

# Package ‘metis’

October 12, 2019

**Title** Sub-Regional Nexus Modeling Tool

**Version** 1.0.0

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

**Depends**

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Imports** raster (*i*= 2.7.15),  
RColorBrewer (*i*= 1.1.2),  
rgcam (*i*= 0.5.0),  
tibble (*i*= 1.4.2),  
ggalluvial (*i*= 0.9.1),  
dplyr (*i*= 0.7.7),  
tmap (*i*= 2.1.1),  
ggplot2 (*i*= 3.1.0),  
scales (*i*= 0.5.0),  
utils (*i*= 3.5.0),  
tidyr (*i*= 0.8.1),  
rlang (*i*= 0.3.0),  
grDevices (*i*= 3.5.0),  
processx (*i*= 3.2.0),  
rgdal (*i*= 1.2.20),  
magrittr (*i*= 1.5),  
sp (*i*= 1.2.7),  
methods (*i*= 3.5.0),  
tidyselect (*i*= 0.2.5),  
rgeos (*i*= 0.3.26),  
zoo (*i*= 1.8.4),  
stats (*i*= 3.5.0),  
dbplyr (*i*= 1.3.0),  
RSQLite (*i*= 2.1.1),  
ggrepel (*i*= 0.8.1),  
data.table,  
stringr (*i*= 1.3.1),  
magick (*i*= 2.1),  
DBI,

testthat (*i*= 2.0.1),  
knitr (*i*= 1.20),  
rmarkdown (*i*= 1.10),  
XML (*i*= 3.98-1.20)

**Remotes** github::JGCRI/rgcam  
**VignetteBuilder** knitr

**R topics documented:**

metis . . . . .	2
metis.assumptions . . . . .	2
metis.bia . . . . .	3
metis.boundaries . . . . .	4
metis.chart . . . . .	6
metis.chartsProcess . . . . .	9
metis.colors . . . . .	11
metis.grid2poly . . . . .	12
metis.gridByPoly . . . . .	13
metis.io . . . . .	14
metis.map . . . . .	15
metis.mapsProcess . . . . .	17
metis.prepGrid . . . . .	21
metis.printPdfPng . . . . .	22
metis.readgcam . . . . .	23

<b>Index</b>	<b>26</b>
--------------	-----------

---

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.bia	<i>metis.bia</i>
-----------	------------------

---

## Description

This function downscales GCAM electricity generation and installed capacity onto a grid, based on WRI PowerWatch dataset of present capacity

## Usage

```
metis.bia(biaInputsFolder = "NA", biaInputsFiles = "NA",
  reReadData = 1, regionsSelect = NULL, dataProj = "dataProj.proj",
  dataProjPath = gcamdatabasePath, scenOrigNames = NULL,
  scenNewNames = NULL, gcamdatabasePath = "NA",
  gcamdatabaseName = "NA", queryxml = "metisQueries.xml",
  queryPath = paste(getwd(), "/dataFiles/gcam", sep = ""),
  queriesSelect = "All", paramsSelect = c("elecByTech",
  "elecCapBySubsector"), gridChoice = "grid_050", diagnosticsON = T,
  subsectorNAistribute = "even", nameAppend = "")
```

## Arguments

biaInputsFolder	Bia Inputs Folder Path
biaInputsFiles	Bia Inputs Folder Path
reReadData	Default = 1. 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.

regionsSelect	The regions to analyze in a vector. Example c('Colombia','Argentina')
dataProj	Optional. A default 'dataProj.proj' is produced if no .Proj file is specified.
dataProjPath	Folder that contains the dataProj or where it will be produced.
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')
gcamdatabasePath	Path to gcam database folder
gcamdatabaseName	Name of gcam database
queryxml	Full path to query.xml file
queryPath	Folder that contains the query.xml file.By default it is the same folder as specified by gcamdatabasePath
queriesSelect	Default = "All". Vector of queries to read from the queryxml for example
paramsSelect	Default = c("elecByTech", "elecCapBySubsector") . Vector of parameters to be read from the GCAM database
gridChoice	Default = "grid_050" . Choice of whether to use 50 km x 50 km grid cells ("grid_050") or 25 km x 25 km ("grid_025").
diagnosticsON	Default = T.
subsectorNAdistribute	Default = "even". Choose "even" for even distribution or "totalOther" to distribute based on sum of all other subsectors..
nameAppend	Default=""

### Value

A tibble with GCAM electricity generation distributed on a grid for selected regions

