Package ‘annotationTools’

December 19, 2024

Version 1.81.0

Date 2020-03-27

Title Annotate microarrays and perform cross-species gene expression analyses using flat file databases

Author Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

Maintainer Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

Imports Biobase, stats

Suggests BiocStyle

Description Functions to annotate microarrays, find orthologs, and integrate heterogeneous gene expression profiles using annotation and other molecular biology information available as flat file database (plain text files).

biocViews Microarray, Annotation

License GPL

ZipData no

git_url https://git.bioconductor.org/packages/annotationTools

git_branch devel

git_last_commit ad67b95

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-12-19

Contents

annot_HGU133A 2
compactList 2
gene_orthologs_part 3
getANNOTATION 4
getGENEID 6
getGENEONTOLOGY 7
getGENESYMBOL 9

getGENETITLE 11
getHOMOLOG 13
getMULTIANNOTATION 15
getOrthologousProbesets 17
getPROBESET 19
HG-U133_Plus_2_annot_part 21
HG-U133_Plus_2_ortholog_part 21
homologene_part 22
listToCharacterVector 22
Mouse430_2_annot_part 23
ortho 23
ps2ps 24
table_human 25
table_mouse 26

Index **27**

`annot_HGU133A` *annot_HGU133A*

Description

Partial Affymetrix annotation for HG-U133A. Used in the example cross-species analysis presented in the vignette of package annotationTools.

Format

data.frame with 188 rows and 43 columns

Source

http://www.affymetrix.com

`compactList` *Concatenate list elements*

Description

Concatenates given elements of a list.

Usage

```
compactList(lst,l)
```

Arguments

`lst`	list object
`l`	numeric vector specifying the number of list elements to be concatenated sequentially

Details

By definition, the output list is shorter than the input list.

Function stops if 'sum(l)' does not equal 'lenght(lst)'.

Value

`clst`	list of length 'length(l)' where the 'i'-th element has length 'l[i]' and contains concatenated elements of input list 'lst'.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

Examples

```
##an example list
lst<-vector('list',3)
lst[[1]]<-c('aaa','bbb')
lst[[2]]<-'ccc'
lst[[3]]<-'zzz'
##merge first 2 list elements
compactList(lst,c(2,1))
```

`gene_orthologs_part` *Example gene_orthologs file*

Description

Shortened gene_orthologs database file.

Format

A short version of 'gene_orthologs' table containing 2 ortholog clusters. Ortholog clusters in 'gene_orthologs' derive from NCBI's Eukaryotic Genome Annotation pipeline (see https://www.ncbi.nlm.nih.gov/kis/info/ho are-orthologs-calculated/).

Source

ftp.ncbi.nlm.nih.gov/gene/DATA/gene_orthologs.gz

`getANNOTATION` *General annotation function*

Description

Takes a vector of identifiers and an annotation table and matches the identifiers in the table to retrieve the corresponding annotation. Only the first occurence of each identifier in the annotation table is considered.

Usage

```
getANNOTATION(identifier, annot, diagnose = FALSE, identifierCol = 1, annotationCol = 15, noAnnotationS
```

Arguments

`identifier`	vector containing identifiers to be annotated.
`annot`	annotation table (data frame) where each row is a record and each column is an annotation field.
`diagnose`	logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.
`identifierCol`	column in annotation table where the provided identifiers are to be looked up.
`annotationCol`	column in annotation table containing the desired annotation.
`noAnnotationSymbol`	character string to be used in output list 'annotation' if no annotation is found or provided.
`noAnnotationProvidedSymbol`	character string used in annotation table and indicating missing annotation.
`sep`	character string used in annotation table to separate multiple annotation of a single identifier.

Details

The annotation is returned as elements of list 'annotation'. If a single annotation is given for a particular identifier, the corresponding element of 'annotation' has length 1. If multiple annotation is provided for a single identifier (i.e. character string with 'sep' separating multiple annotations), the mulitple annotation is split and the corresponding vector is returned as an element of list 'annotation'.

Value

annotation	list of length 'length(identifier)' the 'i'-th element of which contains the annotation for 'identifier[i]'.
empty	logical vector of length 'length(identifier)'. 'empty[i]' is TRUE if 'identifier[i]' is empty or NA.
noentry	locial vector of length 'length(identifier)'. 'noentry[i]' is TRUE if 'identifier[i]' cannot be found in 'annot[,identifierCol]'.
noannotation	locial vector of length 'length(identifier)'. 'noannotation[i]' is TRUE if 'a[i]==noAnnotationProvidedSym is TRUE.

Note

Use `getMULTIANNOTATION` if the identifiers occur on more than one line in the annotation table.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

See Also

`getMULTIANNOTATION`

Examples

```
##example annotation table
annotation<-cbind(gene=c('gene_1a, gene_1b','gene_2','gene_3','gene_4'),probe=c('probe_1','probe_2','probe_3','
print(annotation)

##get sequences for probe_2, probe_3, probe_4 and probe_100
myProbes<-c('probe_2','probe_3','probe_4','probe_100',NA)
getANNOTATION(myProbes,annotation,identifierCol=2,annotationCol=3,noAnnotationProvidedSymbol='',sep=', ')

##track origin of annotation failure for the last 3 probes
getANNOTATION(myProbes,annotation,identifierCol=2,annotationCol=3,noAnnotationProvidedSymbol='',sep=', ',diagno
```

`getGENEID` *Find gene IDs*

Description

Takes a vector of probe set identifiers and an annotation table and retrieves the corresponding gene IDs.

