

Package ‘metis’

January 4, 2019

Title Sub-Regional Nexus Modeling Tool

Version 0.0.1

Description Package to process water-energy-land nexus data to different sub-regional levels.

Depends

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.1.0

Imports

raster, RColorBrewer, rgcam, tibble, dplyr, tmap, ggplot2, scales, utils, tidyr, rlang, grDevices, processx, rgdal, magrittr, sp, methods, tidyselect, rgeos, zoo, stats

Remotes github::JGCRI/rgcam

VignetteBuilder knitr

R topics documented:

metis.io	2
metis.irio	2
metis	3
metis.assumptions	3
metis.boundaries	4
metis.chart	5
metis.chartsProcess	7
metis.colors	9
metis.grid2poly	10
metis.map	11
metis.mapProcess	13
metis.prepGrid	15
metis.readgcam	16
metis.templates	18

Index	19
--------------	-----------

met.io

metis.io

Description

This function prepares gridded data for use with other metis modules.

Usage

```
met.io(Z0 = NULL, D0 = NULL, X0 = NULL, D = NULL,
       dirOutputs = paste(getwd(), "/outputs", sep = ""))
```

Arguments

Z0	Default = NULL,
D0	Default = NULL,
X0	Default = NULL,
D	Default = NULL,
dirOutputs	Default =paste(getwd(),"/outputs",sep=""),

Value

A table with data by polygon ID for each shapefile provided

met.irio

metis.irio

Description

This function prepares gridded data for use with other metis modules.

Usage

```
met.irio(Z0 = NULL, D0 = NULL, X0 = NULL, D = NULL,
         dirOutputs = paste(getwd(), "/outputs", sep = ""))
```

Arguments

Z0	Default = NULL,
D0	Default = NULL,
X0	Default = NULL,
D	Default = NULL,
dirOutputs	Default =paste(getwd(),"/outputs",sep=""),

Value

A table with data by polygon ID for each shapefile provided

metis	<i>metis: Sub-Regional nexus Package</i>
-------	------------------------------------------

Description

The Metis package provides

Metis functions

The Metis functions ...

metis.assumptions	<i>metis.assumptions</i>
-------------------	--------------------------

Description

This function loads holds the different assumptions used throughout the metis package.

Usage

```
metis.assumptions()
```

Details

List of Assumptions

- convEJ2TWh
- convEJ2GW
- conv1975USDperGJ22017USDperMWh
- conv1975USDperGJ22017USDperMBTU
- convertGgTgMTC
- GWPType

Value

A list of assumptions

Examples

```
library(metis)
a<-metis.assumptions()
a # will give full list of assumptions
```

metis.boundaries

metis.boundaries

Description

This function takes a .csv file with gridded lat, long data and aggregates the data by spatial boundaries given different shapefiles.

Usage

```
metis.boundaries(boundaryRegShape = NULL, boundaryRegShpFolder = NULL,
  boundaryRegShpFile = NULL, boundaryRegCol = NULL,
  boundaryRegionsSelect = NULL, subRegShape = NULL,
  subRegShpFolder = NULL, subRegShpFile = NULL, subRegCol = NULL,
  subRegionsSelect = NULL, subRegType = "subRegType",
  dirOutputs = paste(getwd(), "/outputs", sep = ""), nameAppend = "",
  expandPercent = 2, overlapShape = NULL, overlapShpFolder = NULL,
  overlapShpFile = NULL, labelsSize = 1.2, fillColorNA = NULL,
  projX = "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0",
  extendedFillColor = "grey75", extendedBGColor = "lightblue1",
  extendedHighLightColor = "cornsilk1", extendedLabelsColor = "grey30",
  extdendedLabelSize = 0.7, extension = T, fillPalette = "Spectral",
  cropSubShape2Bound = T, grids = NULL)
```

Arguments

boundaryRegShape

Default=NULL. Boundary region shape if already read into R.

boundaryRegShpFolder

Default= NULL. Folder containing boundary region shapefile. Suggested: `paste(getwd(), "/dataFiles/gis/naturalEarth", sep Default="")`,

boundaryRegShpFile

Default=NULL. Name of shapefile. Suggested: `paste("ne_10m_admin_0_countries", sep Default="")`,

boundaryRegCol Default=NULL. Column name with region names. Suggested "NAME_0",

boundaryRegionsSelect

Default=NULL. The region to choose from the given shapefile.

