <francesca.demichelis@unitn.it>, Davide Prandi <davide.prandi@unitn.it>

Maintainer: Davide Prandi <davide.prandi@unitn.it>

References

Demichelis F, Greulich H, Macoska JA, Beroukhim R, Sellers WR, Garraway L, Rubin MA. SNP panel identification assay (SPIA): a genetic-based assay for the identification of cell lines. *Nucleic Acids Res*. 2008;36:2446-2456.

Examples

```
library(SPIAssay)

# An example of genotype calls with:
# - four cell lines
```

```
# - for each cell line five SNP calls
GenotypeCalls <- rbind(
c("SNP1", "AA", "AA", "AB", "AB"),
c("SNP2", "NoCall", "AA", "AB", "AB"),
c("SNP3", "AB", "AB", "AA" , "AB"),
c("SNP4", "BB", "BB", "BB", "BB"),
c("SNP5", "AB", "BB", "AA", "AA"))
colnames(GenotypeCalls) <-
 c("SNP_ID","CellLine1","CellLine2","CellLine3","CellLine4")

# Encode the data into SPIA format.
# SPIA uses 0 for AA, 1 for BB, 2 for AB, and NA for NoCall
# therefore, GenotypeCalls has to be encoded by meand of toSPIAData
encoding <- c("AA","BB","AB","NoCall")
SPIAGenotypeCalls <- toSPIAData(GenotypeCalls,encoding)

# Perform SPIA analysis with parameters SPIAParam
SPIAParam <-
 list(Pmm=0.1, nsigma=1, Pmm_nonM=0.6, nsigma_nonM=1, PercValidCall=0.7)
SPIAanalysis <-
 SPIATest(SPIAGenotypeCalls, row.names=TRUE, test.prob=TRUE, SPIAParam)

# Plot SPIA pairwise comparison
SPIAPlot(SPIAanalysis)
```

`SPIATest` *Function to compute SPIA test*

Description

Function `SPIATest` computes SPIA distance and performs probabilistic test on a set of cell lines.

Usage

```
SPIATest(x, row.names = TRUE, test.prob = TRUE,
         test.param = list(Pmm=0.1, nsigma=2,
         Pmm_nonM=0.6, nsigma_nonM=3,
         PercValidCall=0.9), verbose = FALSE)
```

Arguments

`x`	a matrix with a column for each cell line and a row for each SNP
`row.names`	specify if the first column of x contains SNPs names/identificators
`test.prob`	specify if the function has to perform SPIA probabilistic test
`test.param`	specify the parameters of the probabilistic test. • Pmm: SNP probability of mismatch in a matching population • nsigma: area limit for Pmm

- Pmm_nonM: SNP probability of mismach in a non matching population
- nsigma_nonM: area limit for Pmm_nonM
- PercValidCall: percentage of valid SNP calls to consider the test valid

`verbose` print verbose information

Value

`SPIAresult` a matric with a line for each cell line and with columns with the informationss about distances In particular, each row of SPIAresult has 7 columns:

1. SPIA distance
2. number of valid calls
3. number of total calls
4. number of calls where one of the two SPNs are not available
5. number of calls where both SNPs are not available
6. number of calls where SNP change from AA, BB to AB or from AB to AA, BB
7. number of calls where SNP change from AA to BB or from BB to AA

`parameters` the parameters used by the test (test.param)

`input.param` the number of samples (N_samples), the number of SNPs (N_SNPs), and if the probabilistic test has been done (test.prob)

Author(s)

Francesca Demichelis <francesca.demichelis@unitn.it>, Davide Prandi <davide.prandi@unitn.it>

Examples

```
library(SPIAssay)

# An example of genotype calls with:
# - four cell lines
# - for each cell line five SNP calls
GenotypeCalls <- rbind(
c("SNP1", "AA", "AA", "AB", "AB"),
c("SNP2", "NoCall", "AA", "AB", "AB"),
c("SNP3", "AB", "AB", "AA" , "AB"),
c("SNP4", "BB", "BB", "BB", "BB"),
c("SNP5", "AB", "BB", "AA", "AA"))
colnames(GenotypeCalls) <-
 c("SNP_ID","CellLine1","CellLine2","CellLine3","CellLine4")

# Encode the data into SPIA format.
# SPIA uses 0 for AA, 1 for BB, 2 for AB, and NA for NoCall
# therefore, GenotypeCalls has to be encoded by meand of
# toSPIAData
encoding <- c("AA","BB","AB","NoCall")
```

```
SPIAGenotypeCalls <- toSPIAData(GenotypeCalls,encoding)

# Perform SPIA analysis
SPIAanalysis <- SPIATest(SPIAGenotypeCalls)
# This analysis give an error because there are not enough SNPs

# Perform SPIA analysis with parameters SPIAParam
# to reduce the 'similar' region
SPIAParam <-
 list(Pmm=0.1, nsigma=1, Pmm_nonM=0.6, nsigma_nonM=1, PercValidCall=0.7)
SPIAanalysis <-
 SPIATest(SPIAGenotypeCalls,row.names=TRUE,test.prob=TRUE,SPIAParam)
```

`toSPIAData` *Function to code SNPs data into SPIA format*

Description

Function `toSPIAData` encodes SNPs information in a format accepted by SPIA, i.e., 0 for AA, 1 for BB, 2 for AB, NA for NoCall.

Usage

```
toSPIAData(SNPMatrix, encoding)
```

Arguments

`SNPMatrix`	a matrix with a column for each cell line and a row for each SNP
`encoding`	a four elements ecnoding vector describing the encoding used in SNPMatrix. For instance, (0,2,1,-1) says that SNPMatrix uses 0 for AA, 2 for BB, 1 for AB, and -1 for NoCall

Value

a matrix with a column for each cell line and a row for each SNP encoded with the SPIA format

Author(s)

Francesca Demichelis <francesca.demichelis@unitn.it>, Davide Prandi <davide.prandi@unitn.it>

Examples

```
SNPcall <- c("snp1",0,1,2,-1)
encoding <- c(0,2,1,-1)
SPIA_SNPcall <- toSPIAData(SNPcall, encoding)
print(SNPcall)
print(SPIA_SNPcall)
```

Index