Usage

```
getGENEID(ps, annot, diagnose = FALSE, idCol = 19, noIDsymbol = NA, noIDprovidedSymbol = "---", sep = " /
```

Arguments

`ps`	character vector containing the probe sets identifiers.
`annot`	annotation table (data frame) where each row is a record and each column is an annotation field.
`diagnose`	logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.
`idCol`	column in annotation table containing the gene IDs.
`noIDsymbol`	character string to be used in output list 'geneid' if no gene ID is found or provided in the annotation table.
`noIDprovidedSymbol`	character string used in annotation table and indicating missing gene ID.
`sep`	character string used in annotation table to separate multiple gene IDs.

Details

This function can be used with Affymetrix annotation files (e.g. 'HG-U133_Plus_2_annot.csv'). It retrieves gene IDs corresponding to particular probe set identifiers.

Gene IDs are returned as elements of list 'geneid'. If multiple gene IDs are provided for 'ps[i]' (with 'sep' separating gene IDs in the annotation table), a vector containing all gene IDs is returned as the 'i-th' element of list 'geneid'.

The default values for 'idCol', 'noIDsymbol', 'noIDprovidedSymbol' and 'sep' are chosen to suit the format of Affymetrix annotation files. However, options can be set to look up any annotation table, provided the probe set identifiers are in the first column and occur only once.

Value

`geneid`	list of length 'length(ps)' the 'i'-th element of which contains the gene ID for 'ps[i]'.
`empty`	logical vector of length 'length(ps)'. 'empty[i]' is TRUE if 'ps[i]' is empty or NA.
`noentry`	logial vector of length 'length(ps)'. 'noentry[i]' is TRUE if 'ps[i]' cannot be found in the first column of the annotation table.

noid logial vector of length 'length(ps)'. 'noid[i]' is TRUE if 'geneid[i]==noIDprovidedSymbol' is TRUE.

Note

getANNOTATION provides a more flexible solution to be used with arbitrary annotation tables.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

See Also

getANNOTATION

Examples

```
##example Affymetrix annotation file and its location
annotationFile<-system.file('extdata','HG-U133_Plus_2_annot_part.csv',package='annotationTools')

##load annotation file
annotation<-read.csv(annotationFile,colClasses='character',comment.char='#')

##get gene IDs
myPS<-c('117_at','1007_s_at','1552288_at',NA,'xyz_at')
getGENEID(myPS,annotation)

##track origin of annotation failure for the 3 last probe set IDs
getGENEID(myPS,annotation,diagnose=TRUE)
```

---

getGENEONTOLOGY *Find Gene Ontology (GO) annotation*

Description

Takes a vector of probe set identifiers and an annotation table and retrieves the corresponding GO annotation.

Usage

```
getGENEONTOLOGY(ps, annot, diagnose = FALSE, specifics = 0, GOcol = 31, noGOsymbol = NA, noGOprovidedSym
```

Arguments

- `ps` — character vector containing the probe sets identifiers.
- `annot` — annotation table (data frame) where each row is a record and each column is an annotation field.
- `diagnose` — logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.
- `specifics` — can take value 0, 1, 2, 3, ... If specifics=i with i>0, the GO biological process annotation is parsed (using " // " as separator) and the i-th part of the expression is returned. If specifics=0, the GO biological process annotation is not parsed.
- `GOcol` — column in annotation table containing the GO biological processes.
- `noGOsymbol` — character string to be used in output list 'go' if no GO biological process is found or provided in the annotation table.
- `noGOprovidedSymbol` — character string used in annotation table and indicating missing GO biological process.
- `sep` — character string used in annotation table to separate multiple GO biological processes.

Details

This function can be used with Affymetrix annotation files (e.g. 'HG-U133_Plus_2_annot.csv'). It retrieves GO annotation corresponding to particular probe set identifiers. GO biological processes are returned by default ('GOcol'=31) but GO cellular components ('GOcol'=32) or GO molecular functions ('GOcol'=33) can be returned by setting 'GOcol' appropriately.

GO biological processes are returned as elements of list 'go'. If multiple GO biological processes are provided for 'ps[i]' (with 'sep' separating GO biological processes in the annotation table), a vector containing all GO biological processes is returned as the 'i-th' element of list 'go'.

The default values for 'GOcol', 'noGOsymbol', 'noGOprovidedSymbol' and 'sep' are chosen to suit the format of Affymetrix annotation files. However, options can be set to look up any annotation table, provided the probe set identifiers are in the first column and occur only once.

Note that each GO annotation in Affymetrix annotation files contains 3 attributes: the GO biological process ID, term and quality, separated by " // ". Setting the option 'specifics' to 1, 2, or 3 allows to retrieve any of the 3 attributes separately.

