

Package ‘read.gt3x’

July 23, 2025

Type Package

Title Parse 'ActiGraph' 'GT3X'/'GT3X+' 'Accelerometer' Data

Version 1.2.0

Description Implements a high performance C++ parser for 'ActiGraph' 'GT3X'/'GT3X+' data format (with extension '.gt3x') for 'accelerometer' samples. Activity samples can be easily read into a matrix or data.frame. This allows for storing the raw 'accelerometer' samples in the original binary format to reserve space.

License EUPL

Encoding UTF-8

LinkingTo Rcpp

Imports Rcpp, utils, R.utils, tools

RoxygenNote 7.1.2

Suggests knitr, rmarkdown, testthat (>= 2.1.0), data.table, zoo, readr, lubridate, zip

VignetteBuilder knitr

NeedsCompilation yes

Author Tuomo Nieminen [aut, cre],
John Muschelli [aut] (ORCID: <<https://orcid.org/0000-0001-6469-1750>>),
Patrick Bos [ctb] (ORCID: <<https://orcid.org/0000-0002-6033-960X>>),
Vincent van Hees [ctb]

Maintainer Tuomo Nieminen <tuomo.a.nieminen@gmail.com>

Repository CRAN

Date/Publication 2022-06-30 07:40:02 UTC

Contents

as.data.frame.activity	2
get_n_samples	3
gt3x_datapath	3
gt3x_dataurl	4

gt3x_download	5
gt3x_url	5
is_gt3x	6
list_gt3x	7
parseActivityBin	7
parseGT3X	8
parseLuxBin	9
parse_gt3x_info	10
print.activity_df	11
print.gt3x_info	11
read.gt3x	12
ticks2datetime	14
unzip.gt3x	15
unzip_single_gt3x	16
Index	18

as.data.frame.activity
<i>Convert an activity matrix to a data.frame</i>

Description

Convert an activity matrix to a data.frame

Usage

```
## S3 method for class 'activity'  
as.data.frame(x, ..., verbose = FALSE, add_light = FALSE)
```

Arguments

x	Object of class 'activity' (returned by read.gt3x)
...	not used
verbose	print diagnostic messages
add_light	add light data to the data.frame if data exists in the GT3X

Value

An object of class activity_df which is also a data.frame with the following attributes (and more)

- subject_name : Subject name from info file
- time_zone : Time zone from info file
- missingness : Data frame with timestamps and the number of missing values associated.

See Also

Other gt3x-parsers: [parse_gt3x_info\(\)](#), [print.gt3x_info\(\)](#), [read.gt3x\(\)](#)

get_n_samples	<i>Calculate the expected activity sample size from start time and last sample time in the info.txt of a gt3x directory</i>
---------------	---

Description

Calculate the expected activity sample size from start time and last sample time in the info.txt of a gt3x directory

Usage

```
get_n_samples(x)
```

Arguments

x	info out from parse_gt3x_info
---	---

See Also

Other gt3x-utils: [is_gt3x\(\)](#), [ticks2datetime\(\)](#)

gt3x_datapath	<i>Path to read.gt3x package sample data</i>
---------------	--

Description

Path to read.gt3x package sample data

Usage

```
gt3x_datapath(index = NULL, verbose = TRUE)
```

```
gt3x_filename(index = NULL, zipped = FALSE)
```

Arguments

index	Integer. The index of a sample file to retrieve. If NULL (default) the path to the directory including the sample files will be returned.
verbose	print diagnostic messages
zipped	do the files have a .zip extension

Value

Character vector of files

See Also

Other file manipulations: [is_gt3x\(\)](#), [list_gt3x\(\)](#), [unzip.gt3x\(\)](#)

Examples

```
## Not run:
dir <- gt3x_datapath()
gt3x_filename <- gt3x_datapath(1)
stopifnot(!is.na(gt3x_datapath(2)))

## End(Not run)

testthat::expect_error(gt3x_filename(100))
testthat::expect_error(gt3x_filename(0))
```

gt3x_dataurl	<i>Get url of github release</i>
--------------	----------------------------------

Description

Get url of github release

Usage

```
gt3x_dataurl(
  version = "v1.0",
  baseurl = "https://github.com/THLfi/read.gt3x/releases/download"
)
```

