# Package 'ncmeta'

March 25, 2024

**Title** Straightforward 'NetCDF' Metadata

**Version** 0.4.0

**Description** Extract metadata from 'NetCDF' data sources, these can be files, file handles or servers. This package leverages and extends the lower level functions of the 'RNetCDF' package providing a consistent set of functions that all return data frames. We introduce named concepts of 'grid', 'axis' and 'source' which are all meaningful entities without formal definition in the 'NetCDF' library <https://www.unidata.ucar.edu/software/netcdf/>. 'RNetCDF' matches the library itself with only the named concepts of 'variables', 'dimensions' and 'attributes'.

**Depends** R (>= 3.3.0)

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.0

**Imports** dplyr, rlang, RNetCDF, tibble, stats, tidyr, CFtime

**Suggests** testthat, covr

**URL** https://github.com/hypertidy/ncmeta, https://hypertidy.github.io/ncmeta/

**BugReports** https://github.com/hypertidy/ncmeta/issues

**ByteCompile** TRUE

**NeedsCompilation** no

**Author** Michael Sumner [aut, cre], Tomas Remenyi [ctb], Ben Raymond [ctb], David Blodgett [ctb], Milton Woods [ctb]

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

**Repository** CRAN

**Date/Publication** 2024-03-25 10:20:02 UTC

# R topics documented:



---

`nc_att` *NetCDF attributes*

---

## Description

Variable attributes are number 0:(n-1). Global attributes are indexed by -1 or the label "NC_GLOBAL".

## Usage

```
nc_att(x, variable, attribute, ...)

## S3 method for class 'NetCDF'
nc_att(x, variable, attribute, ...)

## S3 method for class 'character'
nc_att(x, variable, attribute, ...)
```

## Arguments

| | |
|---|---|
| `x` | or file handle |
| `variable` | name or index (zero based) of variable |
| `attribute` | name or index (zero based) of attribute |
| `...` | ignored |

## Details

nc_inq includes the number of global attributes nc_vars includes the number of variable attributes

## Value

data frame of attribute with numeric id, character attribute name, character or numeric variable id or name depending on input, and attribute value.

## Examples

```
f <- system.file("extdata", "S2008001.L3m_DAY_CHL_chlor_a_9km.nc", package = "ncmeta")
nc_att(f, 0, 0)
```

---

# nc_atts *NetCDF attributes*

---

## Description

All attributes in the file, globals are treated as if they belong to variable 'NC_GLOBAL'. Attributes for a single variable may be returned by specifying 'variable' - 'NC_GLOBAL' can stand in to return only those attributes.

## Usage

```
nc_atts(x, variable = NULL, ...)

## S3 method for class 'NetCDF'
nc_atts(x, variable = NULL, ...)

## S3 method for class 'character'
nc_atts(x, variable = NULL, ...)
```

## Arguments

| | |
|---|---|
| x | filename or handle |
| variable | optional single name of a variable, or 'NC_GLOBAL' |
| ... | ignored |

## Value

data frame of attributes

## Examples

```
f <- system.file("extdata", "S2008001.L3m_DAY_CHL_chlor_a_9km.nc", package = "ncmeta")
nc_atts(f)
```

## nc_axes *NetCDF axes*

### Description

An `axis` is an instance of a dimension.

### Usage

```
nc_axes(x, variables = NULL, ...)

## S3 method for class 'character'
nc_axes(x, variables = NULL, ...)

## S3 method for class 'NetCDF'
nc_axes(x, variables = NULL, ...)
```

### Arguments

| | |
|---|---|
| `x` | NetCDF source |
| `variables` | names of vars to query |
| `...` | ignored |

### Details

Each data source has a set of dimensions available for use by variables. Each axis is a 1-dimensional instance.

## nc_axis *NetCDF axes*

### Description

An `axis` is an instance of a dimension.

### Usage

```
nc_axis(x, i)

## S3 method for class 'character'
nc_axis(x, i)

## S3 method for class 'NetCDF'
nc_axis(x, i)
```

### Arguments

| | |
|---|---|
| x | NetCDF source |
| i | index of axis (1-based, 0 is "empty") |

### Details

Each data source has a set of dimensions available for use by variables. Each axis is a 1-dimensional instance.

---

## nc_coord_var *Get Coordinate Variables for Given Variable*

---

### Description

In NetCDF, variables are defined along dimensions and are said to have "coordinate variables" that define the (typically spatio-temporal) positions of the data's cells.