Value

- `go` — list of length 'length(ps)' the 'i'-th element of which contains the GO annotation for 'ps[i]'.
- `empty` — logical vector of length 'length(ps)'. 'empty[i]' is TRUE if 'ps[i]' is empty or NA.
- `noentry` — locial vector of length 'length(ps)'. 'noentry[i]' is TRUE if 'ps[i]' cannot be found in the first column of the annotation table.
- `nogo` — locial vector of length 'length(ps)'. 'nogo[i]' is TRUE if 'go[i]==noIDprovidedSymbol' is TRUE.

Note

getANNOTATION provides a more flexible solution to be used with arbitrary annotation tables.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

See Also

getANNOTATION

Examples

```
##example Affymetrix annotation file and its location
annotationFile<-system.file('extdata','HG-U133_Plus_2_annot_part.csv',package='annotationTools')

##load annotation file
annotation<-read.csv(annotationFile,colClasses='character',comment.char='#')

##get gene GO biological process (full information)
myPS<-c('117_at','1007_s_at','1552288_at',NA,'xyz_at')
getGENEONTOLOGY(myPS,annotation)

##get gene GO biological process terms only
getGENEONTOLOGY(myPS,annotation,specifics=2)

##track origin of annotation failure for the 3 last probe set IDs
getGENEONTOLOGY(myPS,annotation,diagnose=TRUE)

##GO molecular functions are contained in column 33 of the annotation
colnames(annotation)

##get gene GO molecular functions
getGENEONTOLOGY(myPS,annotation,GOcol=33)
```

---

getGENESYMBOL *Find gene symbols*

Description

Takes a vector of probe set identifiers and an annotation table and retrieves the corresponding gene symbols.

Usage

```
getGENESYMBOL(ps, annot, diagnose = FALSE, GScol = 15, noGSsymbol = NA, noGSprovidedSymbol = "---", sep
```

Arguments

`ps`	character vector containing the probe sets identifiers.
`annot`	annotation table (data frame) where each row is a record and each column is an annotation field.
`diagnose`	logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.
`GScol`	column in annotation table containing the gene IDs.
`noGSsymbol`	character string to be used in output list 'symbols' if no gene symbol is found or provided in the annotation table.
`noGSprovidedSymbol`	character string used in annotation table and indicating missing gene symbol.
`sep`	character string used in annotation table to separate multiple gene symbols.

Details

This function can be used with Affymetrix annotation files (e.g. 'HG-U133_Plus_2_annot.csv'). It retrieves the gene symbols corresponding to particular probe set identifiers.

Gene symbols are returned as elements of list 'symbols'. If multiple gene symbols are provided for 'ps[i]' (with 'sep' separating gene symbols in the annotation table), a vector containing all gene symbols is returned as the 'i-th' element of list 'symbols'.

The default values for 'GScol', 'noGSsymbol', 'noGSprovidedSymbol' and 'sep' are chosen to suit the format of Affymetrix annotation files. However, these options can be set to look up any annotation table, provided the probe set identifiers are in the first column and occur only once.

Value

`symbols`	list of length 'length(ps)' the 'i'-th element of which contains the gene symbol for 'ps[i]'.
`empty`	logical vector of length 'length(ps)'. 'empty[i]' is TRUE if 'ps[i]' is empty or NA.
`noentry`	locial vector of length 'length(ps)'. 'noentry[i]' is TRUE if 'ps[i]' cannot be found in the first column of the annotation table.
`nogs`	locial vector of length 'length(ps)'. 'nogs[i]' is TRUE if 'symbols[i]==noIDprovidedSymbol' is TRUE.

Note

`getANNOTATION` provides a more flexible solution to be used with arbitrary annotation tables.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

See Also

getANNOTATION

Examples

```
##example Affymetrix annotation file and its location
annotationFile<-system.file('extdata','HG-U133_Plus_2_annot_part.csv',package='annotationTools')

##load annotation file
annotation<-read.csv(annotationFile,colClasses='character',comment.char='#')

##get gene symbols
myPS<-c('117_at','1007_s_at','1552288_at',NA,'xyz_at')
getGENESYMBOL(myPS,annotation)

##track origin of annotation failure for the 3 last probe set IDs
getGENESYMBOL(myPS,annotation,diagnose=TRUE)
```

---

getGENETITLE *Find gene titles*

Description

Takes a vector of probe set identifiers and an annotation table and retrieves the corresponding gene titles.

Usage

```
getGENETITLE(ps, annot, diagnose = FALSE, TITLEcol = 14, noTITLEsymbol = NA, noTITLEprovidedSymbol = "--
```

Arguments

ps	character vector containing the probe sets identifiers.
annot	annotation table (data frame) where each row is a record and each column is an annotation field.
diagnose	logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.
TITLEcol	column in annotation table containing the gene titles.
noTITLEsymbol	character string to be used in output list 'genetitle' if no gene title is found or provided in the annotation table.
noTITLEprovidedSymbol	character string used in annotation table and indicating missing gene title.
sep	character string used in annotation table to separate multiple gene titles.

Details

This function can be used with Affymetrix annotation files (e.g. 'HG-U133_Plus_2_annot.csv'). It retrieves the gene titles corresponding to particular probe set identifiers.

