

Package ‘DatABEL’

February 14, 2012

Type Package

Title file-based access to large matrices stored on HDD in binary format

Version 0.9-2

Date 2010-10-17

Author Yurii Aulchenko, Stepan Yakovenko

Maintainer Yurii Aulchenko <i.aoultchenko@erasmusmc.nl>

Depends R (>= 2.4.0), methods

Description a package providing interface to C++ FILEVECTOR library facilitating analysis of large (giga- to tera-bytes) matrices; matrix storage is organized in a way that either columns or rows are quickly accessible; primarily aimed to support genome-wide association analyzes e.g. using GenABEL, MixABEL and ProbABEL

License GPL (>= 2)

Repository CRAN

Date/Publication 2010-10-20 13:43:04

R topics documented:

DatABEL-package	2
apply2dfo	2
databel	4
databel-class	4
databel2matrix	6
databel2text	7
extract_text_file_columns	7
get_temporary_file_name	8
make_empty_fvf	8
matrix2databel	9
process_lm_output	10
text2databel	11

Index**14**

DatABEL-package	<i>DatABEL package for fast consecutive access to large out-of-RAM stored matrices...</i>
-----------------	---

Description

DatABEL package for fast consecutive access to large out-of-RAM stored matrices

Details

A package interfacing FILEVECTOR C++ library for storage of and fast consecutive access to large data matrices in out-of-RAM disk mode with regulated cache size. Columns of matrix are accessible very quickly.

Author(s)

Yurii Aulchenko (R code), Stepan Yakovenko & Andrey Chernyh (C++ code)

See Also

[apply2dfo](#), [databel2matrix](#), [databel2text](#), [extract_text_file_columns](#), [matrix2databel](#), [text2databel](#), [databel](#)

apply2dfo	<i>applies a function to 'databel' object...</i>
-----------	--

Description

applies a function to 'databel' object

Usage

```
apply2dfo(..., dfodata, anFUN="lm", MAR=2, procFUN, outclass="matrix",
           outfile, type="DOUBLE", transpose=TRUE)
```

Arguments

dfodata	'databel' object which is iterated over
anFUN	user-defined analysis function
MAR	which margin to iterate over (default = 2, usually these are 'columns' used to store SNP data)

procFUN	function to process the output and present that as a fixed-number-of-columns matrix or fixed-length vector. Can be missing if standard functions listed below are used. Pre-defined processors included are "process_lm_output" (can process functions "lm", "glm", "coxph") and "process_simple_output" (process output from "sum", "prod", "sum_not_NA" [no. non-missing obs], "sum_NA" [no. missing obs].)
outclass	output to ("matrix" or "databel")
outfile	if output class is "databel", the generated object is bond to the outfile
type	if output class is "databel", what data tyoe to use for storage
transpose	whether to transpose the output
...	arguments passed to the anFUN

Details

An iterator applying a user-defined function to an object of 'databel-class' object

Examples

```

unlink("tmp*")
a <- matrix(rnorm(50),10,5)
rownames(a) <- paste("id",1:10,sep="")
colnames(a) <- paste("snp",1:5,sep="")
b <- as(a,"databel")
apply(a,FUN="sum",MAR=2)
apply2dfo(SNP,dfodata=b,anFUN="sum")
tA <- apply2dfo(SNP,dfodata=b,anFUN="sum",outclass="databel",outfile="tmpA")
tA
as(tA,"matrix")
apply2dfo(SNP,dfodata=b,anFUN="sum",transpose=FALSE)
tB <- apply2dfo(SNP,dfodata=b,anFUN="sum",transpose=FALSE,outclass="databel",outfile="tmpB")
tB
as(tB,"matrix")

sex <- 1*(runif(10)>.5)
trait <- rnorm(10)+sex+as(b[,2],"vector")+as(b[,2],"vector")*sex*5
apply2dfo(trait~SNP*sex,dfodata=b,anFUN="lm")
tC <- apply2dfo(trait~SNP*sex,dfodata=b,anFUN="lm",outclass="databel",outfile="tmpC")
tC
as(tC,"matrix")
apply2dfo(trait~SNP*sex,dfodata=b,anFUN="lm",transpose=FALSE)
tD <- apply2dfo(trait~SNP*sex,dfodata=b,anFUN="lm",transpose=FALSE,outclass="databel",outfile="tmpD")
tD
as(tD,"matrix")
rm(tA,tB,tC,tD)
gc()
unlink("tmp*")