---

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,
  subRegCol1 = NULL, subRegionsSelect = NULL,
  subRegType = "subRegType", dirOutputs = paste(getwd(), "/outputs",
  sep = ""), folderName = NULL, 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",
extendedLabelSize = 0.7, extension = T, fillPalette = "Spectral",
cropSubShape2Bound = T, grids = NULL, innerMargins = c(0.1, 0.2,
0.1, 0.2), outerMargins = c(0.01, 0.01, 0.01, 0.01))
```

## 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/metis/naturalEarth", sep="")`,

**boundaryRegShpFile**

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

**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/metis/naturalEarth", sep="")`,

**subRegShpFile**

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

**subRegCol** Default= NULL. Suggested for states "name",

**subRegCol1** Default= NULL. Suggested for states "name",

**subRegionsSelect**

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

**subRegType** Default="subRegType". Type of subregion. Eg. "states", "basins" etc.

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

**folderName** Default = NULL,

**nameAppend** Default="". Name to append to saved files.

**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/metis/gcam", sep="")`

**overlapShpFile** Default = NULL. For GCAM basins use "Global235\_CLM\_final\_5arcmin\_multipart"

**labelsSize** Default =1.2.

**fillcolorNA** Default =NULL. Fill color for NA values.

**projX** Default = "+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0".

**extendedFillColor**

Default = "grey75". Color used to fill extended land areas.

extendedBGColor	Default = "lightblue1". Color used to fill background/water bodies.
extendedHighLightColor	Default = "cornsilk1". Color used to highlight region of analysis.
extendedLabelsColor	Default = "grey30". Color for extended country name labels.
extdendedLabelSize	Default = 0.7. Size of extended country name labels.
extension	Default = T. Should the map be extended beyond chosen shapefile boundaries.
fillPalette	Default = "Spectral". Palette to use to fill subregions.
cropSubShape2Bound	Default = T. If subregion shape file is larger than boundary file.
grids	Default = NULL. Metis comes with 0.5 and 0.25 grids in c(paste(getwd()), "/dataFiles/grids/em
innerMargins	Default = c(0,0.1,0,0.1), # bottom, left, top, right
outerMargins	Default = c(0.01,0.01,0.01,0.01) # bottom, left, top, right paste(getwd()), "/dataFiles/grids/em This may happen in the case of disputed boundaries.

### Value

A table with data by polygon ID for each shapefile provided

---

metis.chart	<i>metis.chart</i>
-------------	--------------------

---

### Description

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

### Usage

