Package ‘rbgm’

April 3, 2025

Type Package

Title Tools for 'Box Geometry Model' (BGM) Files and Topology for the Atlantis Ecosystem Model

Version 0.2.0

Depends R (>= 3.2.2), raster, sp

Imports dplyr, geosphere, rlang, reproj, sfheaders

Suggests bgmfiles, covr, knitr, rmarkdown, roxygen2, testthat

Description Facilities for working with Atlantis box-geometry model (BGM) files. Atlantis is a deterministic, biogeochemical, whole-of-ecosystem model. Functions are provided to read from BGM files directly, preserving their internal topology, as well as helper functions to generate spatial data from these mesh forms. This functionality aims to simplify the creation and modification of box and geometry as well as the ability to integrate with other data sources.

NeedsCompilation no

ByteCompile yes

License GPL-3

RoxygenNote 7.3.2

Encoding UTF-8

URL https://research.csiro.au/atlantis/

BugReports https://github.com/AustralianAntarcticDivision/rbgm/issues/

VignetteBuilder knitr

Author Michael D. Sumner [aut, cre]

Maintainer Michael D. Sumner <mdsumner@gmail.com>

Repository CRAN

Date/Publication 2025-04-03 02:40:02 UTC

Contents

`rbgm-package` *Utilities for BGM files for Atlantis*

Description

Tools for handling network data for Atlantis from box-geometry model (BGM) files

rbgm features

- read .bgm files and faithfully store all information so it can be round-tripped
- conversion from .bgm forms to Spatial classes (lines and polygons)
- (not yet implemented: write to .bgm)

I. Import

`bgmfile`	read directly from a .bgm file

II. Conversion

`boxSpatial`	convert boxes to a `SpatialPolygonsDataFrame`
`faceSpatial`	convert faces to a `SpatialLinesDataFrame`
`boundarySpatial`	convert boundary to a single-row `SpatialPolygonsDataFrame`
`nodeSpatial`	obtain all vertices as points
`pointSpatial`	obtain all instances of vertices as points

III. Miscellaneous

`build_dz`	Build Atlantis dz Values

Author(s)

Maintainer: Michael D. Sumner <mdsumner@gmail.com>

See Also

Useful links:

- https://research.csiro.au/atlantis/
- Report bugs at https://github.com/AustralianAntarcticDivision/rbgm/issues/

`bgmfile` *Read BGM*

Description

Read geometry and full topology from BGM files.

Usage

```
bgmfile(x, ...)

read_bgm(x, ...)
```

Arguments

`x`	path to a bgm file
`...`	ignored for now

Details

BGM is a file format used for the 'Box Geometry Model' in the Atlantis Ecosystem Model. This function reads everything from the .bgm file and returns it as linked tables.

See Also

See helper functions to convert the bgm tables to ‘Spatial‘ objects, `boxSpatial`, `faceSpatial`, `nodeSpatial`, `boundarySpatial`, `pointSpatial`

Examples

```
library(bgmfiles)
bfile <- sample(bgmfiles(), 1L)
bgm <- bgmfile(bfile)
str(bgm)
```

boxSpatial — *Convert to spatial format*

Description

Take the output of `bgmfile` and return a `Spatial` object or a sf object.

Usage

```
boxSpatial(bgm)

box_sp(bgm)

box_sf(bgm)

boundarySpatial(bgm)

boundary_sp(bgm)

boundary_sf(bgm)

node_sp(bgm)

point_sp(bgm)

faceSpatial(bgm)

face_sp(bgm)

face_sf(bgm)
```

Arguments

`bgm`	output of a BGM file, as returned by `bgmfile`

Details

Note that the '_sp' forms are aliased to original functions called '*Spatial', and now have '_sf' counterparts to return that format.

Value

Spatial* object or sf object

- box_sp `SpatialPolygonsDataFrame`
- face_sp `SpatialLinesDataFrame`
- boundary_sp `SpatialPolygonsDataFrame`

- node_sp SpatialPointsDataFrame
- point_sp SpatialPointsDataFrame

- box_sf sf with sfc_POLYGON column
- face_sf sf with sfc_LINESTRING column
- boundary_sf sf with sfc_POLYGON column
- node_sf sf with sfc_POINT column
- point_sf sf with sfc_POINT column

Warning

The sf objects created by ‘box_sf()‘, ‘node_sf()‘, ‘face_sf()‘, ‘boundary_sf()‘ and ‘point_sf()‘ were not created by the sf package. They were created with reference to the sf format prior to November 2019. If you have problems it may be necessary to recreate the ’crs’ part of the of the object with code like ‘x <- box_sf(bgm); library(sf); st_crs(x) <- st_crs(attr(x$geometry, "crs")$proj)‘.