subRegShape

Default=NULL. Sub-region shape if already read into R.

subRegShpFolder

Default=NULL. Folder containing boundary region shapefile. Suggested `paste(getwd(), "/dataFiles/gis/naturalEarth", sep Default="")`,

subRegShpFile

Default=NULL. Name of sub-region shapefile. Suggested `paste("ne_10m_admin_1_states_prov Default="")`,

subRegCol

Default= NULL. Suggested for states "name",

subRegionsSelect

Default=NULL. The region to choose from the given sub-region shapefile.

subRegType

Default="subRegType". Eg. "states", "basins" etc.

dirOutputs

Default=`paste(getwd(), "/outputs", sep Default="")`. Location for outputs.

nameAppend	Default="".
expandPercent	Default=2. Percentage to expand boundary region beyond chosen region.
overlapShape	Default = NULL. If boundary lines of another shapefile are desired specify the shape here.
overlapShpFolder	Default = NULL. For GCAM basins use paste(getwd(),"/dataFiles/gis/basin_gcam",sep="").
overlapShpFile	Default = NULL. For GCAM basins use ="Global235_CLM_final_5arcmin_multipart"
labelsSize	Default =1.2.
fillcolorNA	Default =NULL.
projX	Default =" +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0".
extendedFillColor	Default = "grey75".
extendedBGColor	Default = "lightblue1".
extendedHighLightColor	Default = "cornsilk1".
extendedLabelsColor	Default = "grey30".
extdendedLabelSize	Default =0.7.
extension	Default = T
fillPalette	Default ="Spectral".
cropSubShape2Bound	Default = T. Set to False if subregion shape is larger than boundary, but desired fro extension.
grids	Default = NULL. Suggested is c(paste(getwd(),"/dataFiles/grids/emptyGrids/grid_025.csv",paste(getwd(),"/dataFiles/grids/emptyGrids/grid_050.csv",sep="")) This may happen in the case of disputed boundaries.

Value

A table with data by polygon ID for each shapefile provided

metis.chart	metis.chart
-------------	-------------

Description

This function produce different kinds of charts for the metis package. iIt requires a table in the Metis format. Each figure is accompanied with a csv table.

Usage

```
metis.chart(data, chartType = "bar", position = "stack", xData = "x",
  yData = "value", class = "class1", group = "scenario",
  classPalette = "classPalette1", classLabel = "classLabel1",
  xLabel = "xLabel", yLabel = "yLabel", facet_rows = "region",
  facet_columns = "scenario", ncolrow = 4, scales = "fixed",
  useNewLabels = 0, units = "units", xBreaksMaj = 10,
  xBreaksMin = 5, yBreaksMaj = 5, yBreaksMin = 10,
  sizeBarLines = 0.5, sizeLines = 1.5, printFig = T,
  fileName = "chart", dirOutputs = paste(getwd(), "/outputs", sep =
  ""), figWidth = 13, figHeight = 9, pdfpng = "png")
```

Arguments

<code>data</code>	data table for charting
<code>chartType</code>	Type of chart: "bar" or "line"
<code>position</code>	Position in bar charts. "identity", "stack" or "dodge"
<code>xData</code>	Default "x"
<code>yData</code>	Default "value"
<code>class</code>	Default "class1"
<code>group</code>	Default "scenario"
<code>classPalette</code>	Default "classPalette1"
<code>classLabel</code>	Default "classLabel1"
<code>xLabel</code>	Default "xLabel"
<code>yLabel</code>	Default "units"
<code>facet_rows</code>	Default "region"
<code>facet_columns</code>	Default "scenario"
<code>ncolrow</code>	Number of columns or Rows for Faceted plots
<code>scales</code>	Default "fixed"
<code>useNewLabels</code>	Default 0
<code>units</code>	Default "units"
<code>xBreaksMaj</code>	Default 10
<code>xBreaksMin</code>	Default 5
<code>yBreaksMaj</code>	Default 5
<code>yBreaksMin</code>	Default 10
<code>sizeBarLines</code>	Default 0.5
<code>sizeLines</code>	Default 1.5
<code>printFig</code>	Default = T,
<code>fileName</code>	Default = "map",
<code>dirOutputs</code>	Default = paste(getwd(), "/outputs", sep Default = "")
<code>figWidth</code>	Default = 9,
<code>figHeight</code>	Default = 7,
<code>pdfpng</code>	Default = "png",