```
metis.chart(data, dataNorm = NULL, chartType = "bar",
  position = "stack", xData = "x", yData = "value",
  class = "class1", group = "scenario",
  classPalette = "classPalette1", classLabel = "classLabel1",
  color = NULL, xLabel = "xLabel", yLabel = "yLabel",
  facet_rows = NULL, facet_columns = NULL, ncolrow = 4,
  facetBGColor = "grey30", facetLabelColor = "white",
  facetLabelSize = 24, scales = "fixed", useNewLabels = 0,
  units = "units", xBreaksMaj = 10, xBreaksMin = 5,
  yBreaksMaj = 5, yBreaksMin = 10, sizeBarLines = 0.5,
  sizeLines = 1.5, yMax = NULL, yMin = NULL, sectorToOrder = NULL,
  sectorFromOrder = NULL, removeCols = NULL, bubbleSize = 10,
  sankeyAxis1 = NULL, sankeyAxis2 = NULL,
  sankeyAxis1Label = "axis1Label", sankeyAxis2Label = "axis2Label",
  sankeyGroupColor = NULL, printFig = T, fileName = "chart",
  title = NULL, dirOutputs = NULL, folderName = NULL,
  figWidth = 13, figHeight = 9, pdfpng = "png", sankeyLabelsOn = 1,
  colOrder1 = NULL, colOrderName1 = NULL, colOrder2 = NULL,
  colOrderName2 = NULL, pointsOn = 1, pointsSize = 4,
  paletteRev = T, forceFacets = F)
```

**Arguments**

<code>data</code>	Data table for charting
<code>dataNorm</code>	Normalized data to plot under actual data in bubble plots. Default = NULL,
<code>chartType</code>	Type of chart: "bar", "line", "bubble", "sankey"
<code>position</code>	Position in bar charts. "identity", "stack" or "dodge"
<code>xData</code>	X axis data variable (dataframe or table column name). Default "x".
<code>yData</code>	Y axis data variable (dataframe or table column name).Default "value"
<code>class</code>	Class data variable (dataframe or table column name).Default "class1"
<code>group</code>	Group (dataframe or table column name).Default "scenario"
<code>classPalette</code>	Color palette to use for multiple classes. Must be a color palette eg. <code>c("red","blue","green")</code> or a <code>metis.colors()</code> palette eg. <code>metis.colors()\$pal.Basic</code> . Default "classPalette1"
<code>classLabel</code>	Label to be used for legend title. Default "classLabel1"
<code>color</code>	A single color name for single class charts. Default NULL
<code>xLabel</code>	X axis title. Default "xLabel"
<code>yLabel</code>	Y axis title. Default "units"
<code>facet_rows</code>	Data variable to be used for facet rows (dataframe or table column name).Default "region"
<code>facet_columns</code>	Data variable to be used for facet columns (dataframe or table column name).Default "scenario"
<code>ncolrow</code>	Number of columns or Rows for Faceted plots.
<code>facetBGColor</code>	Facet background color. Default ="grey30",
<code>facetLabelColor</code>	Facet title text color. Default= "white",
<code>facetLabelSize</code>	Facet title text size. Default =1.5,
<code>scales</code>	Fixed or free scales for multiple sankey plots. Default "fixed"
<code>useNewLabels</code>	"1" or "0". Converts labels to title-case.Default 0
<code>units</code>	Data units. Default "units"
<code>xBreaksMaj</code>	X axis major breaks. Default 10
<code>xBreaksMin</code>	X axis minor breaks. Default 5
<code>yBreaksMajn</code>	Y axis major breaks. Default 5
<code>yBreaksMinn</code>	Y axis minor breaks. Default 10
<code>sizeBarLines</code>	Bar plot line size. Default 0.5
<code>sizeLines</code>	Line plot line size. Default 1.5
<code>yMax</code>	Y axis max value
<code>yMin</code>	Y axis min value
<code>sectorToOrder</code>	Order of "to" column variables in bubble plots. Default = NULL,
<code>sectorFromOrder</code>	Order of "from" column variables in bubble plots. Default = NULL,
<code>removeCols</code>	Option to remove certain columns from bubble plots. Default = NULL,
<code>bubbleSize</code>	Bubble plot bubble size. Default = 10,

sankeyAxis1	Sankey axis 1 data variable (dataframe or table column name). Default = NULL,
sankeyAxis2	Sankey axis 2 data variable (dataframe or table column name).Default = NULL,
sankeyAxis1Label	Sankey axis 1 title data variable (dataframe or table column name).Default = "axis1Label",
sankeyAxis2Label	Sankey axis 2 title variable (dataframe or table column name).Default = "axis2Label",
sankeyGroupColor	Which axis variables will be used to color flow paths (One of the sankey axis). Default = NULL,
printFig	Whether plot should be printed or not. Default = T,
fileName	File name for plot to be saved. Default = "chart",
title	Figure title. Default = NULL
dirOutputs	Output directory to save figure. Default = paste(getwd()),"/outputs/Charts",sep Default = "")
folderName	Foldername within output directory. Default=NULL,
figWidth	Figure width. Default = 9,
figHeight	Figure height. Default = 7,
pdfpng	Whether to save plot as pdf or png. Choice between "pdf" or "png". Default = "png",
sankeyLabelsOn	Turn on labels for sankey stratum categories. "1" or "0". Default =1
colOrder1	Order for sankey column 1. Default = NULL,
colOrderName1	Column name with sankey variables for column order 1. Default = NULL,
colOrder2	Order for sankey column 1. Default = NULL,
colOrderName2	Column name with sankey variables for column order 1. Default = NULL,
pointsOn	Include points on lines. Default = 1
pointsSize	ISize of points on line. Default = 4
paletteRev	Default =T
forceFacets	Default =F. When you have one facet only and want to show that.

## Value

Returns the formatted data used to produce chart



---

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_mmetis",
  regionCompare = 1, regionCompareOnly = 0, scenarioCompareOnly = 0,
  useNewLabels = 0, sizeBarLines = 0, sizeLines = 1.5,
  folderName = "folderNameDefault", nameAppend = "",
  scensSelect = "All", colOrder1 = NULL, colOrderName1 = NULL,
  colOrder2 = NULL, colOrderName2 = NULL, scaleRange = NULL,
  xScenCompFacetLabelSize = 35)
```