Get in touch ([create an issue](https://github.com/AustralianAntarcticDivision/rbgm/issues)) if you have any troubles.

Examples

```
fname <- bgmfiles::bgmfiles(pattern = "antarctica_28")
bgm <- bgmfile(fname)
spdf <- box_sp(bgm)
sfdf <- box_sf(bgm)
sldf <- face_sp(bgm)

plot(spdf, col = grey(seq(0, 1, length = nrow(bgm$boxes))))
plot(sldf, col = rainbow(nrow(bgm$faces)), lwd = 2,  add = TRUE)
```

---

build_dz *Build Atlantis dz Values*

Description

Build dz layer values for Atlantis from a bottom value, up through successive intervals. Each value is the positive offset required to rise to the top of the current interval.

Usage

```
build_dz(
  z,
  zlayers = c(-Inf, -2000, -1000, -750, -400, -300, -200, -100, -50, -20, 0)
)
```

Arguments

z	lowermost value
zlayers	intervals of layer values

Details

Offset values are returned to move from `z` against the intervals in `zlayers`. The intervals are assumed to be sorted and increasing in value from `-Infinity`. Once the maximum layer is reached the result is padded by that top value.

Value

numeric vector of offset values

Examples

```
## sanity tests
build_dz(-5000)
build_dz(-1500)
##build_dz(300)  ## error
build_dz(0)    ## ok
## data
dd <- c(-4396.49, -2100.84, -4448.81, -411.96, -2703.56, -5232.96,
       -4176.25, -2862.37, -3795.6, -1024.64, -897.93, -1695.82, -4949.76,
    -5264.24, -2886.81)
## all values in a matrix for checking
## [zlayers, dd]
dzvals <- sapply(dd, build_dz)
## process into text
f1 <- function(x) sprintf("somelabel,%i,%s", x, paste(build_dz(dd[x]), collapse = ","))
tex1 <- sapply(seq(length(dd)),  f1)
## for example
f2 <- function(x) {
sprintf("morelabel,%i,%s", x, paste(as.integer(build_dz(dd[x])), collapse = ","))
}
tex2 <- sapply(seq(length(dd)),  f2)
```

---

nodeSpatial *Vertices as Spatial points.*

Description

Obtain all vertices as a `SpatialPointsDataFrame` or a sf dataframe.

Usage

```
nodeSpatial(bgm)

node_sf(bgm)

pointSpatial(bgm)

point_sf(bgm)
```

Arguments

`bgm`	BGM object from `bgmfile`

Details

Nodes are the unique coordinates (or vertices), points are the instances of those coordinates that exist in the model. `point_sp` or `point_sf` return all instances of the vertices with information about which boxes they belong to. `node_sp` and `node_sf` return all vertices.

Value

`SpatialPointsDataFrame` or sf data frame

Warning

The sf objects created by ‘box_sf()‘, ‘node_sf()‘, ‘face_sf()‘, ‘boundary_sf()‘ and ‘point_sf()‘ were not created by the sf package. They were created with reference to the sf format prior to November 2019. If you have problems it may be necessary to recreate the ’crs’ part of the of the object with code like ‘x <- node_sf(bgm); library(sf); st_crs(x) <- st_crs(attr(x$geometry, "crs")$proj)‘.

Get in touch ([create an issue](https://github.com/AustralianAntarcticDivision/rbgm/issues)) if you have any troubles.

Examples

```
fname <- bgmfiles::bgmfiles(pattern = "antarctica_28")
bgm <- bgmfile(fname)
spnode <- node_sp(bgm)
names(spnode)
nrow(spnode)  ## only unique vertices
nrow(bgm$vertices)

sppoints <- point_sp(bgm)
names(sppoints)
nrow(sppoints)
names(point_sf(bgm))
```

Index