Value

Returns the formatted data used to produce chart

Examples

```
# Examples below show the default chart with minimum information
# and then adding progressively more details.

library(tibble)
library(dplyr)
tbl <- tribble (~x, ~value,
  2010, 15,
  2020, 20,
  2030, 30
)
metis.chart(data=tbl,xData="x",yData="value",chartType = "line")
metis.chart(data=tbl,xData="x",yData="value",chartType = "bar")
```

metis.chartsProcess	<i>metis.chartsProcess</i>
---------------------	----------------------------

Description

This function produces charts given any number of tables in the metis format. The metis.chart() function produces charts for each region and scenario. If there are more than one scenario then the function also produces a folder for diffplots. The input tables should be .csv files with the following columns: scenario, region, sources, param, x, xLabel, vintage, class1, class2, units, value, aggregate, classLabel1, classPalette1, classLabel2, classPalette2. Running the metis.readgcam automatically produces an empty template with these columns for the relevant parameters. Each column is defined below:

Usage

```
metis.chartsProcess(dataTables = NULL, rTable = NULL, scenRef = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""), pdfpng = "png",
  xRange = "All", xCompare = c("2015", "2030", "2050", "2100"),
  paramsSelect = "All", regionsSelect = "All", xData = "x",
  yData = "value", xLabel = "xLabel", yLabel = "units",
  aggregate = "sum", class = "class", classPalette = "pal_Basic",
  regionCompareOnly = 1, useNewLabels = 0, sizeBarLines = 0,
  sizeLines = 1.5, nameAppend = "")
```

Arguments

dataTables	Vector of strings with full path to datatables to be read in. Example <code>c("D:/metis/outputs/Colombia/dataTable.Colombia_1975to2100.csv", "D:/metis/outputs/Colombia/dataTableLocal.Colombia_1975to2100.csv")</code> . Where "dataTableLocal.Colombia_1975to2100.csv" is the new datafile created based on "dataTableTemplate.Colombia_1975to2100.csv" and contains new local data.
rTable	If a table is created directly in R as a data.frame or tibble it can be entered here.

scenRef	The reference scenario to compare against. Default will pick first scenario from list of all scenarios
dirOutputs	Full path to directory for outputs. Default is <code>paste(getwd(), "/outputs", sep="")</code>
pdfpng	Choose the format for outputs. Either "pdf", "png" or "both". Default is "png"
xRange	Default "All". Range of x values eg. <code>c(2001:2005)</code>
xCompare	Choose the years to compare scenarios for <code>xScenSelectYears</code> plot. Default is <code>c("2015", "2030", "2050", "2100")</code>
paramsSelect	Default = "All". Select the parameters to analyze from the tables provided. Full list of parameters: <code>c("finalNrgbySec", "primNrgConsumByFuel", "elecByTech", "watConsumBySec", "watWithdrawBySec", "watWithdrawByCrop", "watBioPhysCons", "irrWatWithBasin", "irrWatConsBasin", "gdpPerCapita", "gdp", "gdpGrowthRate", "pop", "agProdbyIrrRfd", "agProdBiomass", "agProdForest", "agProdByCrop", "landIrrRfd", "aggLandAlloc", "LUCemiss", "co2emission", "co2emissionByEndUse", "ghgEmissionByGHG", "ghgEmissByGHGGROUPS")</code>
regionsSelect	Default = "All". Select regions to create charts for.
xData	Default "x"
yData	Default "value"
xLabel	Default "xLabel"
yLabel	Default "units"
aggregate	Default "sum"
class	Default "class"
classPalette	Default "pal_Basic" from <code>metis.colors()\$pal_Basic</code>
regionCompareOnly	Default 0. If set to 1, will only run comparison plots and not individual
useNewLabels	Default 0
sizeBarLines	Default 0.5
sizeLines	Default 1.5
nameAppend	Default ""