## 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 xScenSelectYears plot. Default is <code>c("2015", "2030", "2050", "2100")</code>
paramsSelect	Default = "All". If desired dplyr::select a subset of parameters to analyze from the full list of parameters: <code>c("# energy", "energyPrimaryByFuelEJ", "energyPrimaryRefLiqProdEJ", "energyFinalConsumBySecEJ", "energyFinalByFuelBuildEJ", "energyFinalSubsecByFuelBuildEJ", "energyFinalSubsecByFuelIndusEJ", "energyFinalSubsecByFuelResidEJ")</code>

```

"energyPrimaryByFuelMTOE","energyPrimaryRefLiqProdMTOE","en-
ergyFinalConsumBySecMTOE","energyFinalbyFuelMTOE","energyFinalSubsecByFuelTran
"energyFinalSubsecByFuelBuildMTOE","energyFinalSubsecByFuelIndusM-
TOE","energyFinalSubsecBySectorBuildMTOE","energyPrimaryByFu-
elTWh","energyPrimaryRefLiqProdTWh","energyFinalConsumBySecTWh","energyFinalb
"energyFinalSubsecByFuelBuildTWh","energyFinalSubsecByFuelIndusTWh","energyFinal
# electricity "elecByTechTWh","elecCapByFuel","elecFinalBySecTWh","elecFinalByFuelT
# transport "transportPassengerVMTByMode","transportFreightVMT-
ByMode","transportPassengerVMTByFuel","transportFreightVMTBy-
Fuel", # water "watConsumBySec","watWithdrawBySec","watWith-
drawByCrop","watBioPhysCons","watIrrWithdrawBasin","watIrrConsBasin",
# socioecon "gdpPerCapita","gdp","gdpGrowthRate","pop", # ag
"agProdbyIrrRfd","agProdBiomass","agProdForest","agProdByCrop",
# land "landIrrRfd","landAlloc","landAllocByCrop", # emissions "emiss-
LUC","emissCO2BySector","emissCO2NonCO2BySectorGWP5","emissCO2NonCO2By
"emissNonCO2ByResProdGWP5","emissTotalFFIBySec","emissMethaneBySource",
"emissCO2BySectorNonCO2GWP5","emissCO2BySectorNonCO2GWP5LUC",
"emissTotalBySec","emissCO2BySectorNoBio")

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_mmetis" from metis.colors()$pal_mmetis
regionCompare Default =1. To turn of regional comparison make 0.
regionCompareOnly
    Default 0. If set to 1, will only run comparison plots and not individual
scenarioCompareOnly
    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
folderName Default ="folderNameDefault"
nameAppend Default =" "
scensSelect Default = "All". Select regions to create charts for.
colOrder1 Default = NULL,
colOrderName1 Default = NULL,
colOrder2 Default = NULL,
colOrderName2 Default = NULL,
scaleRange Default NULL. Dataframe with columns param, maxScale, minScale to
indicate maximum and minumum values for a parameter scale.
xScenCompFacetLabelSize
    Default = 2

```

## 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

*metis.colors*

---

## 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\_mmetis"
- "pal\_16"
- "pal\_Basic"
- "pal\_hot"
- "pal\_wet"
- "pal\_div\_wet"
- "pal\_div\_RdBl"
- "pal\_div\_GrBr"
- "pal\_div\_BluRd"
- "pal\_div\_RdBlu"
- "pal\_green"
- "pal\_div\_BrGn"
- "pal\_div\_BlRd"
- "pal\_sankey"
- "pal\_spectral"
- "pal\_ScarcityCat"

## Value

A list of color palettes.

## Examples

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

---

mettis.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, regionName = "region",
  subRegShape = NULL, subRegShpFolder = NULL, subRegShpFile = NULL,
  subRegCol = NULL, subRegType = "subRegType", aggType = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""), folderName = NULL,
  nameAppend = "", labelsSize = 1.2, paramsSelect = "All",
  scenariosSelect = "All", sqliteUSE = F,
  sqliteDBNamePath = paste(getwd(), "/outputs/Grids/gridMetis.sqlite",
  sep = ""))
```

**Arguments**

grid	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",
regionName	Default = "region"
subRegShape	Default=NULL. shapefile over which grid data is to be aggregated.
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", sep Default=""),
subRegCol	Default= NULL. Suggested for states "name",
subRegType	Default="subRegType". Eg. "states", "basins" etc.
aggType	Default=NULL. Aggregation method to be used. Either "vol" or "depth" depending on the type of data provided.
dirOutputs	Default=paste(getwd(), "/outputs", sep Default=""),
folderName	Default = NULL
nameAppend	Default="",
labelsSize	Default =1.2. Label size for the region names for the gridoverlay plot.
paramsSelect	Default = "All"
scenariosSelect	Default = "All"
sqliteUSE	Default = T,
sqliteDBNamePath	Default = paste(getwd(), "/outputs/Grids/gridMetis.sqlite", sep = "")

**Value**

A table with data by polygon ID for each shapefile provided

---

metis.gridByPoly	<i>metis.gridByPoly</i>
------------------	-------------------------

---

**Description**

This function finds the grids located within a given shapefiles regions

**Usage**

```
metis.gridByPoly(gridDataTables = NULL, shape = NULL,
  shapeFolder = NULL, shapeFile = NULL, colName = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""),
  fname = "gridByPoly", folderName = "folderNameDefault",
  saveFile = F)
```

**Arguments**

gridDataTables	Default = NULL. Full path to grid file.
shape	Default = NULL,
shapeFolder	Default = NULL,
shapeFile	Default = NULL,
colName	Default = NULL,
dirOutputs	Default = paste(getwd(), "/outputs", sep=""),
fname	Default = "gridByPoly"
folderName	Default = "folderNameDefault",
saveFile	Default = F. If want csv output then change to T

**Value**

Prints out graphic

---

metis.io	<i>metis.io</i>
----------	-----------------

---

**Description**

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

**Usage**

