Package 'rlas'

June 27, 2017

Type Package

Title Read and Write 'las' and 'laz' Binary File Formats Used for Remote Sensing Data

Version 1.1.4

Date 2017-06-26

Description Read and write 'las' and 'laz' binary file formats. The LAS file format is a public file format for the interchange of 3-dimensional point cloud data between data users. The LAS specifications are approved by the American Society for Photogrammetry and Remote Sensing. The LAZ file format is an open and lossless compression scheme for binary LAS format versions 1.0 to 1.3.

URL https://github.com/Jean-Romain/rlas

BugReports https://github.com/Jean-Romain/rlas/issues

License GPL-3

Depends R (>= 3.0.0)

Imports data.table,Rcpp

Suggests lidR

LazyData true

RoxygenNote 6.0.1

LinkingTo Rcpp

NeedsCompilation yes

Author Jean-Romain Roussel [aut, cre, cph],
Martin Isenburg [cph] (Is the author of the LASlib and LASzip libraries),
David Auty [ctb] (Reviewed the documentation),
Pierrick Marie [ctb] (Helped to compile LASlib code in R),
Florian De Boissieu [ctb] (Enable support of .lax file)

Maintainer Jean-Romain Roussel <jean-romain.roussel.1@ulaval.ca>

Repository CRAN

Date/Publication 2017-06-27 06:36:03 UTC

R topics documented:

`readlasdata` *Read data from a .las or .laz file*

Description

Reads data from .las or .laz files in format 1 to 4 according to LAS specifications and returns a `data.table` labeled according to LAS specifications. See the ASPRS documentation for the LAS file format. The optional logical parameters enables the user to save memory by choosing to load only the fields they need. Indeed, data is loaded into the computer's memory (RAM) suboptimally because R does not accommodate many different data types. Moreover, the function provides a streaming filter to load only the points of interest into the memory without allocating any superfluous memory.

Usage

```
readlasdata(file, Intensity = TRUE, ReturnNumber = TRUE,
  NumberOfReturns = TRUE, ScanDirectionFlag = TRUE,
  EdgeOfFlightline = TRUE, Classification = TRUE, ScanAngle = TRUE,
  UserData = TRUE, PointSourceID = TRUE, RGB = TRUE, filter = "")
```

Arguments

`file`	filepath character string to the .las or .laz file
`Intensity`	logical. do you want to load the Intensity field? default: TRUE
`ReturnNumber`	logical. do you want to load the ReturnNumber field? default: TRUE
`NumberOfReturns`	logical. do you want to load the NumberOfReturns field? default: TRUE
`ScanDirectionFlag`	logical. do you want to load the ScanDirectionFlag field? default: TRUE
`EdgeOfFlightline`	logical. do you want to load the EdgeOfFlightline field? default: TRUE
`Classification`	logical. do you want to load the Classification field? default: TRUE
`ScanAngle`	logical. do you want to load the ScanAngle field? default: TRUE
`UserData`	logical. do you want to load the UserData field? default: TRUE
`PointSourceID`	logical. do you want to load the PointSourceID field? default: TRUE
`RGB`	logical. do you want to load R,G and B fields? default: TRUE
`filter`	character. filter data while reading the file (streaming filter) without allocating any useless memory. (see Details).

Details

Because rlas relies on the well-known LASlib library written by Martin Isenburg to read the binary files, the package also inherits the filter commands available in LAStools. To use these filters the user can pass the common commands from LAStools into the parameter 'filter'. Type rlas:::lasfilterusage() to display the LASlib documentation and the available filters.

The filter works in two passes. First it streams the file without loading anything and counts the number of points of interest. Then it allocates the necessary amount of memory and reads the file a second time, and stores the points of interest in the computer's memory (RAM).

Value

A data.table

See Also

Other rlas: readlasheader, writelas

Examples

```
lazfile <- system.file("extdata", "example.laz", package="rlas")

lasdata <- readlasdata(lazfile)
lasdata <- readlasdata(lazfile, filter = "-keep_first")
lasdata <- readlasdata(lazfile, filter = "-drop_intensity_below 80")
```

---

readlasheader *Read header from a .las or .laz file*

Description

Reads header from .las or .laz files in format 1 to 4 according to LAS specifications and returns a list labeled according to LAS specifications. See the ASPRS documentation for the LAS file format.

Usage

```
readlasheader(file)
```

Arguments

file	filepath character string to the .las or .laz file

Value

A list

See Also

Other rlas: readlasdata, writelas

Examples

```
lazfile   <- system.file("extdata", "example.laz", package="rlas")
lasheader <- readlasheader(lazfile)
```

---

writelas *Write a .las or .laz file*

Description

Write a .las or .laz file. All the fields are optional except X, Y and Z coordinates. If the user does not provide a field such as Intensity, for example, but this field is required according to the version of the file specified in the header, 0 will be written in this field. For more informations, see the ASPRS documentation for the LAS file format.

Usage

```
writelas(file, header, X, Y, Z, gpstime, Intensity, ReturnNumber,
  NumberOfReturns, ScanDirectionFlag, EdgeOfFlightline, Classification,
  ScanAngle, UserData, PointSourceID, R, G, B)
```

Arguments

file	character. filename of .las or .laz file
header	list of character. The data for the file header properly labelled (see readlasheader)
X	numeric array X data
Y	numeric array Y data
Z	numeric array Z data
gpstime	numeric array gpstime data
Intensity	integer array intensity data
ReturnNumber	integer array return number data
NumberOfReturns	integer array number of returns data
ScanDirectionFlag	integer array scan direction flag data
EdgeOfFlightline	integer array edge of flightline data
Classification	integer array classification data
ScanAngle	integer array scan angle data
UserData	integer array user data data

PointSourceID	integer array point source id data
R	integer array red data
G	integer array green data
B	integer array blue data

Value

void

See Also

Other rlas: readlasdata, readlasheader

Index