```

dabel	<i>initiates databel object...</i>
-------	------------------------------------

Description

initiates databel object

Usage

```
databel(baseobject, cachesizeMb=64, readonly=TRUE)
```

Arguments

baseobject	name of the file or databel-class object
cachesizeMb	cache size (amount of RAM) to be used
readonly	readonly flag

Details

this is a simple wrapper for "new" function creating databel object

Author(s)

Yurii Aulchenko

databel-class	<i>Class "databel"</i>
---------------	------------------------

Description

A class interfacing FILEVECTOR C++ library class FilteredMatrix for storage of and fast consecutive access to large data matrices in out-of-RAM disk mode with regulated cache size. Columns of matrix are accessible very quickly.

Objects from the Class

Objects can be created by calls of the form `new("databel", baseobject)` or `databel(baseobject)`. FILEVECTOR data are stored using files of form `BASE.fvi` (index) and `BASE.fvd` (data). "baseobject" is either the BASE name, or object of class "[databel](#)".

Slots

usedRowIndex: Object of class "integer" which rows are used

usedColIndex: Object of class "integer" which columns are used

uninames: Object of class "list", containing objects 'unique.names' – TRUE if all dimnames are unique; 'unique.colnames' – if column names are unique; 'unique.rownames' – if row names are unique

backingfilename: Object of class "character" providing BASE name

cacheSizeMb: Object of class "integer" size of cache to be used to access the data. If cache is equal to the data size, the object is stored in RAM

data: Object of class "externalptr", pointer to FilteredMatrix C++ object

Methods

[signature(x = "databel"): sub-setting object

[<- signature(x = "databel"): setting the values in the object

connect signature(object = "databel"): connects the data files to R object (calls constructor of FilteredMatrix object, selects rows and columns)

disconnect signature(object = "databel"): disconnects the data files to R object (calls destructor of FilteredMatrix object, selects rows and columns)

dim signature(x = "databel"): returns dimensions of the matrix

dimnames signature(x = "databel"): returns row and column names

dimnames<- signature(x = "databel"): sets row and column names

set_dimnames<- signature(x = "databel"): sets row and column names (these could be non-unique)

length signature(x = "databel"): returns number of elements in the matrix

save_as signature(x = "databel"): saves (a sub-set of) the object as FV-file

save_as_text signature(x = "databel"): saves (a sub-set of) the object as plain text file

backingfilename signature(object = "databel"): returns BASE FILEVECTOR file name used to store the data

cacheSizeMb signature(object = "databel"): returns the cache size used

cacheSizeMb<- signature(x = "databel"): sets new cache size

get_dimnames signature(object = "databel"): returns the names of rows and columns, which may be non-unique

set_dimnames<- signature(x = "databel"): set row and column names, which may be non-unique

setReadOnly<- signature(x = "databel"): sets ReadOnly (TRUE or FALSE) attribute to 'databel' object

Author(s)

Yurii Aulchenko, Stepan Yakovenko, Andrey Chernyh

References

<http://mga.bionet.nsc.ru/~yurii/ABEL/>

See Also

[make_empty_fvf](#), [databel](#), [matrix2databel](#)

Examples

```
showClass("databel")
```

databel2matrix	<i>converts 'databel' to matrix...</i>
----------------	--

Description

converts 'databel' to matrix

Usage

```
databel2matrix(from, rows, cols)
```

Arguments

from	'databel' matrix
rows	which rows to include
cols	which columns to include

Details

Converts regular R matrix to [databel](#) object. This is the procedure used by "as" converting to DatABEL objects, in which case a temporary file name is created