Details

List of Assumptions

- scenario: The name of the new data scenario
- region: The region for the data
- sources: Sources for the data
- param: Name of the parameter
- x: The x axis variable values
- xLabel: X axis Label
- vintage: Vintages if any. If not relevant then just enter "Vintage"
- class1: Classes or types (eg. if param is water_demands then the classes may be Industry, Agriculture etc.)
- class2: A second category of classes if exists.

- units: Units for the parameter. These are used as the y axis label.
- value: The parameter value.
- aggregate: Either "sum" or "mean". This parameter is used to determine how to aggregate across regions or scenarios.
- classLabel1: If class1 exists then this will be legend Label. If it doesn't exist enter "classLabel1"
- classPalette1: An R or metis.colors() palette. Can leave the default as "pal_16".
- classLabel2: If class2 exists then this will be legend Label. If it doesn't exist enter "classLabel2"
- classPalette2: An R or metis.colors() palette. Can leave the default as "pal_16".

Value

Produces charts in output folder and also returns combined table in metis format.

metis.colors	<i>metis.colors</i>
--------------	---------------------

Description

This function loads various color palettes used previously in GCAM as well as new palettes for Metis modeling to the global environment

Usage

```
metis.colors(palx = NULL)
```

Arguments

palx Palette name to view the palette colors. Eg. metis.colors("pal_Basic")

Details

List of Color Palettes

- pal_HDDCDD
- pal_16
- elec_tech_colors
- elec_renew_colors
- building_colors
- trn_fuel_colors
- enduse_fuel_numbered
- enduse_colors
- pal_pri_ene
- pal_pri_fuelcost
- pal_emiss_sector
- pal_landuse

- pal_hydrogen
- pal_refliq
- emiss_by_enduse_colors
- biouse_colors
- pal_Basic
- pal_Gas
- pal_Diff
- pal_Diff5
- pal_Absolute
- pal_Absolute5
- pal_Unassigned
- pal_elec_subsec
- pal_elec_finalNrgFuel
- pal_elec_techs
- pal_elec_sec
- pal_finalNrg_sec
- pal_pri_ene
- pal_elec_tech_colors

Value

A list of color palettes.

Examples

```
library(metis)
a<-metis.colors()
pie(rep(1,length(a$pal_Basic)),label=names(a$pal_Basic),col=a$pal_Basic)
```

metis.grid2poly

metis.grid2poly

Description

This function takes a .csv file with gridded lat, long data and aggregates the data by spatial boundaries given different shapefiles.

Usage

```
metis.grid2poly(grid = NULL, boundaryRegionsSelect = NULL,
  subRegShape = NULL, subRegShpFolder = NULL, subRegShpFile = NULL,
  subRegCol = NULL, subRegType = "subRegType", aggType = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""), nameAppend = "",
  labelsSize = 1.2)
```

Arguments

<code>grid</code>	Default=NULL. Grid file in .csv format or a R table, data frame or tibble with as a minimum columns with "lat", "lon" and "value",
<code>boundaryRegionsSelect</code>	Default=NULL. Larger region name which will be used as the folder name for outputs.
<code>subRegShape</code>	Default=NULL. shapefile over which grid data is to be aggregated.
<code>subRegShpFolder</code>	Default=NULL. Folder containing boundary region shapefile. Suggested <code>paste(getwd(), "/dataFiles/gis/naturalEarth", sep = "")</code> ,
<code>subRegShpFile</code>	Default=NULL. Name of sub-region shapefile. Suggested <code>paste("ne_10m_admin_1_states_prov", sep = "")</code> ,
<code>subRegCol</code>	Default= NULL. Suggested for states "name",
<code>subRegType</code>	Default="subRegType". Eg. "states", "basins" etc.
<code>aggType</code>	Default=NULL. Aggregation method to be used. Either "vol" or "depth" depending on the type of data provided.
<code>dirOutputs</code>	Default= <code>paste(getwd(), "/outputs", sep = "")</code> ,
<code>nameAppend</code>	Default="",
<code>labelsSize</code>	Default =1.2. Label size for the region names for the gridoverlay plot.