Arguments

version	release version
baseurl	URL for GitHub release

Value

URL to file

See Also

Other sample-data: [gt3x_download\(\)](#), [gt3x_url\(\)](#)

gt3x_download	<i>Download and unzip a zipped gt3x file</i>
---------------	--

Description

Download and unzip a zipped gt3x file

Usage

```
gt3x_download(url, exdir, verbose = TRUE)
```

Arguments

url	url of the file to download
exdir	directory to extract the zip file
verbose	print diagnostic messages

Value

file path of exdir

See Also

Other sample-data: [gt3x_dataurl\(\)](#), [gt3x_url\(\)](#)

gt3x_url	<i>Get url of gt3x sample file</i>
----------	------------------------------------

Description

Get url of gt3x sample file

Usage

```
gt3x_url(index = NULL, filename = NULL)
```

Arguments

index	The index of a sample file to retrieve, passed to gt3x_filename
filename	file to grab to make url

Value

file path

See Also

Other sample-data: [gt3x_dataurl\(\)](#), [gt3x_download\(\)](#)

`is_gt3x`*Check if files are .gt3x files*

Description

Check if files are .gt3x files

Check if a .gt3x file or unzipped gt3x directory has both log.bin and info.txt

Usage

```
is_gt3x(path)
```

```
have_log_and_info(path, verbose = TRUE)
```

Arguments

path	Path(s) to file(s)
------	--------------------

verbose	print diagnostic messages
---------	---------------------------

Details

Checks if files have a .gt3x file extension

Value

Logical vector of the same length as path, which is TRUE if the corresponding path is a .gt3x file.

See Also

Other file manipulations: [gt3x_datapath\(\)](#), [list_gt3x\(\)](#), [unzip.gt3x\(\)](#)

Other gt3x-utils: [get_n_samples\(\)](#), [ticks2datetime\(\)](#)

Examples

```
is_gt3x("test.gt3x") # TRUE
is_gt3x("test") # FALSE
is_gt3x(NULL)
```

```
have_log_and_info(tempfile(), verbose = TRUE)
```

list_gt3x	<i>List full paths to all gt3x files in a directory</i>
-----------	---

Description

List full paths to all gt3x files in a directory

Usage

```
list_gt3x(path)
```

Arguments

path	Path(s) to file(s)
------	--------------------

See Also

Other file manipulations: [gt3x_datapath\(\)](#), [is_gt3x\(\)](#), [unzip.gt3x\(\)](#)

Examples

```
path <-  
  system.file(  
    "extdata",  
    package = "read.gt3x")  
list_gt3x(path)  
## Not run:  
list_gt3x(gt3x_datapath())  
  
## End(Not run)
```

parseActivityBin	<i>Parse activity samples from a NHANES-GT3X file</i>
------------------	---

Description

Parse activity samples from a NHANES-GT3X file

Usage

```
parseActivityBin(  
  filename,  
  max_samples,  
  scale_factor,  
  sample_rate,  
  verbose = FALSE,  
  debug = FALSE  
)
```

Arguments

filename	path to a activity.bin file inside the unzipped gt3x folder, which contains the activity samples
max_samples	Maximum number of rows to parse. The returned matrix will always contain this number of rows, having zeroes if not data is found.
scale_factor	Scale factor for the activity samples.
sample_rate	sampling rate for activity samples.
verbose	Print the parameters from the activity.bin file and other messages?
debug	Print information for every activity second

Value

Returns a matrix with max_samples rows and 3 columns, where the first 3 columns are the acceleration samples and the last column is timestamps in seconds (including hundredth of seconds) starting from 00:00:00 1970-01-01 UTC (UNIX time)

parseGT3X	<i>Parse activity samples from a GT3X file</i>
-----------	--

Description

Parse activity samples from a GT3X file

Usage

```
parseGT3X(  
  filename,  
  max_samples,  
  scale_factor,  
  sample_rate,  
  start_time,  
  batch_begin = 0L,  
  batch_end = 0L,  
  verbose = FALSE,  
  debug = FALSE,  
  impute_zeroes = FALSE  
)
```

Arguments

filename	(char*) path to a log.bin file inside the unzipped gt3x folder, which contains the activity samples
max_samples	Maximum number of rows to parse. The returned matrix will always contain this number of rows, having zeroes if not data is found.
scale_factor	Scale factor for the activity samples.

sample_rate	sampling rate for activity samples.
start_time	starting time of the sample recording.
batch_begin	first second in time relative to start of raw non-imputed recording to include in this batch
batch_end	last second in time relative to start of raw non-imputed recording to include in this batch
verbose	Print the parameters from the log.bin file and other messages?
debug	Print information for every activity second
impute_zeroes	Impute zeros in case there are missingness?