### Usage

```
nc_coord_var(x, variable = NULL, ...)

## S3 method for class 'character'
nc_coord_var(x, variable = NULL, ...)

## S3 method for class 'NetCDF'
nc_coord_var(x, variable = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | NetCDF source |
| variable | variable name of interest. If not included, all variables will be returned. |
| ... | ignored |

### Details

This function attempts to identify the X, Y, Z, and T coordinate variables for each data variable in the provided NetCDF source. The NetCDF-CF attribute conventions are used to make this determination.

All variables that can be related to a spatio-temporal axis, including coordinate variables are returned. For coordinate variables, a "bounds" column is included in the response indicating which variable contains bounds information.

See http://cfconventions.org/cf-conventions/v1.6.0/cf-conventions.html#coordinate-system for more.

## Value

tibble with "variable", "X", "Y", "Z", "T", and "bounds" columns that reference variables by name.

## Examples

```
f <- system.file("extdata", "S2008001.L3m_DAY_CHL_chlor_a_9km.nc", package = "ncmeta")
nc_coord_var(f, "chlor_a")

f <- system.file("extdata", "guam.nc", package = "ncmeta")
nc_coord_var(f)
```

---

`nc_dim` *NetCDF variables Obtain information about a single dimension by index.*

---

## Description

NetCDF variables Obtain information about a single dimension by index.

## Usage

```
nc_dim(x, i, ...)

## S3 method for class 'character'
nc_dim(x, i, ...)

## S3 method for class 'NetCDF'
nc_dim(x, i, ...)

## S3 method for class 'ncdf4'
nc_dim(x, i, ...)
```

## Arguments

| | |
|---|---|
| `x` | filename or handle |
| `i` | index of dimension (zero based) |
| `...` | ignored |

## See Also

`nc_vars` to obtain information about all dimensions, `nc_inq` for an overview of the file

## nc_dims *NetCDF dimension*

### Description

Get information about the dimensions in a NetCDF source.

### Usage

```
nc_dims(x, ...)

## S3 method for class 'character'
nc_dims(x, ...)

## S3 method for class 'NetCDF'
nc_dims(x, ...)

## S3 method for class 'ncdf4'
nc_dims(x, ...)
```

### Arguments

| | |
|---|---|
| x | file address or handle |
| ... | ignored |

## nc_extended *NetCDF extended dimension attributes*

### Description

Generate a table of all extended dimension attributes. For now that means interpretation of any "time" dimension.

### Usage

```
nc_extended(x, ...)

## S3 method for class 'character'
nc_extended(x, ...)

## S3 method for class 'NetCDF'
nc_extended(x, ...)

## S3 method for class 'ncdf4'
nc_extended(x, ...)
```

### Arguments

| | |
|---|---|
| x | filename or handle |
| ... | ignored currently |

### Value

data frame of extended dimension attribute information

---

`nc_gm_to_prj` *Get projection from NetCDF-CF Grid Mapping*

---

### Description

Takes NetCDF-CF grid mapping attributes and returns a proj4 string.

### Usage

```
nc_gm_to_prj(x)

## S3 method for class 'data.frame'
nc_gm_to_prj(x)

## S3 method for class 'list'
nc_gm_to_prj(x)
```

### Arguments

| | |
|---|---|
| x | list or data.frame of attributes of the grid mapping variable as returned by ncdf or ncdf4's get attributes functions or ncmeta's nc_grid_mapping_atts. |

### Details

The WGS84 datum is used as a default if one os not provided in the grid mapping.

If only a semi_major axis is provided, a sperical earth is assumed.

### Value

A proj4 string.

### References

1. https://en.wikibooks.org/wiki/PROJ.4
2. https://trac.osgeo.org/gdal/wiki/NetCDF_ProjectionTestingStatus
3. http://cfconventions.org/cf-conventions/cf-conventions.html#appendix-grid-mappings

## Examples

```
crs <- list(grid_mapping_name="latitude_longitude",
            longitude_of_prime_meridian = 0,
            semi_major_axis = 6378137,
            inverse_flattening = 298)
nc_gm_to_prj(crs)
```

---

# nc_grids *NetCDF grids*

---

## Description

A grid is a discretized space, defined by a set of dimensions. These are the spaces used by one or more variables in a source. Traditional summaries are organized by variable, but when organized by space or grid we can treat multiple variables together using standard database techniques.

## Usage

```
nc_grids(x, ...)

## S3 method for class 'character'
nc_grids(x, ...)

## S3 method for class 'NetCDF'
nc_grids(x, ...)

## S3 method for class 'tidync'
nc_grids(x, ...)
```

## Arguments

| | |
|---|---|
| x | NetCDF source |
| ... | ignored |

## Details

Each data source has a set of dimensions available for use by variables. Each grid is an n-dimensional space available for use by 0, 1 or more variables. A grid only really exists if variable is defined for it, and 'grid' is an implicit entity not an explicit part of the NetCDF API definition. The Unidata pages refer to "shape", which is more or less what we mean by "grid".

## nc_grid_mapping_atts *Get Grid Mapping*

### Description

Get the grid mapping from a NetCDF file

### Usage

```
nc_grid_mapping_atts(x, data_variable = NULL)