```
metis.io(ioTable0 = NULL, useIntensity = 0, A0 = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""), nameAppend = "",
  figWidth = 9, figHeight = 7, sankeyLabelAbsPlots = 1,
  combSubRegionPlots = 1, folderName = "folderNameDefault",
  pdfpng = "png", sankeyAxis1Label = "From", sankeyAxis2Label = "To",
  plotSankeys = T)
```

**Arguments**

ioTable0	Initial ioTable. Must have columns: supplySubSector,total,export and cap. Each supply sector should also have imports. Default = NULL,
useIntensity	Boolean to use given intensity or not. Default is set to 0.
A0	Intensity matrix. Default Null.
dirOutputs	Default =paste(getwd(), "/outputs", sep=""),
nameAppend	Modified intensity matrix. Default =NULL,
figWidth	Default = 9,
figHeight	Default = 7,
sankeyLabelAbsPlots	Default = 1
combSubRegionPlots	Default = 1
folderName	Default = "folderNameDefault"

pdfpng	Save IO figures as pdf or png. Type=String. Options: 'pdf' or 'png'. Default = 'png'.
sankeyAxis1Label	Set left-hand side label of sankey diagram. Type=String. Default = 'From'.
sankeyAxis2Label	Set right-hand side label of sankey diagram. Type=String. Default = 'To'.
plotSankeys	Default = T

## Value

A table with data by polygon ID for each shapefile provided

---

metis.map	<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 = F, 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/Maps", sep = ""), folderName = NULL, facetFreeScale = F,
  facetRows = NA, facetCols = 3, facetBGColor = "grey30",
  facetLabelColor = "white", facetLabelSize = 1.5, alpha = 1,
  fillcolorNA = "gray", fillshowNA = NA, fillcolorNULL = "gray",
  facetsON = T, panelLabel = NULL, multiFacetRows = NULL,
  multiFacetCols = NULL, mapTitle = NULL, mapTitleSize = 1,
  numeric2Cat_list = NULL, catParam = NULL, innerMargins = c(0, 0, 0,
  0), outerMargins = c(0.01, 0.01, 0.01, 0.01),
  legendSingleColorOn = T, legendSingleValue = 0,
  legendSingleColor = "white")
```

**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 = ""),



```

folderName      Default = NULL,
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
fillshowNA      Default =NA
fillcolorNULL   Default =NULL
facetsON        Default =F,
panelLabel      Default = NULL,
multiFacetRows  Default=NULL,
multiFacetCols  Default=NULL,
mapTitle        Default=NULL
mapTitleSize    Default=1
numeric2Cat_list
                Default=NULL,
catParam        Default=NULL
innerMargins    Default =c(0,0,0,0), # bottom, left, top, right
outerMargins    Default =c(0.01,0.01,0.01,0.01) # bottom, left, top, right
legendSingleColorOn
                Default=F
legendSingleValue
                Default=0
legendSingleColor
                Default="white"

```

### Value

Returns the formatted data used to produce chart

---

metis.mapsProcess	<i>metis.mapsProcess</i>
-------------------	--------------------------

---

### Description

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

## Usage

```
metis.mapsProcess(polygonDataTables = NULL, gridDataTables = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""),
  folderName = "folderNameDefault", xRange = "All", labels = F,
  labelsSize = 1.2, subRegShape = NULL, subRegShpFolder = NULL,
  subRegShpFile = NULL, subRegCol = NULL, dirNameAppend = "",
  nameAppend = "", legendOutsideSingle = T,
  legendOutsidePosition = NULL, legendPosition = NULL,
  legendFixedBreaks = 5, legendTitleSize0 = 2, legendTextSize0 = 1,
  legendTitleSizeI = 1.5, legendTextSizeI = 1, animateOn = T,
  fps = 1, scenRef = NULL, extension = F, boundaryRegShape = NULL,
  boundaryRegShpFolder = NULL, boundaryRegShpFile = NULL,
  boundaryRegCol = NULL, boundaryRegionsSelect = NULL,
  extendedLabels = T, extendedFillColor = "grey75",
  extendedBGColor = "lightblue1", extendedHighLightColor = "cornsilk1",
  extendedLabelsColor = "grey30", extendedLabelSize = 0.7,
  extendedShape = NULL, extendedShapeCol = NULL, expandPercent = 3,
  projX = "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0",
  figWidth = 6, figHeight = 7, scaleRange = NULL,
  paramsSelect = "All", indivScenarios = "All", scensSelect = "All",
  GCMRCPSSPPol = F, multiFacetCols = "scenarioGCM",
  multiFacetRows = "scenarioRCP", legendOutsideMulti = T,
  legendPositionMulti = NULL, legendTitleSizeMulti = NULL,
  legendTextSizeAnim = NULL, legendTextSizeMulti = NULL,
  refGCM = NULL, refRCP = NULL, chosenRefMeanYears = NULL,
  mapTitleSize = 0.5, facetLabelSizeMulti = 3,
  facetLabelSizeGCMRCP = 1.5, numeric2Cat_list = NULL, diffOn = F,
  frameShow = T, pdfpng = "png", fillcolorNA = "gray",
  fillshowNA = NA, fillcolorNULL = "gray", legendSingleColorOn = T,
  legendSingleValue = 0, legendSingleColor = "white")
```