Value

Returns a matrix with max_samples rows and 3 columns with the acceleration samples. The matrix has attributes "time_index", "missingness", "start_time_log", "sample_rate", "impute_zeroes".

parseLuxBin	<i>Parse activity samples from a GT3X file</i>
-------------	--

Description

Parse activity samples from a GT3X file

Usage

```
parseLuxBin(filename, max_samples, scale_factor, max_value, verbose = FALSE)
```

Arguments

filename	(char*) path to a log.bin file inside the unzipped gt3x folder, which contains the activity samples
max_samples	Maximum number of rows to parse. The returned matrix will always contain this number of rows, having zeroes if not data is found.
scale_factor	Scale factor for the activity samples.
max_value	Maximum value to truncate
verbose	Print the parameters from the log.bin file and other messages?

Value

Returns a vector with max_samples elements

parse_gt3x_info	<i>Parse GT3X info.txt file</i>
-----------------	---------------------------------

Description

Parse GT3X info.txt file

Usage

```
parse_gt3x_info(path, tz = "GMT")  
extract_gt3x_info(path, tz = "GMT")
```

Arguments

path	Path to a .gt3x file or an unzipped gt3x directory
tz	timezone, passed to ticks2datetime

Note

The input for `parse_gt3x_info` is a gt3x file, but the path for `extract_gt3x_info` is the `info.txt` file, which can also pass in a connection

See Also

Other gt3x-parsers: [as.data.frame.activity\(\)](#), [print.gt3x_info\(\)](#), [read.gt3x\(\)](#)

Examples

```
gt3xfile <-  
  system.file(  
    "extdata", "TAS1H30182785_2019-09-17.gt3x",  
    package = "read.gt3x")  
parse_gt3x_info(gt3xfile)  
  
## Not run:  
gt3xfile <- gt3x_datapath(1)  
parse_gt3x_info(gt3xfile)  
  
## End(Not run)
```

print.activity_df	<i>Print the contents of the activity data</i>
-------------------	--

Description

Print the contents of the activity data

Usage

```
## S3 method for class 'activity_df'  
print(x, ...)
```

```
## S3 method for class 'activity_df'  
head(x, ...)
```

```
## S3 method for class 'activity'  
print(x, ...)
```

```
## S3 method for class 'activity'  
head(x, ...)
```

Arguments

x	gt3x_info object returned by parse_gt3x_info()
...	additional arguments passed to head

print.gt3x_info	<i>Print the contents of the info.txt file in a gt3x folder</i>
-----------------	---

Description

Print the contents of the info.txt file in a gt3x folder

Usage

```
## S3 method for class 'gt3x_info'  
print(x, ...)
```

Arguments

x	gt3x_info object returned by parse_gt3x_info()
...	not used

See Also

Other gt3x-parsers: [as.data.frame.activity\(\)](#), [parse_gt3x_info\(\)](#), [read.gt3x\(\)](#)

read.gt3x

*Read GT3X***Description**

Read activity samples from a GT3X file as a matrix. Please note that all timestamps are in local time (of the device) even though they are represented as POSIXct with GMT timezone.

