Package: mrmagpie (via r-universe)

September 2, 2024

Type Package

Title madrat based MAgPIE Input Data Library

Version 1.52.0

Date 2024-07-04

Description Provides functions for MAgPIE country and cellular input data generation.

License LGPL-3 | file LICENSE

URL https://github.com/pik-piam/mrmagpie, https://doi.org/10.5281/zenodo.4319612

BugReports https://github.com/pik-piam/mrmagpie/issues

Depends madrat (>= 2.8.0), magclass (>= 3.17), mrcommons (>= 1.41.0), mrlandcore (>= 1.1.0), mrland (>= 0.59.0), mrsoil (>= 2.0.0), mrwater (>= 1.13.0), R (>= 3.5.0)

Imports abind, class, digest, dplyr, ggplot2, lpjclass, luplot (>= 3.64.0), magpiesets, mstools (>= 0.6.0), ncdf4, pbapply, raster, readxl, stats, stringr, terra (>= 1.7.18), tidyr, withr

Suggests covr, knitr, rmarkdown, zip

VignetteBuilder knitr

Encoding UTF-8

LazyData no

RoxygenNote 7.3.1

Repository https://pik-piam.r-universe.dev

RemoteUrl https://github.com/pik-piam/mrmagpie

RemoteRef HEAD

RemoteSha 9ec4da527c2f068a08d21ff0265a0d3668dea146

Contents

`mrmagpie-package` *mrmagpie: madrat based MAgPIE Input Data Library*

Description

Provides functions for MAgPIE country and cellular input data generation.

Author(s)

Maintainer: Kristine Karstens <`karstens@pik-potsdam.de`>

Authors:

- Jan Philipp Dietrich <`dietrich@pik-potsdam.de`>
- David Chen
- Michael Windisch
- Marcos Alves
- Felicitas Beier <`beier@pik-potsdam.de`>
- Alexandre Köberle <`alexkob@pik-potsdam.de`>
- Patrick v. Jeetze <`vjeetze@pik-potsdam.de`>
- Abhijeet Mishra <`mishra@pik-potsdam.de`>
- Florian Humpenoeder <`humpenoeder@pik-potsdam.de`>
- Pascal Sauer <`pascal.sauer@pik-potsdam.de`>

See Also

Useful links:

- https://github.com/pik-piam/mrmagpie
- doi:10.5281/zenodo.4319612
- Report bugs at https://github.com/pik-piam/mrmagpie/issues

---

calcAfforestationMask *calcAfforestationMask*

Description

Afforestation mask for where afforestation possible

Usage

```
calcAfforestationMask(subtype, cells = "lpjcell")
```

Arguments

subtype	afforestation mask sub type
cells	"magpiecell" or "lpjcell"

Value

magpie object in cellular resolution

Author(s)

David Chen, Florian Humpenoeder

Examples

```
## Not run:
calcOutput("AfforestationMask", subtype = "noboreal", aggregate = FALSE)

## End(Not run)
```

calcAgeClassDistribution

calcAgeClassDistribution

Description

This function calculates the share of each age class in secondary forests in each MAgPIE simulation cluster based on Global Forest Age Dataset from Poulter et al. 2019

Usage

```
calcAgeClassDistribution(cells = "lpjcell")
```

Arguments

cells	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

magpie object in cluster resolution

Author(s)

Abhijeet Mishra, Felicitas Beier

Examples

```
## Not run:
calcOutput("AgeClassDistribution", aggregate = FALSE)

## End(Not run)
```

calcAreaActuallyIrrigated

calcAreaActuallyIrrigated

Description

retrieves irrigated crop area from croparea intialization

Usage

```
calcAreaActuallyIrrigated(aggregationlevel = "iso", selectyears = "y1995")
```

Arguments

`aggregationlevel`	default is iso
`selectyears`	select years

Value

magpie object with results on cellular or iso country level

Author(s)

Felicitas Beier

Examples

```
## Not run:
calcOutput("AreaActuallyIrrigated")

## End(Not run)
```

---

calcAreaEquippedForIrrigation

calcAreaEquippedForIrrigation

Description

Calculates the area equipped for irrigation based on LU2v2 or Mehta data sets. For LUH2v2, it assumes, that all cropland irrigated in the last 20 years at least once is equipped for irrigation. Mehta et al. (2022) directly report Global Area Equipped for Irrigation for the years 1900-2015

Usage

```
calcAreaEquippedForIrrigation(
  cellular = FALSE,
  cells = "lpjcell",
  selectyears = "past"
)
```

Arguments

`cellular`	if TRUE: 0.5 degree resolution returned
`cells`	number of cells to be returned: magpiecell (59199), lpjcell (67420)
`selectyears`	default on "past"

Value

List of magpie objects with results on country/cellular level, weight on country level, unit and description.