Value

object of [matrix](#) class

Author(s)

Stepan Yakovenko

databel2text	<i>Exports DatABEL object to a text file...</i>
--------------	---

Description

Exports DatABEL object to a text file

Usage

```
databel2text(databel, file, NAString="NA", row.names=TRUE,  
             col.names=TRUE, transpose=FALSE)
```

Arguments

databel	DatABEL object
file	output file name
NAString	string to replace NA with
row.names	export row names if TRUE
col.names	export col names if TRUE
transpose	whether the matrix should be transposed

Details

Exports DatABEL object to a text file

Author(s)

Stepan Yakovenko

extract_text_file_columns	<i>extract_text_file_columns</i>
---------------------------	----------------------------------

Description

extracts columns from text file

Usage

```
extract_text_file_columns(file, whichcols)
```

Arguments

file	file name
whichcols	which columns to extract

Details

Extracts a column from text file to a matrix. If in a particular file line the number of columns is less than a column specified, returns last column!

Value

matrix of strings with values from that columns

get_temporary_file_name

generates temporary file name...

Description

generates temporary file name

Usage

get_temporary_file_name(path=".", withFVext=TRUE)

Arguments

path	path to directory where the temporary file will be located
withFVext	whether function should check presence of *FVD and *FVI files too

Details

function to generate temporary file name

make_empty_fvf

makes empty filevector object...

Description

makes empty filevector object

Usage

make_empty_fvf(name, nvariables, nobservations, type="DOUBLE",
cachesizeMb=64, readonly=FALSE)

Arguments

name	name fo the file to be assoiated with new object
nvariables	number of variables (R columns)
nobservations	number of observations (R rows)
type	data type of the object ("UNSIGNED_SHORT_INT", "SHORT_INT", "UNSIGNED_INT", "INT", "FLOAT", "DOUBLE", "CHAR", "UNSIGNED_CHAR")
cacheSizeMb	what cache size to use for newly generated 'databel' object
readonly	whether to open new 'databel' in readonly mode

Details

function to generate empty filevector object (and disk files)

Value

databel object; also file is created in file system

matrix2databel	<i>converts matrix to 'databel'...</i>
----------------	--

Description

converts matrix to 'databel'

Usage

```
matrix2databel(from, filename, cacheSizeMb=64, type="DOUBLE",
               readonly=FALSE)
```

Arguments

from	R matrix
filename	which FILEVECTOR BASE file name to use
cacheSizeMb	cache size to be used when accessing the object
type	type of data to use for storage ("DOUBLE", "FLOAT", "INT", "UNSIGNED_INT", "UNSIGNED_SHORT_INT", "SHORT_INT", "CHAR", "UNSIGNED_CHAR")
readonly	whether to generate new 'databel' in read only mode

Details

Converts regular R matrix to [databel](#) object. This is the procedure used by "as" converting to DatABEL objects, in which case a temporary file name is created

Value

object of class [databel](#)

Author(s)

Yurii Aulchenko

process_lm_output *'apply2dfo'-associated functions...*

Description

'apply2dfo'-associated functions

Usage

```
process_lm_output(lmo,verbosity=2)
process_simple_output(o)
sum_not_NA(x)
sum_NA(x)
```

Arguments

lmo	object returned by analysis with "lm", "glm", etc.
verbosity	verbosity
o	output for processing
x	vector of data on which function is applied

Details

A number of functions used in conjunction with 'apply2dfo'. Standardly supported apply2dfo's anFUN analysis functions include 'lm', 'glm', 'coxph', 'sum', 'prod', "sum_not_NA" (no. non-missing obs), and "sum_NA" (no. missing obs.). Pre-defined processing functions include "process_lm_output" (can process functions "lm", "glm", "coxph") and "process_simple_output" (process output from "sum", "prod", "sum_not_NA", "sum_NA")

See Also

[apply2dfo](#)