Usage

```
read.gt3x(
  path,
  verbose = FALSE,
  asDataFrame = FALSE,
  imputeZeroes = FALSE,
  flag_idle_sleep = FALSE,
  cleanup = FALSE,
  ...,
  add_light = FALSE
)
```

Arguments

path	Path to gt3x folder
verbose	print diagnostic messages
asDataFrame	convert to an activity_df, see as.data.frame.activity
imputeZeroes	Impute zeros in case there are missingness? Default is FALSE, in which case the time series will be incomplete in case there is missingness.
flag_idle_sleep	flag idle sleep mode. If imputeZeroes = TRUE, this finds where all 3 axes are zero.
cleanup	should any unzipped files be deleted?
...	additional arguments to pass to parseGT3X C++ code, e.g. batch-loading options as now documented in vignette "Batch loading a gt3x file"
add_light	add light data to the data.frame if data exists in the GT3X

Value

A numeric matrix with 3 columns (X, Y, Z) and the following attributes:

- start_time : Start time from info file in POSIXct format.
- subject_name : Subject name from info file
- time_zone : Time zone from info file
- missingness : Named integer vector. Names are POSIXct timestamps and values are the number of missing values.

Note

The timestamps in the .gt3x data format are saved in .NET format, which is nanoseconds in local time since 0001-01-01. This is a bit tricky to parse into an R datetime format. DateTimes are therefore represented as POSIXct format with the 'GMT' timezone attribute, which is false; the datetime actually represents local time.

See Also

Other gt3x-parsers: [as.data.frame.activity\(\)](#), [parse_gt3x_info\(\)](#), [print.gt3x_info\(\)](#)

Examples

```
gt3xfile <-
  system.file(
    "extdata", "TAS1H30182785_2019-09-17.gt3x",
    package = "read.gt3x")
is_gt3x(gt3xfile)
have_log_and_info(gt3xfile, verbose = TRUE)

x <- read.gt3x(gt3xfile, imputeZeroes = FALSE, asDataFrame = FALSE,
  verbose = TRUE)
attr(x, "features")
df2 <- as.data.frame(x, verbose = TRUE)
attr(df2, "features")
head(df2)
rm(x); gc(); gc()
rm(df2); gc()

x <- read.gt3x(gt3xfile, imputeZeroes = TRUE, asDataFrame = TRUE,
  verbose = TRUE)

## Not run:
# first unzip, then read
datadir <- gt3x_datapath()
gt3xfolders <- unzip.gt3x(datadir)
gt3xfile <- gt3xfolders[2]
# temporary unzip, read, convert to a data frame
gt3xfile <- gt3x_datapath(1)
memory.limit()
df <- read.gt3x(gt3xfile, asDataFrame = FALSE, verbose = 2)
head(df)
rm(df); gc(); gc()

df <- read.gt3x(gt3xfile, asDataFrame = TRUE, verbose = 2)
head(df)

## End(Not run)

## Not run:
```

```

url <- paste0("https://github.com/THLfi/read.gt3x/",
"files/", "3522749/", "GT3X%2B.01.day.gt3x.zip")
destfile <- tempfile(fileext = ".zip")
dl <- download.file(url, destfile = destfile, mode = "wb")
gt3x_file <- unzip(destfile, exdir = tempdir())
gt3x_file <- gt3x_file[!grepl("__MACOSX", gt3x_file)]
path <- gt3x_file

res <- read.gt3x(path)

gz <- R.utils::gzip(path, remove = FALSE, overwrite = FALSE)
df2 <- read.gt3x(gz, asDataFrame = FALSE, verbose = 2)
head(df2)

rm(df2); gc(); gc()

## End(Not run)

```

ticks2datetime

Convert NET ticks to POSIXct datetime

Description

Convert NET ticks to POSIXct datetime

Usage

```
ticks2datetime(ticks, tz = "GMT")
```

```
datetime2ticks(x)
```

Arguments

ticks	values in NET ticks format
tz	timezone, passed to as.POSIXct
x	values in date-time format coerced to ticks

Details

reference: <https://stackoverflow.com/questions/35240874/r-net-ticks-to-timestamp-in-r>

See Also

Other gt3x-utils: [get_n_samples\(\)](#), [is_gt3x\(\)](#)

Other gt3x-utils: [get_n_samples\(\)](#), [is_gt3x\(\)](#)

Examples

```

mystr = "599633592352500000"
x = read.gt3x::ticks2datetime(mystr)
x
out = read.gt3x::datetime2ticks(as.POSIXct("1901-03-02 08:40:35.25", tz = "UTC"))
out = as.character(out)
out
stopifnot(out == mystr)
read.gt3x::datetime2ticks(x = as.POSIXct(Sys.time(), tz = "EST"))

```

unzip.gt3x

Unzip gt3x files

Description

unzip.gt3x() makes it convenient to unzip multiple .gt3x files.

