# Package 'Rarmec'

September 19, 2007

**Type** Package

**Title** R interface to ARMeC

**Version** 1.0

**Date** 2007-09-18

**Author** David Enot

**Maintainer** David Enot <dle@aber.ac.uk>

**Description** Functions to interface with ARMec.

**License** GPL version 2.

**SaveImage** no

**LazyLoad** yes

## R topics documented:



---

`Rarmec-package` *A simple R interface to query ARMeC*

---

### Description

Set of functions to query ARMeC directly from the data analysis process - This is why it is in R! This is a first version and would probably improvements.

### Details

| | |
|---|---|
| Package: | Rarmec |
| Type: | Package |
| Version: | 1.0 |
| Date: | 2007-09-18 |
| License: | GPL version 2. |

## Author(s)

David Enot

## Examples

```
### Real example using Brachypodium from the FIEmspro package
library(Rarmec)
library(FIEmspro)
library(spatstat) ### used in ar.corplot

### Select classes 1 and 2
data(abr1)
dat <- abr1$pos[,110:1000]
l=which(abr1$fact$class==1 | abr1$fact$class==2)
y   <- factor(abr1$fact$class[l])
x   <- preproc(dat[l,],y, method=c("log10","TICnorm"))

#### Use randomForest to rank variables
res=fs.rf(x,y,ntree=2000)
#### Select variables with importance score >0.003
lvar=names(which(res$stats>0.003))

#### who are they?
print(lvar)

#### Let's retrieve all possible solutions
## measured adducts is enough for the time being
all.sol<-ar.getlist(lvar,meas=T)

#### Let's dump them in a CSV file for Excel manipulation
write.table(all.sol,"tmp-allsolutions.csv",row.names=F,sep=",")

#### Perform cluster analysis
hc<-ar.clust(x,lvar,method="complete")
## plot clustering analysis result
plot(hc)

#### plot overall correlation matrix
ar.corplot(hc,x,main="")  ## main="" <-> no title

#### Retrieve candidates for those adducts have been measured
### Dissimilarity is set to 0.3 to form the clusters
cl.res=ar.splitcl(hc,all.sol,x,h=0.3)

#### Output in the console only nominal masses in each cluster
ar.writecl(cl.res,icl = NULL, cor.mat = FALSE, armec.mat = FALSE)
```

```
#### Output in the console guesses for cluster 1 and 2 except the pathway
ar.writecl(cl.res,icl = c(1,2), cor.mat = FALSE,
   armec.mat = TRUE, path=FALSE)

#### Output in the console guesses for cluster 1 with correlation matrices
## but no pathway information
ar.writecl(cl.res,icl = 1, cor.mat = TRUE, armec.mat = TRUE, path=FALSE)

#### Dump everything in file="tmp-ex-armec.csv"
ar.writecl(cl.res,file="tmp-ex-armec.csv")
```

---

`ar.clust` *Wrapper for clustering signals*

---

## Description

Wrapper for clustering a subset of variables of a matrix using `1-abs(cor)` as a dissimilarity metric. Variables entered are in a form of a list corresponding to variables names in the matrix rather than their ids.

## Usage

```
ar.clust(x, lvar=NULL, method = "complete")
```

## Arguments

| | |
|---|---|
| `x` | Original matrix |
| `lvar` | List of variables - if NULL all variables enter the clustering |
| `method` | Method to used in `hclust` |

## Value

An object of class `hclust`

## Author(s)

David Enot ⟨dle@aber.ac.uk>⟩

## See Also

`hclust`

## Examples

```
x<-USArrests
### Show variable names
dimnames(x)[[2]]

### Select some interesting ones (!?)
lvar<-c("Murder" , "Assault", "Rape")
```

```
hc<-ar.clust(x,lvar)
plot(hc)
```

---

`ar.corplot` *Plotting signals correlations*

---

## Description

Function for producing 2D plots of the actual correlation between a subset of variables contained in a `hclust`

## Usage

```
ar.corplot(mclust, x, ...)
```

## Arguments

| | |
|---|---|
| `mclust` | Object produced by `ar.clust` |
| `x` | Original matrix |
| `...` | Arguments for the function `plot` |

## Value

NULL

## Author(s)

David Enot ⟨dle@aber.ac.uk>⟩

## References

put references to the literature/web site here

## See Also

`ar.clust`, `plot`

## Examples

```
x<-USArrests
### Show variable names
dimnames(x)[[2]]

lvar<-c("Murder" , "Assault", "Rape")
hc<-ar.clust(x,lvar)

plot(hc)

### Between variable correlation plot
ar.corplot(hc,x,main="")
```

`ar.getcmp` *Retrieve ARMeC compound information*

## Description

Download and parse information related to a compound on the ARMeC server. Query the following URL:

http://www.armec.org/MetaboliteLibrary/servlet/MetaboliteSummary?metaboliteID=XXX

## Usage

```
ar.getcmp(cmpid)
```

## Arguments

| | |
|---|---|
| `cmpid` | ARMeC compound id |

## Details

If necessary, more details than the description above

## Value

Return a list comprising:

Molecular formula

Molecular weight

KEGG identifier if available

TAIR identifier if available

Metabolic pathway(s) if available

## Author(s)

David Enot ⟨dle@aber.ac.uk>⟩

## See Also

`ar.getone`

## Examples

```
### get information related to compound 896 available at:
### metaboliteID=896

ar.getcmp(896)
```

`ar.getlist` *Retrieve ionisation products from a list of signals*

## Description

Wrapper to parse variables names and retrieve all possible ionisation products corresponding to a list of nominal masses. By default, variables labels starting with P(or p) and N(or n) will be treated as positive and negative mode ionisation product. If a variable id is solely a number, search will be carried according to `ionMode`.