Examples

```

a <- matrix(rnorm(50),10,5)
rownames(a) <- paste("id",1:10,sep="")
colnames(a) <- paste("snp",1:5,sep="")
b <- as(a,"databel")
apply(a,FUN="sum",MAR=2)
apply2dfo(SNP,dfodata=b,anFUN="sum",procFUN="process_simple_output")
apply2dfo(SNP,dfodata=b,anFUN="sum",transpose=FALSE)

sex <- 1*(runif(10)>.5)
trait <- rnorm(10)+sex+as(b[,2], "vector")+as(b[,2], "vector")*sex*5
apply2dfo(trait~SNP*sex,dfodata=b,anFUN="lm",procFUN="process_lm_output")

```

text2databel	<i>converts text file to filevector format...</i>
--------------	---

Description

converts text file to filevector format

Usage

```

text2databel(infile, outfile, colnames, rownames, skipcols, skiprows,
  transpose=FALSE, R_matrix=FALSE, type="DOUBLE", cachesizeMb=64,
  readonly=TRUE, naString="NA")

```

Arguments

infile	input text file name
outfile	output filevector file name; if missing, it is set to infile+".filevector"
colnames	where are the column names stored? If missing, no column names; if integer, this denotes the row of the input file where the column names are specified; if character string then the string specifies the name of the file with column names
rownames	where are the row names stored? If missing, no row names; if integer, this denotes the column of the input file where the row names are specified; if character string then the string specifies the name of the file with row names
skipcols	how many columns of the input file to skip
skiprows	how many rows of the input file to skip
transpose	whether the file is to be transposed
R_matrix	if true, the file format is assumed to follow the format of R data matrix produced with "write.table(...,col.mnames=TRUE,row.names=TRUE)"
type	data DatABEL type to use ("DOUBLE", "FLOAT", "INT", "UNSIGNED_INT", "UNSIGNED_SHORT_INT", "SHORT_INT", "CHAR", "UNSIGNED_CHAR")
cachesizeMb	cache size for the resulting 'databel-class' object
readonly	whether the resulting 'databel-class' object should be opened in readonly mode
naString	the string used for missing data (default: NA)

Details

The file provides the data to be converted to filevector format. The file may provide the data only (no row and column names) in which case col/row names may be left empty or provided in separate files (in which case it is assumed that names are provided only for the imported columns/rows – see skip-options). There is an option to skip a number of first rows and columns. The row and column names may also be provided in the file itself, in which case one needs to tell the row/column number providing column/row names. Unless option "R_matrix" is set to TRUE, it is assumed that the number of columns is always the same across the file. If above option is provided, it is assumed that both column and row names are provided in the file, and the first line contains one column less than other lines (such is the case with file produced from R using function "write.table(...,col.mnames=TRUE,row.names=TRUE)"

Value

file converted is stored in file system, [databel-class](#) object connection to the file

Author(s)