Gene titles are returned as elements of list 'genetitle'. If multiple gene titles are provided for 'ps[i]' (with 'sep' separating gene titles in the annotation table), a vector containing all gene titles is returned as the 'i-th' element of list 'genetitle'.

The default values for 'TITLEcol', 'noTITLEsymbol', 'noTITLEprovidedSymbol' and 'sep' are chosen to suit the format of Affymetrix annotation files. However, these options can be set to look up any annotation table, provided the probe set identifiers are in the first column and occur only once.

Value

`genetitle`	list of length 'length(ps)' the 'i'-th element of which contains the gene title for 'ps[i]'.
`empty`	logical vector of length 'length(ps)'. 'empty[i]' is TRUE if 'ps[i]' is empty or NA.
`noentry`	locial vector of length 'length(ps)'. 'noentry[i]' is TRUE if 'ps[i]' cannot be found in the first column of the annotation table.
`notitle`	locial vector of length 'length(ps)'. 'noid[i]' is TRUE if 'genetitle[i]==noTITLEprovidedSymbol' is TRUE.

Note

`getANNOTATION` provides a more flexible solution to be used with arbitrary annotation tables.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

See Also

`getANNOTATION`

Examples

```
##example Affymetrix annotation file and its location
annotationFile<-system.file('extdata','HG-U133_Plus_2_annot_part.csv',package='annotationTools')

##load annotation file
annotation<-read.csv(annotationFile,colClasses='character',comment.char='#')

##get gene titles
```

```
myPS<-c('117_at','1007_s_at','1552288_at',NA,'xyz_at')
getGENETITLE(myPS,annotation)

##track origin of annotation failure for the 3 last probe set IDs
getGENETITLE(myPS,annotation,diagnose=TRUE)
```

---

getHOMOLOG *Find homologous/orthologous gene (ID)*

Description

Takes a vector of gene IDs, a table of homologs/orthologs, and a target species and returns gene IDs corresponding to homologous/orthologous genes.

Usage

```
getHOMOLOG(geneid, targetspecies, homol, cluster = FALSE, diagnose = FALSE, noIDsymbol = NA, clusterCol
```

Arguments

geneid	character vector containing gene IDs.
targetspecies	identifier of the target species in the homology/orthology table.
homol	homology/orthology table (data frame) listing gene IDs (1 per line) along with the species and the homology/orthology cluster they belong to.
cluster	logical. If TRUE, the identifiers provided in 'geneid' are homology/orthology cluster IDs. If FALSE, they are gene IDs.
diagnose	logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.
noIDsymbol	character string to be used in output list 'targetid' if no homologous/orthologous gene is found or provided in the annotation table.
clusterCol	column in homology/orthology table containing homology/orthology cluster IDs.
speciesCol	column in homology/orthology table containing species IDs.
idCol	column in homology/orthology table containing gene IDs.
tableType	character string specifying the type of homology/orthology table used. Either 'homologene' (default) or 'gene_orthologs'.

Details

The homology/orthology table lists gene IDs (from several species) and the homology/orthology cluster they belong to. Homologous and orthologous genes share a common cluster identifier. Given a certain gene ID, a target species, and a homology/orthology table, all gene IDs belonging to the same homology/orthology cluster and to the specified target species are returned. Various homology/orthology databases can be used, in particular NCBI's HomoloGene and their 'Orthologs from Annotation pipeline', referred to as 'gene_orthologs' database (see details below). If 'targetspecies' is the species 'geneid' belongs to, by definition, homologous genes are returned (if listed). On the

contrary, specifying a 'targetspecies' different from the host species 'geneid' belongs to, results in orthologous genes to be returned. Note that each gene ID is assumed to be unique and to belong to a single homology/orthology cluster.

Gene IDs of homologous/orthologous genes are returned as elements of list 'targetid'. If multiple (homologous/orthologous) gene IDs are provided for 'geneid[i]', a vector containing all gene IDs is returned as the 'i-th' element of list 'targetid'.

Default values for 'clusterCol', 'speciesCol', and 'idCol' are chosen to match the table provided by HomoloGene (homologene.data provided by www.ncbi.nlm.nih.gov/HomoloGene). Homology/orthology tables from other sources might require setting these values appropriately.

Orthologs defined in NCBI's 'Orthologs from Annotation pipeline' database (available at ftp.ncbi.nlm.nih.gov/gene/DATA/ge and hence referred to as 'gene_orthologs') can be mined by setting 'tableType' to 'gene_orthologs' instead of 'homologene' (default). In this case, arguments 'clusterCol', 'speciesCol',and 'idCol' are overridden to fit the data structure used in 'gene_orthologs'. In short, one difference between 'gene_orthologs' compared to HomoloGene is that 'gene_orthologs' does not use ortholog cluster IDs but anchors each ortholog groups using the human gene ID of the ortholog group. If a specific ortholog group does not contain a human gene, a gene ID from another species within the ortholog group may be used as anchor.

'gene_orthologs' is a rich source of homologs/orthologs between selected vertebrate species and introduced by NCBI in 2014 (https://www.ncbi.nlm.nih.gov/kis/info/how-are-orthologs-calculated/). Note that 'gene_orthologs' does not list (and thus cannot be used to search for) homologs (i.e. only 'Ortholog' relationships, as specified in the 3rd column of 'gene_orthologs' are considered).