Value

A table with data by polygon ID for each shapefile provided

<code>metis.map</code>	<i>metis.map</i>
------------------------	------------------

Description

This function produce different kinds of maps for the metis package. Each figure is accompanied with a csv table.

Usage

```
metis.map(dataPolygon = NULL, dataGrid = NULL, dataRaster = NULL,
  shpFolder = NULL, shpFile = NULL, fillPalette = "Spectral",
  borderColor = "gray20", lwd = 1, lty = 1, bgColor = "white",
  frameShow = F, fillColumn = NULL, labels = F, labelsSize = 1.2,
  labelsColor = "black", labelsAutoPlace = F, figWidth = 9,
  figHeight = 7, legendWidth = -1, legendShow = F,
  legendOutside = T, legendTextSize = 1, legendTitleSize = 2,
  legendOutsidePosition = NULL, legendPosition = NULL,
  legendDigits = NULL, legendTitle = "Legend",
  legendStyle = "pretty", legendFixedBreaks = 5, legendBreaks = NULL,
  pdfpng = "png", underLayer = NULL, overLayer = NULL,
  printFig = T, fileName = "map", dirOutputs = paste(getwd(),
    "/outputs", sep = ""), facetFreeScale = F, facetRows = NA,
  facetCols = 3, facetBGColor = "grey30", facetLabelColor = "white",
  facetLabelSize = 1.5, alpha = 1, fillcolorNA = NULL,
  facetsON = T, panelLabel = NULL)
```

Arguments

dataPolygon	Default = NULL,
dataGrid	Default = NULL,
dataRaster	Default = NULL,
shpFolder	Default = paste(getwd(),"/dataFiles/gis/admin_gadm36_1",sep = ""),
shpFile	Default = paste("gadm36_1",sep = ""),
fillPalette	Default = "Spectral",
borderColor	Default = "gray20",
lwd	Default = 1,
lty	Default = 1,
bgColor	Default = "white",
frameShow	Default = F,
fillColumn	Default = NULL, # Or give column data with
labels	Default = F,
labelsSize	Default = 1.2,
labelsColor	Default = "black",
labelsAutoPlace	Default = F,
figWidth	Default = 9,
figHeight	Default = 7,
legendWidth	Default = -1,
legendShow	Default = F,
legendOutside	Default = T,
legendTextSize	Default = 0.8,
legendTitleSize	Default = 1,
legendOutsidePosition	Default = NULL, # "right","left","top","bottom", "center"
legendPosition	Default = NULL, # c("RIGHT','top') - RIGHT LEFT TOP BOTTOM
legendDigits	Default = NULL,
legendTitle	Default = "Legend",
legendStyle	Default = "pretty",
legendFixedBreaks	Default = "5",
legendBreaks	Default = NULL,
pdfpng	Default = "png",
underLayer	Default = NULL,
overLayer	Default = NULL,
printFig	Default = T,
fileName	Default = "map",
dirOutputs	Default = paste(getwd(),"/outputs",sep = ""),

```

facetFreeScale Default = F,
facetRows      Default = NA,
facetCols      Default = 3,
facetBGColor   Default = "grey75",
facetLabelColor
               Default = "black",
facetLabelSize Default = 1.5,
alpha          Default = 1
fillcolorNA    Default = NULL
facetsON       Default = F,
panelLabel     Default = NULL,

```

Value

Returns the formatted data used to produce chart

metis.mapProcess	<i>metis.mapProcess</i>
------------------	-------------------------

Description

This function produce different kinds of maps for the metis package. Each figure is accompanied with a csv table.