Yurii Aulchenko

Examples

```
cat("this is an example which you can run if you can write to the file system\n")

## Not run:

# create matrix
NC <- 5
NR <- 10
data <- matrix(rnorm(NC*NR),ncol=NC,nrow=NR)
rownames(data) <- paste("r",1:NR,sep="")
colnames(data) <- paste("c",1:NC,sep="")
data

# create text files
write.table(data,file="test_matrix_dimnames.dat",row.names=TRUE,col.names=TRUE,quote=FALSE)
write.table(data,file="test_matrix_colnames.dat",row.names=FALSE,col.names=TRUE,quote=FALSE)
write.table(data,file="test_matrix_rownames.dat",row.names=TRUE,col.names=FALSE,quote=FALSE)
write.table(data,file="test_matrix_NOnames.dat",row.names=FALSE,col.names=FALSE,quote=FALSE)
write(colnames(data),file="test_matrix.colnames")
write(rownames(data),file="test_matrix.rownames")

# generate identical data
text2databel(infile="test_matrix_dimnames.dat",outfile="test_matrix_dimnames",R_matrix=TRUE)
x <- databel("test_matrix_dimnames")
data <- as(x,"matrix")
data

# convert text two filevector format

text2databel(infile="test_matrix_NOnames.dat",outfile="test_matrix_NOnames.fvf",
```

```

colnames="test_matrix.colnames",rownames="test_matrix.rownames")
x <- databel("test_matrix_N0names.fvf")
if (!identical(data,as(x,"matrix"))) stop("not identical data")

text2databel(infile="test_matrix_N0names.dat",outfile="test_matrix_N0names_T.fvf",
colnames="test_matrix.colnames",rownames="test_matrix.rownames",transpose=TRUE)
x <- databel("test_matrix_N0names_T.fvf")
if (!identical(data,t(as(x,"matrix")))) stop("not identical data")

text2databel(infile="test_matrix_rownames.dat",outfile="test_matrix_rownames.fvf",
rownames=1,colnames="test_matrix.colnames")
x <- databel("test_matrix_rownames.fvf")
if (!identical(data,as(x,"matrix"))) stop("not identical data")

text2databel(infile="test_matrix_colnames.dat",outfile="test_matrix_colnames.fvf",
colnames=1,rownames="test_matrix.rownames")
x <- databel("test_matrix_colnames.fvf")
if (!identical(data,as(x,"matrix"))) stop("not identical data")

text2databel(infile="test_matrix_dimnames.dat",outfile="test_matrix_dimnames.fvf",R_matrix=TRUE)
x <- databel("test_matrix_dimnames.fvf")
if (!identical(data,as(x,"matrix"))) stop("not identical data")

# stupid extended matrix in non-R format
newmat <- matrix(-100,ncol=NC+3,nr=NR+2)
newmat[3:(NR+2),4:(NC+3)] <- data
newmat[2,4:(NC+3)] <- paste("c",1:NC,sep="")
newmat[3:(NR+2),3] <- paste("r",1:NR,sep="")
newmat
write.table(newmat,file="test_matrix_strange.dat",col.names=FALSE,row.names=FALSE,quote=FALSE)

text2databel(infile="test_matrix_strange.dat",outfile="test_matrix_strange.fvf",
colnames=2,rownames=3)
x <- databel("test_matrix_strange.fvf")
if (!identical(data,as(x,"matrix"))) stop("not identical data")

## End(Not run)

```

Index

*Topic **classes**

databel-class, 4
[, databel-method (databel-class), 4
[<-, databel-method (databel-class), 4
apply2dfo, 2, 2, 10
backingfilename (databel-class), 4
backingfilename, databel-method
(databel-class), 4
cachesizeMb (databel-class), 4
cachesizeMb, databel-method
(databel-class), 4
cachesizeMb<- (databel-class), 4
cachesizeMb<-, databel-method
(databel-class), 4
connect (databel-class), 4
connect, databel-method (databel-class),
4
databel, 2, 4, 4, 6, 9, 10
databel-class, 4, 4, 12
DatABEL-package, 2
databel2matrix, 2, 6
databel2text, 2, 7
dim, databel-method (databel-class), 4
dimnames, databel-method
(databel-class), 4
dimnames<-, databel-method
(databel-class), 4
disconnect (databel-class), 4
disconnect, databel-method
(databel-class), 4
extract_text_file_columns, 2, 7
get_dimnames (databel-class), 4
get_dimnames, databel-method
(databel-class), 4
get_temporary_file_name, 8
length, databel-method (databel-class), 4
make_empty_fvf, 6, 8
matrix, 6
matrix2databel, 2, 6, 9
process_lm_output, 10
process_simple_output
(process_lm_output), 10
save_as (databel-class), 4
save_as, databel-method (databel-class),
4
save_as_text (databel-class), 4
save_as_text, databel-method
(databel-class), 4
set_dimnames<- (databel-class), 4
set_dimnames<-, databel-method
(databel-class), 4
setReadOnly<- (databel-class), 4
setReadOnly<-, databel-method
(databel-class), 4
sum_NA (process_lm_output), 10
sum_not_NA (process_lm_output), 10
text2databel, 2, 11