Finally, if 'cluster' is TRUE, cluster IDs can be provided in 'geneid' (instead of gene IDs) and the function will return all (homologous/orthologous) gene IDs belonging to a given cluster ID and a given 'targetspecies'. This can be used to mine orthology tables provided by Affymetrix (e.g. 'Mouse430_2_ortholog.csv') for orthologous probe set IDs (see 'examples' below).

Value

`targetid` list of length 'length(geneid)' the 'i'-th element of which contains the homologous/orthologous gene IDs for 'geneid[i]' and 'targetspecies'.

`empty` logical vector of length 'length(geneid)'. 'empty[i]' is TRUE if 'geneid[i]' is empty or NA.

`noentry` locial vector of length 'length(geneid)'. 'noentry[i]' is TRUE if 'geneid[i]' cannot be found in column 'idCol' (default is column 3) of the homology/orthology table 'homol'.

`notargetid` locial vector of length 'length(geneid)'. 'notargetid[i]' is TRUE if 'geneed[i]' is found in the homology/orthology table but no homolog/ortholog is listed for 'targetspecies'.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

Examples

```
##example Homologene file and its location
homologeneFile<-system.file('extdata','homologene_part.data',package='annotationTools')

##load Homologene file
homologene<-read.delim(homologeneFile,header=FALSE)

##get mouse (species ID 10090) orthologs of several human (species ID 9606) gene ID (among those: 5982, gene symbol F
myGenes<-c(5982,93587,NA,100)
getHOMOLOG(myGenes,10090,homologene)

##track origin of annotation failure for the last 2 gene IDs
getHOMOLOG(myGenes,10090,homologene,diagnose=TRUE)

##get mouse gene belonging to homologene cluster IDs 6885 and 6886
myClusters<-c(6885,6886)
getHOMOLOG(myClusters,10090,homologene,cluster=TRUE)

##get mouse orthologs of human genes using 'gene_orthologs'
gene_orthologsFile<-system.file('extdata','gene_orthologs_part.data',package='annotationTools')
gene_orthologs<-read.delim(gene_orthologsFile,header=TRUE)
getHOMOLOG(myGenes,10090,gene_orthologs,tableType='gene_orthologs')

##mine Affymetrix (example) ortholog file
affyOrthologFile<-system.file('extdata','HG-U133_Plus_2_ortholog_part.csv',package='annotationTools')
affyOrthologs<-read.csv(affyOrthologFile,colClasses='character')

##get Mouse430_2 probe set IDs 'orthologous' to HG-U133_Plus_2 probe set IDs 1053_at and 121_at
myPS<-c('1053_at','121_at')
getHOMOLOG(myPS,'Mouse430_2',affyOrthologs,cluster=TRUE,clusterCol=1,speciesCol=4,idCol=3)
```

---

getMULTIANNOTATION *General (multiple) annotation function*

Description

Takes a vector of identifiers and an annotation table and matches the identifiers in the table to retrieve the corresponding annotation. Identifiers can occur on multiple records (i.e. lines) of the annotation table.

Usage

```
getMULTIANNOTATION(identifier, annot, diagnose = FALSE, identifierCol = 19, annotationCol = 1, noAnnota
```

Arguments

identifier	vector containing identifiers to be annotated.
annot	annotation table (data frame) where each row is a record and each column is an annotation field.

`diagnose` logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.

`identifierCol` column in annotation table where the provided identifiers are to be looked up.

`annotationCol` column in annotation table containing the desired annotation.

`noAnnotationSymbol` chacter string to be used in output list 'annotation' if no annotation is found or provided.

`noAnnotationProvidedSymbol` character string used in annotation table and indicating missing annotation.

Details

The annotation is returned as elements of list 'annotation'. If the 'i'-th identifier occur on multiple lines, all corresponding annotation are retrieved and output as a vector. The length of the 'i'-th element of 'annotation' thus equals the number of lines for 'identifier[i]' in the annotation table.

Value

`annotation` list of length 'length(identifier)' the 'i'-th element of which contains the annotation for 'identifier[i]'.

`empty` logical vector of length 'length(identifier)'. 'empty[i]' is TRUE if 'identifier[i]' is empty or NA.

`noentry` locial vector of length 'length(identifier)'. 'noentry[i]' is TRUE if 'identifier[i]' cannot be found in 'annot[,identifierCol]'.

`noannotation` locial vector of length 'length(identifier)'. 'noannotation[i]' is TRUE if 'a[i]==noAnnotationProvidedSym is TRUE.

