Package 'ncdfgeom'

August 28, 2019

Type Package

Title 'NetCDF' Geometry and Time Series

Version 1.1.0

Date 2019-08-28

Description Tools to create time series and geometry 'NetCDF' files.

URL https://code.usgs.gov/water/ncdfgeom

BugReports https://github.com/USGS-R/ncdfgeom/issues

Imports RNetCDF, ncmeta, sf, dplyr, methods

Depends R (>= 3.0)

Suggests testthat, knitr, rmarkdown, pkgdown, tidyverse, sp, geoknife, ncdf4, jsonlite

License CC0

LazyData TRUE

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author David Blodgett [aut, cre], Luke Winslow [ctb]

Maintainer David Blodgett <dblodgett@usgs.gov>

Repository CRAN

Date/Publication 2019-08-28 19:20:02 UTC

R topics documented:

read_attribute_data 2
read_geometry 2
read_timeseries_dsg 3
write_attribute_data 4
write_geometry 5
write_timeseries_dsg 6

Index **8**

`read_attribute_data` *Read attribute dataframe from NetCDF-DSG file*

Description

Gets attribute data from a NetCDF-DSG file and returns it in a `data.frame`. This function is intended as a convenience to be used within workflows where the netCDF file is already open and well understood.

Usage

```
read_attribute_data(nc, instance_dim)
```

Arguments

`nc`	A NetCDF path or urlto be opened.
`instance_dim`	The NetCDF instance/station dimension.

Examples

```
hucPolygons <- sf::read_sf(system.file('extdata','example_huc_eta.json', package = 'ncdfgeom'))
hucPolygons_nc <- ncdfgeom::write_geometry(tempfile(), hucPolygons)

read_attribute_data(hucPolygons_nc, "instance")
```

`read_geometry` *Read NetCDF-CF spatial geometries*

Description

Attempts to convert a NetCDF-CF DSG Simple Geometry file into a sf data.frame.

Usage

```
read_geometry(nc_file)
```

Arguments

`nc_file`	character file path to the nc file to be read.

Value

sf `data.frame` containing spatial geometry of type found in the NetCDF-CF DSG file.

References

http://cfconventions.org/index.html

1. http://cfconventions.org/cf-conventions/cf-conventions.html#_features_and_feature_types

Examples

```
huc_eta_nc <- tempfile()
file.copy(system.file('extdata','example_huc_eta.nc', package = 'ncdfgeom'),
         huc_eta_nc, overwrite = TRUE)

vars <- ncmeta::nc_vars(huc_eta_nc)

hucPolygons <- sf::read_sf(system.file('extdata','example_huc_eta.json', package = 'ncdfgeom'))
plot(sf::st_geometry(hucPolygons))
names(hucPolygons)

hucPolygons_nc <- ncdfgeom::write_geometry(nc_file=huc_eta_nc,
                                           geom_data = hucPolygons,
                                           instance_dim_name = "station",
                                           variables = vars$name)
huc_poly <- read_geometry(huc_eta_nc)
plot(sf::st_geometry(huc_poly))
names(huc_poly)
```

`read_timeseries_dsg` *Read NetCDF-CF timeSeries featuretype*

Description

This function reads a timeseries discrete sampling geometry NetCDF file and returns a list containing the file's contents.

Usage

```
read_timeseries_dsg(nc_file)
```

Arguments

`nc_file`	character file path to the nc file to be read.

Details

The current implementation checks several NetCDF-CF specific conventions prior to attempting to read the file. The Conventions and featureType global attributes are checked but not strictly required.

Variables with standard_name and/or cf_role of station_id and/or timeseries_id are searched for to indicate which variable is the 'timeseries identifier'. The function stops if one is not found.

All variables are introspected for a coordinates attribute. This attribute is used to determine which variables are coordinate variables. If none are found an attempt to infer data variables by time and timeseries_id dimensions is made.

The coordinates variables are introspected and their standard_names used to determine which coordinate they are. Lat, lon, and time are required, height is not.

Variables with a coordinates attribute are assumed to be the 'data variables'.

Data variables are traversed and their metadata and data content put into lists within the main response list.

See the timeseries vignette for more information.

Value

list containing the contents of the NetCDF file.

References

http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/reference/FeatureDatasets/CFpointImplement.html

---

`write_attribute_data` *Write attribute data to NetCDF-CF*

Description

Creates a NetCDF file with an instance dimension, and any attributes from a data frame. Use to create the start of a NetCDF-DSG file. One character length dimension is created long enough to contain the longest provided character string. This function does not implement any CF convention attributes or standard names. Any columns of class date will be converted to character.

Usage

```
write_attribute_data(nc_file, att_data, instance_dim_name = "instance",
  units = rep("unknown", ncol(att_data)), overwrite = FALSE)
```

Arguments

nc_file	character file path to the nc file to be created. If adding to a file, it must already have the named instance dimension.
att_data	data.frame with instances as columns and attributes as rows.
instance_dim_name	character name for the instance dimension. Defaults to "instance"
units	character vector with units for each column of att_data. Defaults to "unknown" for all.
overwrite	boolean overwrite existing file? Will append if FALSE.