## Arguments

polygonDataTables	Default = NULL,
gridDataTables	Default = NULL,
dirOutputs	Default = paste(getwd(), "/outputs", sep = ""),
folderName	Default = "folderNameDefault",
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",
dirNameAppend	Default = ""
nameAppend	Default = ""

```

legendOutsideSingle
    Default =F, Single plots by default have legends inside. This can be
    moved out if wanted.
legendOutsidePosition
    Default = NULL, # "right","left","top","bottom", "center"
legendPosition Default = NULL, # c("RIGHT','top') - RIGHT LEFT TOP BOTTOM
legendFixedBreaks
    Default = "5",
legendTitleSize0
    Default = 2,
legendTextSize0
    Default =1,
legendTitleSizeI
    Default = 1,
legendTextSizeI
    Default =0.5,
animateOn      Default = T,
fps            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,
extendedLabels Default = T
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"
figWidth	Default =9
figHeight	Default =7
scaleRange	Default NULL. Dataframe with columns param, maxScale, minScale to indicate maximum and minimum values for a parameter scale.
paramsSelect	Default ="All"
indvScenarios	Default ="All", If only want to run single scenarios without comparing with others
scensSelect	Default ="All", choose scenarios from data
GCMRCPSSPPol	Default = F,
multiFacetCols	Default ="scenarioRCP",
multiFacetRows	Default ="scenarioGCM",
legendOutsideMulti	Default = NULL,
legendPositionMulti	Default = NULL,
legendTitleSizeMulti	Default = NULL,
legendTextSizeAnim	Default = NULL,
legendTextSizeMulti	Default = NULL,
refGCM	Default = NULL , eg. "gfdl-esm2m"
refRCP	Default = NULL , eg. "rcp2p6"
chosenRefMeanYears	Default=NULL
mapTitleSize	Default=0.5
facetLabelSizeMulti	Default =3
facetLabelSizeGCMRCP	Default =1.5
numeric2Cat_list	Default=NULL,
diffOn	Default = F. Whether to calculate diff values between scenarios.
frameShow	Default = T. Whether to plot frame around maps and facets.
pdfpng	Save IO figures as pdf or png. Type=String. Options: 'pdf' or 'png'. Default = 'png'
fillcolorNA	Default="gray",
fillshowNA	Default=NA,
fillcolorNULL	Default="gray"
legendSingleColorOn	Default=F,
legendSingleValue	Default =0,
legendSingleColor	Default="white"

**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 = "NA", demeterScenario = "NA",
  demeterTimesteps = seq(from = 2005, to = 2100, by = 5),
  demeterUnits = "NA", tethysFolder = "NA", tethysScenario = "NA",
  tethysUnits = "NA", tethysFiles = c("wddom", "wdelec", "wdirr",
    "wdliv", "wdmfg", "wdmin", "wdnonag", "wdtotal"),
  copySingleTethysScenbyXanthos = NULL, xanthosFolder = "NA",
  xanthosFiles = "NA", xanthosScenarioAssign = "NA",
  xanthosCoordinatesPath = "NA", xanthosGridAreaHeCsPath = "NA",
  scarcityXanthosRollMeanWindow = 10, spanLowess = 0.25,
  popFolder = "NA", popFiles = "NA", biaFolder = "NA",
  biaFiles = "NA", popUnits = "NA", dirOutputs = paste(getwd(),
    "/outputs", sep = ""), reReadData = 1, gridMetisData = paste(getwd(),
    "/outputs/Grids/gridMetis.RData", sep = ""), sqliteUSE = F,
  sqliteDBNamePath = paste(getwd(), "/outputs/Grids/gridMetis.sqlite",
    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")
copySingleTethysScenbyXanthos	Default=NULL,
xanthosFolder	Xanthos Folder Path
xanthosFiles	Xanthos Files to Read
xanthosScenarioAssign	Default "NA". Scenario name if testing single scenario.
xanthosCoordinatesPath	paste(getwd(), "/dataFiles/grids/xanthosCoords/coordinates.csv", sep = "")