Note

`getANNOTATION` runs faster and is to be used if the identifiers occur only once in the annotation table.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

See Also

`getANNOTATION`

Examples

```
##example annotation table
annotation<-cbind(gene=c('gene_1','gene_2','gene_2','gene_3','gene_4'),probe=c('probe_1','probe_2a','probe_2b'
print(annotation)

##get sequences for gene_2, gene_3, gene_4 and gene_100
myGenes<-c('gene_2','gene_3','gene_4','gene_100', NA)
getMULTIANNOTATION(myGenes,annotation,identifierCol=1,annotationCol=2,noAnnotationProvidedSymbol='')

##track origin of annotation failure for the 3 last genes
getMULTIANNOTATION(myGenes,annotation,identifierCol=1,annotationCol=2,noAnnotationProvidedSymbol='',diagnose=T
```

---

getOrthologousProbesets

Find orthologous/homologous probe sets present in a target set using a mapping table

---

Description

Used for cross-species analysis of gene expression profiles. Takes a vector of probe sets (species 1), a data.frame containing a second set of probe sets (species 2) and associated values (e.g. log fold change, t-statistic, ...) , a mapping table of orthologous probe sets (species 1 to species 2) and returns the list of probe sets orthologous to those in the first vector and found in the second set.

Usage

```
getOrthologousProbesets(ps1,ps2,ps2ps,fct=function(x){x},forceProbesetSelection=FALSE)
```

Arguments

`ps1` A vector of probe sets identifiers.

`ps2` A data.frame where the first column contains the target probe sets. The second column contains quantities associated with each probe set like e.g. log fold change, t-statistic, etc.

`ps2ps` A data.frame containing a mapping table of orthologous probe sets. Can be generated with the function ps2ps.R.

`fct` A function for probe sets selection. The function is applied to the values (second column of ps2) associated with all orthologous probe sets corresponding to a given probe set in ps1.

`forceProbesetSelection`
Logical. If FALSE (default), all orthologous probe sets found in ps2 are returned. If TRUE, only the probe sets associated with values selected with the function fct are returned.

Details

Each probe set in the first input vector (ps1) is looked up in the mapping table (ps2ps). Orthologous probe sets given in the mapping table and present in the second input argument (more precisely in the first column of ps2) are returned as a list (ps2_probeSel). By default, values associated with these orthologous probe sets (and given in the second column of ps2) are returned as a list too (ps2_value).

A function can be specified and applied to the values associated with the orthologous probe sets. This can for instance be used to assign a summary value associated with a set orthologous probe sets (e.g. fct=mean or fct=median in case the associated values are log fold changes). Alternatively, this can be used to filter a single probe set out of multiple orthologous probe sets (e.g. fct=min in case the associated value is a p-value) (see below).

By default, the function returns all orthologous probe sets found. You can force the function to return only the probe sets associated with the values selected by the application of the user-specified function (specify forceProbesetSelection=TRUE). For instance, if the values in the second column of ps2 are p-values and fct=min, for each probe set in ps1 the minimal p-value associated with orthologous probe sets in ps2 is returned (ps2_value) as well as the probe set associated with the minimal p-value (ps2_probeSel).

Value

`ps2\_probeSel` A list of orthologous probe sets given by the mapping table (ps2ps) and present in ps2. By default, all orthologous probe sets are returned. If forceProbesetSelection=TRUE, only the probe sets whose values are matched after application of the user-defined function (fct) are returned.

`ps2\_value` A list of values associated with the orthologous probe sets. By default, all values are returned. If a function is specified (fct), it is applied to the values before they are returned.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

Examples

```
data(orthologs_example)

##select the first 3 probe sets listed in 'table_mouse' and their orthologs in 'table_human'
##note that no ortholog is found for the top mouse probe set
table_mouse[1:3,]
orthops<-getOrthologousProbesets(table_mouse[1:3,1],table_human,ortho)
orthops[[1]]

##the second item returned contains the values associated with orthologs (second column of 'table_human')
orthops[[2]]
```

```
##calculates, for each mouse probe set, the median orthologous log fold change (in case of multiple orthologs)
##(in this case log fold changes need to be in the second column of 'table_mouse')
orthops<-getOrthologousProbesets(table_mouse[1:3,1],table_human,ortho,'median')
orthops[[2]]

##for each mouse probe set having multiple orthologous human probe sets
##select the orthologous probe set with the smallest p-value (column 4 of 'table_human')
orthops<-getOrthologousProbesets(table_mouse[1:3,1],table_human[,c(1,4)],ortho,'min',forceProbesetSelection=TR
orthops[[1]]

##orthologous probe set selection can based on arbitrary functions
##e.g. select the 2 orthologous probe sets with the smallest p-values
orthops<-getOrthologousProbesets(table_mouse[1:3,1],table_human[,c(1,4)],ortho,function(x){sort(x)[1:2]},force
orthops[[1]]
```

`getPROBESET` *Find probe set IDs*

Description

Takes a vector of gene IDs (or identifiers of other types) and an annotation table and looks up the gene IDs in the table to retrieve the corresponding probe set identifiers. Each gene ID can occur multiple times (i.e. on mulitple lines) in the annotation table.

Usage

```
getPROBESET(geneid, annot, uniqueID = FALSE, diagnose = FALSE, idCol = 19, noPSsymbol = NA, noPSprovided
```

Arguments

`geneid`	character vector containing the gene IDs.
`annot`	annotation table (data frame) where each row is a record and each column is an annotation field.
`uniqueID`	logical. If TRUE, only probe set IDs annotated with a single gene ID are returned. If FALSE, probe set IDs annotated with multiple gene IDs are returned too.
`diagnose`	logical. If TRUE, 3 (logical) vectors used for diagnostic purpose are returned in addition to the annotation. If FALSE (default) only the annotation is returned.
`idCol`	column in annotation table containing the gene identifiers.
`noPSsymbol`	character string to be used in output list 'ps' if no probe set ID is found or provided in the annotation table.
`noPSprovidedSymbol`	character string used in annotation table and indicating missing probe set ID.