## S3 method for class 'character'
nc_grid_mapping_atts(x, data_variable = NULL)

## S3 method for class 'NetCDF'
nc_grid_mapping_atts(x, data_variable = NULL)

## S3 method for class 'data.frame'
nc_grid_mapping_atts(x, data_variable = NULL)
```

### Arguments

| | |
|---|---|
| x | open NetCDF object, character file path or url to be opened with RNetCDF::open.nc, or data.frame as returned from ncmeta::nc_atts |
| data_variable | character variable of interest |

### Value

tibble containing attributes that make up the file's grid_mapping. A data_variable column is included to indicate which data variable the grid mapping belongs to.

### Examples

```
nc_grid_mapping_atts(system.file("extdata/daymet_sample.nc", package = "ncmeta"))
```

## nc_inq *File info*

### Description

Get information about a NetCDF data source, may be a file path, or a RNetCDF file handle, or an OpenDAP/Thredds server address.

## Usage

```
nc_inq(x, ...)

## S3 method for class 'NetCDF'
nc_inq(x, ...)

## S3 method for class 'character'
nc_inq(x, ...)
```

## Arguments

| | |
|---|---|
| x | filename or handle |
| ... | ignored |

## Examples

```
## Not run:
 f <- raadfiles:::cmip5_files()$fullname[1]
 nc_inq(f)
 nc_var(f, 0)
 nc_dim(f, 0)

## End(Not run)

f <- system.file("extdata", "S2008001.L3m_DAY_CHL_chlor_a_9km.nc", package = "ncmeta")
nc_inq(f)
nc_var(f, 0)
nc_dim(f, 0)

nc_vars(f)
nc_dims(f)
```

---

# nc_meta *Top level NetCDF metadata.*

---

## Description

This function exists to maintain the open connection while all dimension, variable, and attribute metadata is extracted.

## Usage

```
nc_meta(x, ...)

## S3 method for class 'NetCDF'
nc_meta(x, ...)
```

```
## S3 method for class 'character'
nc_meta(x, ...)
```

## Arguments

| | |
|---|---|
| x | data source address, file name or handle |
| ... | ignored |

## Details

This function is pretty ambitious, and will send nearly any string to the underlying NetCDF library other than "", which immediately generates an error. This should be robust, but might present fairly obscure error messages from the underlying library.

## Examples

```
f <- system.file("extdata", "S2008001.L3m_DAY_CHL_chlor_a_9km.nc", package = "ncmeta")
nc_meta(f)

## Not run:
u <- "https://upwell.pfeg.noaa.gov/erddap/tabledap/FRDCPSTrawlLHHaulCatch"
nc_meta(u)

## End(Not run)
```

---

# nc_prj_to_gridmapping *Get NetCDF-CF grid mapping from projection*

---

## Description

Takes a proj4 string and returns a NetCDF-CF projection as a named list of attributes.

## Usage

```
nc_prj_to_gridmapping(prj)
```

## Arguments

| | |
|---|---|
| prj | character PROJ string as used in raster, sf, sp, proj4, and rgdal packages. |

## Value

A named list containing attributes required for that grid_mapping.

## References

1. https://en.wikibooks.org/wiki/PROJ.4
2. https://trac.osgeo.org/gdal/wiki/NetCDF_ProjectionTestingStatus
3. http://cfconventions.org/cf-conventions/cf-conventions.html#appendix-grid-mappings

## Examples

```
prj <- "+proj=longlat +datum=NAD27 +no_defs"
nc_prj_to_gridmapping(prj)
p1 <- "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96"
p2 <- "+x_0=0 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs"
prj2 <- sprintf("%s %s", p1, p2)
nc_prj_to_gridmapping(prj2)

nc_prj_to_gridmapping("+proj=longlat +a=6378137 +f=0.00335281066474748 +pm=0 +no_defs")
```

---

# nc_sources *NetCDF sources*

---

## Description

A record of file, URL, or any data source with NetCDF.

## Usage

```
nc_sources(x, ...)

## S3 method for class 'character'
nc_sources(x, ...)
```

## Arguments

| | |
|---|---|
| x | data source string |
| ... | ignored |

---

# nc_var *NetCDF variable*

---

## Description

Return a data frame about the variable at index `i`.

## Usage

```
nc_var(x, i)

## S3 method for class 'character'
nc_var(x, i)

## S3 method for class 'NetCDF'
nc_var(x, i)
```

### Arguments

| | |
|---|---|
| x | file name or handle |
| i | variable index (zero based) |

### Value

data frame of variable information

### See Also

nc_vars to obtain information about all variables, nc_inq for an overview of the file

---

## nc_vars *NetCDF variables*

---

### Description

Generate a table of all variables.

### Usage

```
nc_vars(x, ...)

## S3 method for class 'character'
nc_vars(x, ...)

## S3 method for class 'NetCDF'
nc_vars(x, ...)
```

### Arguments

| | |
|---|---|
| x | filename or handle |
| ... | ignored currently |

### Value

data frame of variable information

# Index