Package 'tsvio'

May 8, 2026

Title Simple Utilities for Tab-Separated-Value (TSV) Files

Version 1.0.6

Depends R (>= 3.4.0)

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Description Utilities for rapidly loading specified rows and/or columns of data from large tab-separated value (tsv) files (large: e.g. 1 GB file of 10000 x 10000 matrix). 'tsvio' is an R wrapper to 'C' code that creates an index file for the rows of the tsv file, and uses that index file to collect rows and/or columns from the tsv file without reading the whole file into memory.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/MD-Anderson-Bioinformatics/tsvio

NeedsCompilation yes

LazyLoad Yes

Collate 'interface.R'

Config/testthat/edition 3

VignetteBuilder knitr

RoxygenNote 7.2.3

Author Bradley M Broom [aut] (ORCID: <https://orcid.org/0000-0002-0915-3164>),
Mary A Rohrdanz [ctb, cre],
Chris Wakefield [ctb],
James Melott [ctb],
Paul Hsieh [ctb],
MD Anderson Cancer Center [cph]

Maintainer Mary A Rohrdanz <marohrdanz@mdanderson.org>

Repository CRAN

Date/Publication 2023-11-28 17:00:02 UTC

Contents

`tsvio-package` *tsvio: Simple Utilities for Tab Separated Value (TSV) Files*

Description

Simple Utilities for Tab Separated Value (TSV) Files

Details

Utilities for indexing and rapidly loading (subsets of) data from (large) tab separated value (TSV) files. The TSV files are required to have a unique row label in the first column of each line and a unique column label in the first line of the file. Files may be formatted in either spreadsheet/Unix format (same number of fields on each line) or R format (one less column on the first line only). The data matrix in the files are expected to have the same data types in all entries. (The row and column labels are always expected to be strings.)

See Also

tsvGenIndex, tsvGetLines, tsvGetData

`tsvGenIndex` *Produce a simple index of a tsv file.*

Description

This function reads a TSV file and produces an index to the start of each row. The TSV file is required to have a header line and at least one data line. The header line may contain either the same number or one fewer columns than the data lines, which must all contain the same number of columns. The first column of each data line will be indexed.

Usage

```
tsvGenIndex(filename, indexfile)
```

Arguments

`filename`	The name (and path) of the file(s) containing the data to index.
`indexfile`	The name (and path) of the file(s) to which the index will be written. There must be exactly one index file for every filename.

Value

NULL. This function generates an index file.

See Also

tsvGetLines, tsvGetData

Examples

```
datafile = tempfile("data");
df <- data.frame(C1 = c("Foo", "Boing", "The"), C2 = c("Bar", "Boing", "End"));
rownames(df) <- c("R1", "R2", "R3");
write.table(df, file=datafile, sep="\t", quote=FALSE, row.names=TRUE, col.names=TRUE);
indexfile = tempfile("index");
tsvGenIndex (datafile, indexfile)
```

---

tsvGetData *Read matching lines from a tsv file, using a pre-computed index file.*

Description

This function reads lines that match the given patterns from a TSV file with the assistance of a pre-computed index file to the start of each row.

Usage

```
tsvGetData(
  filename,
  indexfile,
  rowpatterns,
  colpatterns,
  dtype = "",
  findany = TRUE
)
```

Arguments

filename	The name (and path) of the file containing the data to index.
indexfile	The name (and path) of the file to which the index will be written.
rowpatterns	A vector of strings containing the string to match against the index entries. Only lines with keys that exactly match at least one pattern string are returned. If rowpatterns is NULL, data from all rows is returned.
colpatterns	A vector of strings to match against the column headers in the first row
dtype	A prototype element that specifies by example the type of matrix to return. The value of the parameter is ignored. Accepted types are string (default), numeric (float), and integer.

findany	If false, all patterns must be matched. If true (default) at least one pattern must match.

Details

The index file must have been created by tsvGenIndex and the data file must not have changed since the index file was created.

Value

A matrix containing one row for each matched line and one column for each matched column.

See Also

tsvGenIndex, tsvGetLines

Examples

```
datafile = tempfile("data");
df <- data.frame(C1 = c("Foo", "Boing", "The"), C2 = c("Bar", "Boing", "End"));
rownames(df) <- c("R1", "R2", "R3");
write.table(df, file=datafile, sep="\t", quote=FALSE, row.names=TRUE, col.names=TRUE);
indexfile = tempfile("index");
tsvGenIndex (datafile, indexfile);
tsvGetData (datafile, indexfile, c("R1", "R3"), c('C2'))
```

---

tsvGetLines *Read matching lines from a tsv file, using a pre-computed index file.*

Description

This function reads lines that match the given patterns from a TSV file with the assistance of a pre-computed index file to the start of each row.

Usage

```
tsvGetLines(filename, indexfile, patterns, findany = TRUE)
```

Arguments

filename	The name (and path) of the file containing the data to index.
indexfile	The name (and path) of the file to which the index will be written.
patterns	A vector of strings containing the string to match against the index entries. Only lines with keys that exactly match at least one pattern string are returned.
findany	If false, all patterns must be matched. If true (default) at least one pattern must match.

Details

The index file must have been created by tsvGenIndex and the data file must not have changed since the index file was created.

Value

A string vector whose first element is the first line from data file. Subsequent elements of the vector are lines from the data file whose labels match an entry in patterns.

See Also

tsvGenIndex, tscGetData

Examples

```
datafile = tempfile("data");
df <- data.frame(C1 = c("Foo", "Boing", "The"), C2 = c("Bar", "Boing", "End"));
rownames(df) <- c("R1", "R2", "R3");
write.table(df, file=datafile, sep="\t", quote=FALSE, row.names=TRUE, col.names=TRUE);
indexfile = tempfile("index");
tsvGenIndex (datafile, indexfile);
tsvGetLines (datafile, indexfile, c("R1", "R3"))
```

Index