Details

This function can be used with Affymetrix annotation files (e.g. 'HG-U133_Plus_2_annot.csv'). It retrieves probe set IDs corresponding to particular gene identifiers. By default, the function takes gene IDs but any type of identifier (e.g. gene symbol) can be used (set 'idCol' accordingly).

Probe set IDs are returned as elements of list 'ps'. If multiple probe set IDs are found for 'geneid[i]', a vector containing all probe set IDs is returned as the 'i-th' element of list 'ps'.

The default values for 'idCol', 'noPSsymbol', and 'noPSprovidedSymbol' are chosen to suit the format of Affymetrix annotation files. However, options can be set to look up any annotation table, provided the probe set identifiers are in the first column.

Value

`ps`	list of length 'length(geneid)' the 'i'-th element of which contains the probe set IDs for 'geneid[i]'.
`empty`	logical vector of length 'length(geneid)'. 'empty[i]' is TRUE if 'geneid[i]' is empty or NA.
`noentry`	locial vector of length 'length(geneid)'. 'noentry[i]' is TRUE if 'geneid[i]' cannot be found in column 'idCol' (default is column 19) of the annotation table.
`noid`	locial vector of length 'length(geneid)'. 'noid[i]' is TRUE if 'ps[i]==noIDprovidedSymbol' is TRUE.

Note

`getMULTIANNOTATION` provides a more flexible solution that can be used with arbitrary annotation tables.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

See Also

`getMULTIANNOTATION`

Examples

```
##example Affymetrix annotation file and its location
annotationFile<-system.file('extdata','HG-U133_Plus_2_annot_part.csv',package='annotationTools')

##load annotation file
annotation<-read.csv(annotationFile,colClasses='character',comment.char='#')

##genes of interest
```

```
myGenes<-c('DDR1','GUCA1A','HSPA6',NA,'XYZ')

##column 15 in annotation contains gene symbols
colnames(annotation)

##find probe sets probing for particular genes
getPROBESET(myGenes,annotation,idCol=15)

##find probe sets probing only for the genes of interest (i.e. with unique annotation)
getPROBESET(myGenes,annotation,idCol=15,uniqueID=TRUE)

##track origin of annotation failure for the 2 last probe set IDs
getPROBESET(myGenes,annotation,idCol=15,diagnose=TRUE)
```

HG-U133_Plus_2_annot_part

Example Affymetrix annotation file

Description

Truncated Affymetrix annotation file for GeneChip array HG-U133 Plus 2.

Format

A table containing 7 probe set IDs and the associated annotation.

Source

http://www.affymetrix.com

HG-U133_Plus_2_ortholog_part

Example Affymetrix ortholog file

Description

Truncated Affymetrix ortholog file for GeneChip array HG-U133 Plus 2.

Format

A table containing 5 probe set IDs and their orthologs probe set IDs on other GeneChip arrays.

Source

http://www.affymetrix.com

homologene_part *Example HomoloGene file*

Description

Truncated HomoloGene database file.

Format

A table containing 3 homology clusters defined by HomoloGene.

Source

http://www.ncbi.nlm.nih.gov/HomoloGene

listToCharacterVector *Turn list into character vector*

Description

Takes a list and returns a character vector by (separately) concatenating the vectors in the list.

Usage

```
listToCharacterVector(lst,sep=' /// ')
```

Arguments

lst	list object
sep	character string specifying the separator to concatenate each elements of vectors in the list

Details

The output vector has same length as the input list.

Value

v	vector of length 'length(lst)' where the 'i'-th element contains the concatenation of the 'i'-th element of 'lst'.

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

Examples

```
##an example list
lst<-vector('list',3)
lst[[1]]<-c('a1','a2')
lst[[2]]<-'b'
lst[[3]]<-c('c1','c2','c3')
##merge first 2 list elements
listToCharacterVector(lst,sep=', ')
```

Mouse430_2_annot_part *Example Affymetrix annotation file*

Description

Truncated Affymetrix annotation file for GeneChip array Mouse 430 2.

Format

A table containing 9 probe set IDs and the associated annotation.

Source

http://www.affymetrix.com

---

ortho *ortho*

Description

Partial ortholog probe sets mapping table of MG-U74Av2 probe sets to HG-U133A probe sets. Generated with ps2ps.R and Affymetrix annotations for MG-U74Av2 and HG-U133A arrays. Used in the example cross-species analysis presented in the vignette of package annotationTools.

Format

data.frame with 100 rows and 4 columns.

Source

http://www.affymetrix.com for Affymetrix annotations

`ps2ps` *Find orthologous/homologous probe sets across two different Affymetrix microarray formats using a table of orthologs*

Description

Takes two Affymetrix annotation files, a table of orthologs (either 'gene_orthologs' or the HomoloGene database), a target species ID and returns a mapping table with homologous/orthologous probe sets.