Usage

```
unzip.gt3x(path, verbose = TRUE, ...)
```

Arguments

path	One of the following: (1) A path to a directory with .gt3x files in which case they are all unzipped, or (2) A character vector of direct paths to .gt3x files.
verbose	print diagnostic messages
...	arguments to pass to unzip_single_gt3x

Details

A .gt3x file is a zipped directory with two files: log.bin and info.txt. This function simply unzips the contents of the directories.

Value

Returns a vector of paths to unzipped gt3x folders.

See Also

Other file manipulations: [gt3x_datapath\(\)](#), [is_gt3x\(\)](#), [list_gt3x\(\)](#)

Examples

```

gt3xfile <-
  system.file(
    "extdata", "TAS1H30182785_2019-09-17.gt3x",
    package = "read.gt3x")
gt3xdirs <- unzip.gt3x(gt3xfile)
## Not run:
# unzip a single .gt3x file
path <- gt3x_datapath(1)
gt3xdir <- unzip.gt3x(path)

# unzip multiple .gt3x files
dir <- gt3x_datapath()
gt3xdirs <- unzip.gt3x(dir)

## End(Not run)
tfile = tempfile()
testthat::expect_error(unzip.gt3x(c(dir, tfile)))
testthat::expect_error(unzip.gt3x(c("", "")))

```

unzip_single_gt3x	<i>Unzip a single gt3x file</i>
-------------------	---------------------------------

Description

A .gt3x file is a zipped archive with two files: log.bin and info.txt. This function unzips the contents of the archive to a single folder. This is a helper for unzip.gt3x()

Usage

```

unzip_single_gt3x(
  path,
  dirname = basename(gsub(".gt3x$| ", "", path)),
  location = tempdir(),
  files = c("info.txt", "log.bin"),
  remove_original = FALSE,
  check_structure = TRUE,
  verbose = TRUE
)

```

Arguments

path	Path to a .gt3x file
dirname	The name of the resulting directory where the content of path are extracted. Default is the name of the input file, sans the .gt3x extension.
location	A path to an output directory. Default is a tempdir.

<code>files</code>	The names of files to extract. Default is <code>info.txt</code> and <code>log.bin</code>
<code>remove_original</code>	Remove the zip file after unzipping?
<code>check_structure</code>	check to see if the structure is right for the file
<code>verbose</code>	print diagnostic messages

Index

- * **file manipulations**
 - gt3x_datapath, 3
 - is_gt3x, 6
 - list_gt3x, 7
 - unzip.gt3x, 15
- * **gt3x-parsers**
 - as.data.frame.activity, 2
 - parse_gt3x_info, 10
 - print.gt3x_info, 11
 - read.gt3x, 12
- * **gt3x-utils**
 - get_n_samples, 3
 - is_gt3x, 6
 - ticks2datetime, 14
- * **sample-data**
 - gt3x_dataurl, 4
 - gt3x_download, 5
 - gt3x_url, 5

as.data.frame.activity, 2, 10, 11, 13

as.POSIXct, 14

datetime2ticks (ticks2datetime), 14

extract_gt3x_info (parse_gt3x_info), 10

get_n_samples, 3, 6, 14

gt3x_datapath, 3, 6, 7, 15

gt3x_dataurl, 4, 5

gt3x_download, 4, 5, 5

gt3x_filename, 5

gt3x_filename (gt3x_datapath), 3

gt3x_url, 4, 5, 5

have_log_and_info (is_gt3x), 6

head, 11

head.activity (print.activity_df), 11

head.activity_df (print.activity_df), 11

is_gt3x, 3, 4, 6, 7, 14, 15

list_gt3x, 4, 6, 7, 15

parse_gt3x_info, 2, 3, 10, 11, 13

parseActivityBin, 7

parseGT3X, 8

parseLuxBin, 9

print.activity (print.activity_df), 11

print.activity_df, 11

print.gt3x_info, 2, 10, 11, 13

read.gt3x, 2, 10, 11, 12

ticks2datetime, 3, 6, 10, 14

unzip.gt3x, 4, 6, 7, 15

unzip_single_gt3x, 15, 16