Usage

```

metis.mapProcess(polygonDataTables = NULL, gridDataTables = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""), xRange = "All",
  labels = F, labelsSize = 1.2, subRegShape = NULL,
  subRegShpFolder = NULL, subRegShpFile = NULL, subRegCol = NULL,
  subRegType = "subRegType", nameAppend = "",
  legendOutsidePosition = NULL, legendPosition = NULL,
  legendFixedBreaks = 5, animateOn = T, delay = 100,
  legendTitleSize = 1, scenRef = NULL, extension = F,
  boundaryRegShape = NULL, boundaryRegShpFolder = NULL,
  boundaryRegShpFile = NULL, boundaryRegCol = NULL,
  boundaryRegionsSelect = NULL, fillcolorNA = NULL,
  extendedFillColor = "grey75", extendedBGColor = "lightblue1",
  extendedHighLightColor = "cornsilk1", extendedLabelsColor = "grey30",
  extendedLabelSize = 0.7, extendedShape = NULL,
  extendedShapeCol = NULL, expandPercent = 2,
  projX = "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0")

```

Arguments

```

polygonDataTables      Default = NULL,
gridDataTables         Default = NULL,
dirOutputs             Default = paste(getwd(),"/outputs",sep=""),
xRange                Default = "All",
labels                Default = F,
labelsSize            Default = 1.2,
subRegShape           Default = NULL,
subRegShpFolder        Default = paste(getwd(),"/dataFiles/gis/admin_gadm36",sep=""),
subRegShpFile          Default = paste("gadm36_1",sep=""),
subRegCol              Default = "NAME_1",
subRegType             Default = "subRegType",
nameAppend            Default = ""
legendOutsidePosition  Default = NULL, # "right","left","top","bottom", "center"
legendPosition         Default = NULL, # c("RIGHT",'top') - RIGHT LEFT TOP BOTTOM
legendFixedBreaks      Default = "5",
animateOn             Default = T,
delay                 Default = 100,
legendTitleSize        Default = 1,
scenRef               Default = NULL
extension             Default = F,
boundaryRegShape       Default = NULL,
boundaryRegShpFolder   Default= NULL . Suggested paste(getwd(),"/dataFiles/gis/naturalEarth",sep
Default="")
boundaryRegShpFile     Default=NULL . Suggested paste("ne_10m_admin_0_countries",sep De-
fault=""),
boundaryRegCol         Default=NULL. Suggested "NAME_0",
boundaryRegionsSelect  Default = NULL,
fillcolorNA           Default = NULL
extendedFillColor      Default = "grey75",
extendedBGColor        Default = "lightblue1",
extendedHighLightColor Default = "cornsilk1",

```

```

extendedLabelsColor
    Default ="grey30",
extdendedLabelSize
    Default =0.7,
extendedShape      Default =NULL,
extendedShapeCol
    Default =NULL,
expandPercent      Default =2
projX              Default = projX="+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
                    +towgs84=0,0,0"

```

Value

Returns the formatted data used to produce chart

metis.prepGrid	<i>metis.prepGrid</i>
----------------	-----------------------

Description

This function prepares gridded data for use with other metis modules.

Usage

```

metis.prepGrid(demeterFolder = NULL, demeterScenario = NULL,
  demeterTimesteps = seq(from = 2005, to = 2100, by = 5),
  demeterUnits = NULL, tethysFolder = NULL, tethysScenario = NULL,
  tethysUnits = NULL, tethysFiles = c("wddom", "wdelec", "wdirr",
    "wdliv", "wdmfg", "wdmin", "wdnonag", "wdtotal"), xanthosFolder = NULL,
  xanthosScenario = NULL, xanthosUnits = NULL, xanthosFiles = NULL,
  xanthosCoordinatesPath = paste(getwd(),
    "/dataFiles/grids/xanthosCoords/coordinates.csv", sep = ""),
  scarcityXanthosRollMeanWindow = 10, popFolder = paste(getwd(),
    "/dataFiles/grids/griddedIDsPop/", sep = ""),
  popFiles = "grid_pop_map", popUnits = "person",
  dirOutputs = paste(getwd(), "/outputs", sep = ""), reReadData = 1,
  gridMetisData = paste(dirOutputs, "/Grids/gridMetis.RData", sep = ""))

```

Arguments

```

demeterFolder  Full path to demeter outputs
demeterScenario
    Name of demeter scenario
demeterTimesteps
    Default is seq(from=2005,to=2100,by=5)
demeterUnits    No Default
tethysFolder    Folder for tethys results
tethysScenario  Scenario name for tethys run
tethysUnits     No Default