Examples

```
sample_data <- sf::st_set_geometry(sf::read_sf(system.file("shape/nc.shp",
                                                           package = "sf")),
                                   NULL)
example_file <-write_attribute_data(tempfile(), sample_data,
                                    units = rep("unknown", ncol(sample_data)))

try({
  ncdump <- system(paste("ncdump -h", example_file), intern = TRUE)
  cat(ncdump ,sep = "\n")
}, silent = TRUE)
```

---

write_geometry *Write geometries and attributes to NetCDF-CF*

Description

Creates a file with point, line or polygon instance data ready for the extended NetCDF-CF timeSeries featuretype format.

Will also add attributes if provided data has them.

Usage

```
write_geometry(nc_file, geom_data, instance_dim_name = NULL,
  variables = list())
```

Arguments

nc_file	character file path to the nc file to be created.
geom_data	sf data.frame with POINT, LINESTRING, MULTILINESTRING, POLYGON, or MULTIPOLYGON geometries. Note that three dimensional geometries are not supported. sp geometries will be coerced to sf with sf::as_Spatial.

instance_dim_name
: character Not required if adding geometry to a NetCDF-CF Discrete Sampling Geometries timeSeries file. For a new file, will use package default – "instance" – if not supplied.

variables
: character If a an existing netCDF files is provided, this list of variables that should be related to the geometries.

References

1. http://cfconventions.org/cf-conventions/cf-conventions.html

Examples

```
hucPolygons <- sf::read_sf(system.file('extdata','example_huc_eta.json', package = 'ncdfgeom'))

hucPolygons_nc <- ncdfgeom::write_geometry(nc_file=tempfile(),
                                           geom_data = hucPolygons)
try({
  ncdump <- system(paste("ncdump -h", hucPolygons_nc), intern = TRUE)
  cat(ncdump ,sep = "\n")
}, silent = TRUE)
```

---

write_timeseries_dsg *Write time series to NetCDF-CF*

Description

This function creates a timeseries discrete sampling geometry NetCDF file. It uses the orthogonal array encoding to write one `data.frame` per function call. This encoding is best suited to data with the same number of timesteps per instance (e.g. geometry or station).

Usage

```
write_timeseries_dsg(nc_file, instance_names, lats, lons, times, data,
  alts = NA, data_unit = "", data_prec = "double",
  data_metadata = list(name = "data", long_name = "unnamed data"),
  time_units = "days since 1970-01-01 00:00:00", attributes = list(),
  add_to_existing = FALSE, overwrite = FALSE)
```

Arguments

nc_file
: character file path to the nc file to be created.

instance_names
: character or numeric vector of names for each instance (e.g. station or geometry) to be added to the file.

lats
: numeric vector of latitudes

lons	numeric vector of longitudes
times	POSIXct vector of times. Must be of type POSIXct or an attempt to convert it will be made using as.POSIXct(times).
data	data.frame with each column corresponding to an instance. Rows correspond to time steps. nrow must be the same length as times. Column names must match instance names.
alts	numeric vector of altitudes (m above sea level) (Optional)
data_unit	character vector of data units. Length must be the same as number of columns in data parameter.
data_prec	character precision of observation data in NetCDF file. Valid options: 'short' 'integer' 'float' 'double' 'char'.
data_metadata	list A named list of strings: list(name='ShortVarName', long_name='A Long Name')
time_units	character units string in udunits format to use for time. Defaults to 'days since 1970-01-01 00:00:00'
attributes	list An optional list of attributes that will be added at the global level. See details for useful attributes.
add_to_existing	boolean If TRUE and the file already exists, variables will be added to the existing file. See details for more.
overwrite	boolean error if file exists.

Details

Suggested Global Variables: c(title = "title", abstract = "history", provider site = "institution", provider name ="source", description = "description")

Note regarding add_to_existing: add_to_existing = TRUE should only be used to add variables to an existing NetCDF discrete sampling geometry file. All other inputs should be the same as are already in the file. If the functions is called with add_to_existing=FALSE (the default), it will overwrite an existing file with the same name. The expected usage is to call this function repeatedly only changing the data, data_unit, data_prec and data_metadata inputs.

See the timeseries vignette for more information.

References

1. http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/reference/FeatureDatasets/CFpointImplement.html
2. http://cfconventions.org/cf-conventions/cf-conventions.html#_orthogonal_multidimensional_array_representation
3. http://cfconventions.org/Data/cf-conventions/cf-conventions-1.7/build/cf-conventions.html#time-series-data

Index

read_attribute_data, 2
read_geometry, 2
read_timeseries_dsg, 3

write_attribute_data, 4
write_geometry, 5
write_timeseries_dsg, 6