```

xanthosGridAreaHechsPath
    =paste(getwd(),"/dataFiles/grids/xanthosRunsChris/reference/Grid_Areas_ID.csv",sep="")
scarcityXanthosRollMeanWindow
    Default = 10,
spanLowess
    Default = 0.25
popFolder
    Default = i-paste(getwd(),"/dataFiles/grids/griddedIDsPop/",sep="")
popFiles
    Default = i-"grid_pop_map"
biaFolder
    Default = i-paste(getwd(),"/dataFiles/grids/griddedIDsbia/",sep="")
biaFiles
    Default = i-"grid_bia_map"
popUnits
    Default = i-"person"
dirOutputs
    Default =paste(getwd(),"/outputs",sep=""),
reReadData
    Default =1,
gridMetisData
    Default = paste(dirOutputs, "/Grids/gridMetis.RData", sep = "")
sqliteUSE
    Default = T,
sqliteDBNamePath
    Default = paste(getwd(),"/outputs/Grids/gridMetis.sqlite", sep = "")

```

### Value

A table with data by polygon ID for each shapefile provided

---

metis.printPdfPng	<i>metis.printPdfPng</i>
-------------------	--------------------------

---

### Description

This function prints figure to pdf or png.

### Usage

```

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

```

### Arguments

figure	Default=NULL. Figure to be printed
dir	Default = getwd(). Directory to print figure
filename	Default = "plot". File name
figWidth	Default=13.
figHeight	Default=9.
pdfpng	Default="png". Either "pdf" or "png"

### Value

Prints out graphic

---

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 = NULL, gcamdatabaseName = NULL,
  queryxml = "metisQueries.xml", queryPath = paste(getwd(),
    "/dataFiles/gcam", sep = ""), scenOrigNames = NULL,
  scenNewNames = NULL, reReadData = T, dataProj = "dataProj.proj",
  dataProjPath = paste(getwd(), "/outputs", sep = ""),
  dirOutputs = paste(getwd(), "/outputs", sep = ""),
  regionsSelect = NULL, queriesSelect = "All", paramsSelect = "All")
```

## Arguments

<code>gcamdatabasePath</code>	Path to gcam database folder
<code>gcamdatabaseName</code>	Name of gcam database
<code>queryxml</code>	Name of the query.xml file. By default it is "metisQueries.xml"
<code>queryPath</code>	Folder that contains the query.xml file. By default it is the same folder as specified by gcamdatabasePath
<code>scenOrigNames</code>	Original Scenarios names in GCAM database in a string vector. For example c('scenario1','scenario2').
<code>scenNewNames</code>	New Names which may be shorter and more useful for figures etc. Default will use Original Names. For example c('scenario1','scenario2')
<code>reReadData</code>	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.
<code>dataProj</code>	Optional. A default 'dataProj.proj' is produced if no .Proj file is specified.
<code>dataProjPath</code>	Folder that contains the dataProj or where it will be produced. By default it is the same folder as specified by gcamdatabasePath
<code>dirOutputs</code>	Full path to directory for outputs
<code>regionsSelect</code>	The regions to analyze in a vector. Example c('Colombia','Argentina'). Full list: c(USA, Africa_Eastern, Africa_Northern, Africa_Southern, Africa_Western, Australia_NZ, Brazil, Canada Central America and Caribbean, Central Asia, China, EU-12, EU-15, Europe_Eastern, Europe_Non_EU, European Free Trade Association, India, Indonesia, Japan, Mexico, Middle East, Pakistan, Russia, South Africa, South America_Northern, South America_Southern, South Asia, South Korea, Southeast Asia,

**queriesSelect** Default = "All". Predetermined subsets or a vector of queries to read from the queryxml for example predetermined subsets would be c('water','energy') or selection of queries would be c("Total final energy by aggregate end-use sector", "Population by region"). The queries must be available in the queryxml file. Queryset names include: c("water", "energy", "land", "emissions", "ag", "socioecon", "transport") Current list of queries for each set include:

water

- "water withdrawals by crop"
- "water withdrawals by water mapping source"
- "water consumption by water mapping source"
- "water withdrawals by sector"
- "water consumption by sector"
- "biophysical water demand by crop type and land region"

energy

- "primary energy consumption by region (direct equivalent) ORDERED SUBSECTORS"
- "Electricity generation by aggregate technology ORDERED SUBSECTORS"
- "Final energy by detailed end-use sector and fuel"
- "total final energy by aggregate sector"
- "refined liquids production by subsector"
- "building final energy by fuel"
- "industry final energy by fuel"
- "building final energy by subsector"
- "transport final energy by fuel"
- "transport final energy by mode and fuel"

land

- "land allocation by crop and water source",
- "aggregated land allocation",
- "land allocation by crop"

emissions

- "nonCO2 emissions by resource production",
- "nonCO2 emissions by sector"
- "Land Use Change Emission (future)"
- "CO2 emissions by sector (no bio)"
- "CO2 emissions by sector"

ag

- "Ag Production by Crop Type"
- "ag production by tech"

socioecon

- "GDP MER by region"
- "GDP per capita MER by region"
- "Population by region"

transport

- "transport service output by mode"