## Usage

```
ar.getlist(lvar, meas = TRUE, ionMode = "positive", ...)
```

## Arguments

| | |
|---|---|
| `lvar` | Describe `lvar` here |
| `meas` | Look for measured adducts - If false, return predicted adducts |
| `ionMode` | Default mode if ionisation mode is not specified in the variable name |
| `...` | arguments for `ar.getone` |

## Value

Return a matrix/vector containing information of the possible candidates:

| | |
|---|---|
| `Ion` | Nominal mass with ionisation mode id |
| `ArmecId` | ARMeC identifier |
| `Name` | ARMeC name |
| `Adduct` | ARMeC ionisation product name |
| `MFormula` | Molecular formula |
| `MWeight` | Molecular weight |
| `KeggId` | KEGG identifier |
| `TairId` | TAIR identifier |
| `Pathway` | Metabolic pathway(s) |

## Author(s)

David Enot ⟨dle@aber.ac.uk>⟩

## See Also

`ar.getone`

### Examples

```
### Looking for measured mass=193 in the positive mode
ar.getlist(c("P193","pos193",193))[,1:4]

### Looking for measured mass=193 in the negative mode
ar.getlist("193",meas=T,ionMode="negative")[,1:4]

### A mixture of both modes
ar.getlist(c("neg193","193"),meas=T,ionMode="positive")[,1:4]

### A mixture of both modes - same result as before
ar.getlist(c("neg193","193"),meas=T)[,1:4]
```

---

## ar.getone *Interface to query ARMeC*

---

### Description

Function to send, retrieve and parse one query to ARMeC. This is equivalent to fill each field of the ARMeC search page:

http://www.armec.org/MetaboliteLibrary/search$_by_e si_d ata.html$

In the context of the current package, this function is better used from `ar.getlist` where parsing actual variables with mixture of both modes is allowed.

### Usage

```
ar.getone(bioSource = "Embryophyta", ionMode = "positive", accuracy = "nominal",
```

### Arguments

| | |
|---|---|
| `bioSource` | Biological source |
| `ionMode` | positive or negative |
| `accuracy` | Nominal mass is only effective for the moment |
| `measuredMass` | Mass if measured adducts are queried |
| `predictedMass` | Mass if predicted adducts are queried |
| `anyMass` | List of masses to be queried |
| `daughterIon` | Daughter ion mass |

### Value

Return a matrix/vector containing information of the possible candidates:

| | |
|---|---|
| `ArmecId` | ARMeC identifier |
| `Name` | ARMeC name |
| `Adduct` | ARMeC ionisation product name |
| `MFormula` | Molecular formula |

| | |
|---|---|
| MWeight | Molecular weight |
| KeggId | KEGG identifier |
| TairId | TAIR identifier |
| Pathway | Metabolic pathway(s) |

### Author(s)

David Enot ⟨dle@aber.ac.uk>⟩

### See Also

hclust

### Examples

```
### Fetch compounds that have ionisation products of
### measured nominal mass=193 in the positive mode
ar.getone(measuredMass=193,ionMode="positive")

### same but predicted this time
ar.getone(predictedMass=193,ionMode="positive")
```

---

## ar.splitcl *Retrieve putative candidates from a cluster analysis*

---

### Description

Wrapper to group potential candidates by cluster analysis (ar.clust)). Clusters can be formed by either choosing a fixed number of clusters (argument k) or setting a dissimilarity threshold (argument h).

### Usage

```
ar.splitcl(hc, sol,x, k = NULL, h = 0.3, meas = TRUE)
```

### Arguments

| | |
|---|---|
| hc | Object from ar.clust |
| sol | Matrix from ar.getlist |
| x | Original data matrix |
| k | Number of clusters to be generated |
| h | Dissimilarity based threshold to define the clusters (1-correlation) |

### Value

List of clusters - Each contains

| | |
|---|---|
| cor | Correlation matrix |
| met | Characteristics of each solution as for ar.getlist |

### Author(s)

David Enot ⟨dle@aber.ac.uk>⟩

### See Also

`ar.writecl`

### Examples

```
### Check ar.writecl for complete example
```

---

## ar.writecl *Ouput for lists of putative candidates*

---

### Description

Produces output of lists of potential candidates generated from `ar.splitcl` in the console or in a file. If only one nominal mass constitutes a cluster, the correlation is not printed out. A nominal mass is not printed if its query does not produce a candidate. Nevertheless, the original list of variables is always given for each cluster.

### Usage

```
ar.writecl(res, icl = NULL, cor.mat = TRUE, armec.mat = TRUE, file = NULL, path
```

### Arguments

| | |
|---|---|
| `res` | Object from `ar.clust` |
| `icl` | List of clusters - If `NULL`, all clusters are considered |
| `cor.mat` | Should correlation matrices be printed? |
| `armec.mat` | Should candidates be printed? |
| `file` | Name of a file - `file=NULL` induces console output |
| `path` | Should pathways be printed? |

### Value

NULL

### Author(s)

David Enot ⟨dle@aber.ac.uk>⟩

### References

put references to the literature/web site here

### See Also

`ar.splitcl`,`ar.getcl`

**Examples**

```
### Extended example
```

# Index