```

```

tethysFiles      Default = c("wddom", "wdelec", "wdirr", "wdliv", "wdmfg", "wdmin", "wdnonag", "wdtotal"),
xanthosFolder    Xanthos Folder Path
xanthosScenario  Xanthos Scenario Name
xanthosUnits     Xanthos Untis
xanthosFiles     Xanthos Files to Read
xanthosCoordinatesPath
                  paste(getwd(), "/dataFiles/grids/xanthosCoords/coordinates.csv", sep="")
scarcityXanthosRollMeanWindow
                  Default = 10,
popFolder        Default = i-paste(getwd(), "/dataFiles/grids/griddedIDsPop/", sep="")
popFiles         Default = i-"grid_pop_map"
popUnits         Default = i-"person"
dirOutputs       Default = paste(getwd(), "/outputs", sep=""),
reReadData       Default = 1,
gridMetisData    Default = paste(dirOutputs, "/Grids/gridMetis.RData", sep = "")

```

Value

A table with data by polygon ID for each shapefile provided

metis.readgcam	<i>metis.readgcam</i>
----------------	-----------------------

Description

This function connects to a gcamdatabase and uses a query file to out results into a table ready for plotting.

Usage

```

metis.readgcam(gcamdatabasePath, gcamdatabaseName,
  queryxml = "metisQueries.xml", scenOrigNames, scenNewNames = NULL,
  reReadData = T, dataProj = "dataProj.proj",
  dirOutputs = paste(getwd(), "/outputs", sep = ""),
  regionsSelect = NULL, queriesSelect = "All", paramsSelect = "All")

```

Arguments

```

gcamdatabasePath  Path to gcam database folder
gcamdatabaseName  Name of gcam database
queryxml          Full path to query.xml file
scenOrigNames     Original Scenarios names in GCAM database in a string vector. For
                  example c('scenario1', 'scenario2').
scenNewNames      New Names which may be shorter and more useful for figures etc. Default
                  will use Original Names. For example c('scenario1', 'scenario2')

```


reReadData	If TRUE will read the GCAM data base and create a queryData.proj file in the same folder as the GCAM database. If FALSE will load a '.proj' file if a file with full path is provided otherwise it will search for a dataProj.proj file in the existing folder which may have been created from an old run.
dataProj	Optional. A default 'dataProj.proj' is produced if no .Proj file is specified.
dirOutputs	Full path to directory for outputs
regionsSelect	The regions to analyze in a vector. Example c('Colombia','Argentina')
queriesSelect	Default = "All". Vector of queries to read from the queryxml for example c("Total final energy by aggregate end-use sector", "Population by region"). The queries must be available in the queryxml file. Current list of queries and generated paramaters are: <ul style="list-style-type: none"> • "Total final energy by aggregate end-use sector". Parameters generated: finalNrgbySec. • "primary energy consumption by region (direct equivalent)". Parameters generated: primNrgConsumByFuel • "Electricity generation by aggregate technology". Parameters generated: elecByTech • "water withdrawals by sector". Parameters generated: watWithdrawBySec • "water consumption by sector". Parameters generated: watConsumBySec • "water withdrawals by crop". Parameters generated: watWithdrawByCrop • "biophysical water demand by crop type and land region". Parameters generated: watBioPhysCons • "water withdrawals by water mapping source". Parameters generated: irrWatWithBasin • "water consumption by water mapping source". Parameters generated: irrWatConsBasin • "GDP per capita MER by region". Where MER is "Market Exchange Rate". Parameters generated: gdpPerCapita. • "GDP MER by region". Where MER is "Market Exchange Rate". Parameters generated: gdp, gdpGrowthRate • "Population by region". Parameters generated: pop. • "ag production by tech". Where technologies signify irrigated or rain-fed. Parameters generated: agProdbyIrrRfd • "Ag Production by Crop Type". Parameters generated: agProd-Biomass, agProdForest, agProdByCrop • "land allocation by crop and water source". Parameters generated: landIrrRfd • "aggregated land allocation". Parameters generated: aggLandAlloc • "Land Use Change Emission". Parameters generated: LUCemissFut • "CO2 Emissions by enduse". Parameters generated: co2emission, co2emissionByEndUse, • "GHG emissions by subsector". Parameters generated: ghgEmiss-ByGHGGROUPS, ghgEmissionByGHG