- "transport service output by tech (new)"

**paramsSelect** Default = "All". If desired dplyr::select a subset of paramaters to analyze from the full list of parameters: c(# Energy "energyPrimaryByFuelEJ", "energyPrimaryRefLiqProdEJ", "energyFinalConsumBySecEJ", "energyFinalByFuelB", "energyFinalSubsecByFuelBuildEJ", "energyFinalSubsecByFuelIndusEJ", "energyFinalSubs", "energyPrimaryByFuelMTOE", "energyPrimaryRefLiqProdMTOE", "energyFinalConsumBySecMTOE", "energyFinalbyFuelMTOE", "energyFinalSubsecByFuelTran", "energyFinalSubsecByFuelBuildMTOE", "energyFinalSubsecByFuelIndusMTOE", "energyFinalSubsecBySectorBuildMTOE", "energyPrimaryByFuelTWh", "energyPrimaryRefLiqProdTWh", "energyFinalConsumBySecTWh", "energyFinalb", "energyFinalSubsecByFuelBuildTWh", "energyFinalSubsecByFuelIndusTWh", "energyFinal", # Electricity "elecByTechTWh", "elecCapByFuel", "elecFinalBySecTWh", "elecFinalByFuelT", # Transport "transportPassengerVMTByMode", "transportFreightVMTByMode", "transportPassengerVMTByFuel", "transportFreightVMTByFuel", # Water "watConsumBySec", "watWithdrawBySec", "watWithdrawByCrop", "watBioPhysCons", "watIrrWithdrawBasin", "watIrrConsBasin", # Socio-economics "gdpPerCapita", "gdp", "gdpGrowthRate", "pop", # Agriculture "agProdbyIrrRfd", "agProdBiomass", "agProdForest", "agProdByCrop", # Land use "landIrrRfd", "landAlloc", "landAllocByCrop", # Emissions "emissLUC", "emissCO2BySector", "emissCO2NonCO2BySectorGWP5", "emissNonCO2ByResProdGWP5", "emissTotalFFIBySec", "emissMethaneBySource", "emissCO2BySectorNonCO2GWP5", "emissCO2BySectorNonCO2GWP5LUC", "emissTotalBySec", "emissCO2BySectorNoBio")

## 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.

# Index

- \*Topic **assumptions**
  - metis.assumptions, 2
- \*Topic **bubble**,
  - metis.chart, 6
- \*Topic **charts**,
  - metis.chart, 6
  - metis.chartsProcess, 9
  - metis.map, 15
  - metis.mapsProcess, 17
  - metis.printPdfPng, 22
- \*Topic **colors**,
  - metis.colors, 11
- \*Topic **database**,
  - metis.boundaries, 4
  - metis.grid2poly, 12
  - metis.io, 14
  - metis.prepGrid, 21
  - metis.readgcam, 23
- \*Topic **diffplots**,
  - metis.chart, 6
- \*Topic **diffplots**
  - metis.chartsProcess, 9
  - metis.map, 15
  - metis.mapsProcess, 17
  - metis.printPdfPng, 22
- \*Topic **downscale**,
  - metis.bia, 3
- \*Topic **downscaled**
  - metis.bia, 3
- \*Topic **downscaling**,
  - metis.bia, 3
- \*Topic **electricity**,
  - metis.bia, 3
- \*Topic **gcam**,
  - metis.bia, 3
  - metis.boundaries, 4
  - metis.grid2poly, 12
  - metis.io, 14
  - metis.prepGrid, 21
  - metis.readgcam, 23
- \*Topic **gcam**
  - metis.boundaries, 4
  - metis.grid2poly, 12
- metis.io, 14
- metis.prepGrid, 21
- metis.readgcam, 23
- \*Topic **generation**,
  - metis.bia, 3
- \*Topic **grid**,
  - metis.gridByPoly, 13
- \*Topic **gridded**,
  - metis.bia, 3
- \*Topic **palette**
  - metis.colors, 11
- \*Topic **polygon**
  - metis.gridByPoly, 13
- \*Topic **query**
  - metis.boundaries, 4
  - metis.grid2poly, 12
  - metis.io, 14
  - metis.prepGrid, 21
  - metis.readgcam, 23
- \*Topic **sankey**.
  - metis.chart, 6
- \*Topic **shape**,
  - metis.gridByPoly, 13
- metis, 2
- metis-package (metis), 2
- metis.assumptions, 2
- metis.bia, 3
- metis.boundaries, 4
- metis.chart, 6
- metis.chartsProcess, 9
- metis.colors, 11
- metis.grid2poly, 12
- metis.gridByPoly, 13
- metis.io, 14
- metis.map, 15
- metis.mapsProcess, 17
- metis.prepGrid, 21
- metis.printPdfPng, 22
- metis.readgcam, 23