Usage

```
ps2ps(annotation_1,annotation_2,ortholog,target_species,probesets=NULL,tableType="homologene")
```

Arguments

`annotation_1`	A data.frame with Affymetrix annotation for the source microarray format.
`annotation_2`	A data.frame with Affymetrix annotation for the target microarray format.
`ortholog`	A data.frame with the HomoloGene or 'gene_orthologs' database.
`target_species`	The target species identifier (i.e. the species corresponding to the target microarray).
`probesets`	A vector of probe set identifiers. If not specified, all probe sets on the source microarray format are mapped (default).
`tableType`	Character string (either 'homologene' or 'gene_orthologs') specifying the type of ortholog table used ('homologene' is the default).

Details

A table of orthologous/homologous probe sets is built by looking up gene IDs (corresponding to the probe sets on the source microarray array) in HomoloGene or 'gene_orthologs' to find their orthologs, and identifying probe sets in the target microarray that probe the orthologous gene transcripts.

Affymetrix annotation files can be obtained from Affymetrix (http://www.affymetrix.com). A flat file database version can be obtained from HomoloGene (http://www.ncbi.nlm.nih.gov/HomoloGene) or from NCBI's 'Orthologs from Annotation pipeline' database, referred to as 'gene_orthologs' (ftp.ncbi.nlm.nih.gov/gene/DATA/gene_orthologs.gz). Target species ID are defined by the NCBI Taxonomy database (http://www.ncbi.nlm.nih.gov/Taxonomy)

Value

`mappingTable`	A data.frame with four columns and as many rows as there are probe sets in the source annotation. Eache row corresponds to a probe set in the source annotation (column1), the corresponding gene IDs (column 2), the orthologous gene IDs in the target species (column 3), and the probe sets in the target annotation correspondin to the orthologous gene IDs (column 4).

Author(s)

Alexandre Kuhn

References

Kuhn et al. Cross-species and cross-platform gene expression studies with the Bioconductor-compliant R package 'annotationTools'. BMC Bioinformatics, 9:26 (2008)

Examples

```
## Not run:
##load Affymetrix annotations
annotMouse<-read.csv('Mouse430_2_annot.csv',colClasses='character',comment.char='#')
annotHuman<-read.csv('HG-U133A_annot.csv',colClasses='character',comment.char='#')

##load HomoloGene database
homologene<-read.delim('homologene.data',header=FALSE)

##define target species ID
homoSapiens_ID<-9609

##map all probe sets on mouse array Mouse 430 2.0 to their orthologs on human array HG-U133A
mappingTable<-ps2ps(annotMouse,annotHuman,homologene,homoSapiens_ID)

##write mapping table to disk
write.table(mappingTable,file='Mouse4302_HGU133A.txt',sep='\t',col.names=T,row.names=F,quote=FALSE)

##to map the first 10 probe sets given in the annotation only
mappingTable<-ps2ps(annotMouse,annotHuman,homologene,targetSpecies,probesets=annotMouse[1:10,1])

## End(Not run)
```

---

table_human *table_human*

Description

Partial list of probe sets from differential expression analysis of Huntington's disease patients versus controls. Used in the example cross-species analysis presented in the vignette of package annotationTools.

Format

data.frame with 188 rows (probe sets) and 12 columns.

Source

http://hdbase.org/cgi-bin/welcome.cgi

`table_mouse` *table_mouse*

Description

Top 100 probe sets from the differential expression analysis of CHL2 mice versus wild-type mice (MG-U74Av2). Used in the example cross-species analysis presented in the vignette of package annotationTools.

Format

data.frame with 100 rows and 8 columns

Source

http://hdbase.org/cgi-bin/welcome.cgi

Index

∗ **datasets**
- annot_HGU133A, 2
- gene_orthologs_part, 3
- HG-U133_Plus_2_annot_part, 21
- HG-U133_Plus_2_ortholog_part, 21
- homologene_part, 22
- Mouse430_2_annot_part, 23
- ortho, 23
- table_human, 25
- table_mouse, 26

∗ **manip**
- compactList, 2
- getANNOTATION, 4
- getGENEID, 6
- getGENEONTOLOGY, 7
- getGENESYMBOL, 9
- getGENETITLE, 11
- getHOMOLOG, 13
- getMULTIANNOTATION, 15
- getOrthologousProbesets, 17
- getPROBESET, 19
- listToCharacterVector, 22
- ps2ps, 24

annot_HGU133A, 2

compactList, 2

gene_orthologs_part, 3
getANNOTATION, 4, *7*, *9*, *11*, *12*, *16*
getGENEID, 6
getGENEONTOLOGY, 7
getGENESYMBOL, 9
getGENETITLE, 11
getHOMOLOG, 13
getMULTIANNOTATION, *5*, 15, *20*
getOrthologousProbesets, 17
getPROBESET, 19

HG-U133_Plus_2_annot_part, 21
HG-U133_Plus_2_ortholog_part, 21
homologene_part, 22

listToCharacterVector, 22

Mouse430_2_annot_part, 23

ortho, 23

ps2ps, 24

table_human, 25
table_mouse, 26