paramsSelect Default = "All". If desired select a subset of parameters to analyze from the full list of parameters: c("finalNrgbySec", "primNrgConsumByFuel", "elecByTech", "watConsumBySec", "watWithdrawBySec", "watWithdrawByCrop", "watBioPhysCons", "irrWatWithBasin", "irrWatConsBasin", "gdpPerCapita", "gdp", "gdpGrowthRate", "pop", "agProdbyIrrRfd", "agProdBiomass", "agProdForest", "agProdByCrop", "landIrrRfd", "aggLandAlloc", "LUCemiss", "co2emission", "co2emissionByEndUse", "ghgEmissionByGHG", "ghgEmissByGHGGROUPS")

Value

A list with the scenarios in the gcam database, queries in the queryxml file and a tibble with gcam data formatted for metis charts.

metis.templates	<i>metis.templates</i>
-----------------	------------------------

Description

This script holds various templates used for different scripts.

Usage

```
metis.printPdfPng(figure, dir, filename, figWidth = 13, figHeight = 9,
  pdfpng = "png")
```

```
metis.chartsThemeLight()
```

Arguments

figure	Figure to be printed in function metis.printPdfPng
dir	Directory to print figure to in function metis.printPdfPng
filename	Filename for figure printed in function metis.printPdfPng
figWidth	Figure Width in inches for figures to be printed in function metis.printPdfPng
figHeight	Figure height in inches for figures to be printed in function metis.printPdfPng
pdfpng	Either "pdf", "png" or "both" to define the format of output

Details

List of Templates in this script:

- metis.printPdfPng: Function used to print charts to a pdf or png or both.
- metis.chartsThemeLight: A light ggplot theme for charts
- metis.tmapAnimate: A function to animate tmaps across a variable.
- metis.tmapLayout: A function to define tmap layouts

Value

A list of different templates

Index

- *Topic **assumptions**
 - metis.assumptions, 3
- *Topic **charts**,
 - metis.chart, 5
 - metis.chartsProcess, 7
 - metis.map, 11
 - metis.mapProcess, 13
 - metis.templates, 18
- *Topic **colors**,
 - metis.colors, 9
- *Topic **database**,
 - met.io, 2
 - met.irio, 2
 - metis.boundaries, 4
 - metis.grid2poly, 10
 - metis.prepGrid, 15
 - metis.readgcam, 16
- *Topic **diffplots**
 - metis.chart, 5
 - metis.chartsProcess, 7
 - metis.map, 11
 - metis.mapProcess, 13
- *Topic **gcam**,
 - met.io, 2
 - met.irio, 2
 - metis.boundaries, 4
 - metis.grid2poly, 10
 - metis.prepGrid, 15
 - metis.readgcam, 16
- *Topic **gcam**
 - met.io, 2
 - met.irio, 2
 - metis.boundaries, 4
 - metis.grid2poly, 10
 - metis.prepGrid, 15
 - metis.readgcam, 16
- *Topic **maps**,
 - metis.templates, 18
- *Topic **palette**
 - metis.colors, 9
- *Topic **print**
 - metis.templates, 18
- *Topic **query**
 - met.io, 2
 - met.irio, 2
 - metis.boundaries, 4
 - metis.grid2poly, 10
 - metis.prepGrid, 15
 - metis.readgcam, 16
- *Topic **templates**,
 - metis.templates, 18
- met.io, 2
- met.irio, 2
- metis, 3
- metis-package (metis), 3
- metis.assumptions, 3
- metis.boundaries, 4
- metis.chart, 5
- metis.chartsProcess, 7
- metis.chartsThemeLight
(metis.templates), 18
- metis.colors, 9
- metis.grid2poly, 10
- metis.map, 11
- metis.mapProcess, 13
- metis.prepGrid, 15
- metis.printPdfPng (metis.templates), 18
- metis.readgcam, 16
- metis.templates, 18