Author(s)

Benjamin Leon Bodirsky, Kristine Karstens, Felicitas Beier

See Also

[calcLanduseInitialisation()]

Examples

```
## Not run:
calcOutput("AreaEquippedForIrrigation", source = "LUH2v2", cellular = TRUE, aggregate = FALSE)

## End(Not run)
```

calcAvlLandSi *calcAvlLandSi*

Description

Extracts si0 and nsi0 areas based on Ramankutty dataset

Usage

```
calcAvlLandSi(cells = "lpjcell")
```

Arguments

`cells`	magpiecell (59199 cells) or lpjcell (67420 cells)

Value

magpie object in cellular resolution

Author(s)

Felicitas Beier

Examples

```
## Not run:
calcOutput("AvlLandSi", aggregate = FALSE)

## End(Not run)
```

`calcBinnedLsuDensity` *calcRangeSoilCarbonHist*

Description

calculates soil carbon for rangelands

Usage

```
calcBinnedLsuDensity(
  breaks = c(seq(0, 2, 0.1), 2.25, 2.5),
  labels = c(0, 0.2, 0.2, 0.4, 0.4, 0.6, 0.6, 0.8, 0.8, 1, 1, 1.2, 1.2, 1.4, 1.4, 1.6,
    1.6, 1.8, 1.8, 2, 2, 2.5),
  years = 1995
)
```

Arguments

`breaks`	Binning breaks
`labels`	Binning labels
`years`	years where data should binned

Value

Magpie object with lsu per cell.

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("BinnedLsuDensity ", breaks, labels, years)

## End(Not run)
```

calcBphEffect *calcBphEffect*

Description

Biogeophysical temperature change of afforestation (degree C). File is based on observation datasets of Bright et al. 2017 and Duveiller et al. 2018

Usage

```
calcBphEffect(cells = "lpjcell")
```

Arguments

cells	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

magpie object in cellular resolution

Author(s)

Michael Windisch, Felicitas Beier

Examples

```
## Not run:
calcOutput("BphEffect", aggregate = FALSE)

## End(Not run)
```

calcBphMask *calcBphMask*

Description

Mask of Datapoints of biogeophysical temperature change of afforestation (degree C) to be used as weight. File is based on observation datasets of Bright et al. 2017 and Duveiller et al. 2018

Usage

```
calcBphMask(cells = "lpjcell")
```

Arguments

cells	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

magpie object in cellular resolution

Author(s)

Michael Windisch, Felicitas Beier

Examples

```
## Not run:
calcOutput("BphMask", aggregate = FALSE)

## End(Not run)
```

---

calcBphTCRE *calcBphTCRE*

Description

Transient Climate Response to accumulated doubling of CO2. File based on CMIP5 +1perc CO2 per year experiment. To be used in the translation to carbon equivalents of BphEffect

Usage

```
calcBphTCRE(cells = "lpjcell")
```

Arguments

`cells`	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

magpie object in cellular resolution

Author(s)

Michael Windisch, Felicitas Beier

Examples

```
## Not run:
calcOutput("BphTCRE", aggregate = FALSE)

## End(Not run)
```

calcCarbon *calcCarbon*

Description

This function extracts carbon densities from LPJ to MAgPIE

Usage

```
calcCarbon(
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb"),
  climatetype = "GSWP3-W5E5:historical",
  cells = "lpjcell"
)
```

Arguments

`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`climatetype`	Switch between different GCM climate scenarios
`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

magpie object in cellular resolution

Author(s)

Kristine Karstens, Patrick v. Jeetze

Examples

```
## Not run:
calcOutput("Carbon", aggregate = FALSE)

## End(Not run)
```

calcCarbonTests *calcCarbonTests*

Description

This function extracts carbon densities from LPJ to MAgPIE

Usage

```
calcCarbonTests(
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb"),
  climatetype = "GSWP3-W5E5:historical",
  stage = "raw"
)
```

Arguments

`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`climatetype`	Switch between different GCM climate scenarios
`stage`	Switch for raw data or harmonization

Value

magpie object in cellular resolution

Author(s)

Kristine Karstens, Florian Humpenoeder

Examples

```
## Not run:
calcOutput("CarbonTests", aggregate = FALSE)

## End(Not run)
```

calcCellCountryFraction *calcCellCountryFraction*

Description

cell fraction belonging to a country based on LanduseInitialisation

Usage

```
calcCellCountryFraction(cells = "lpjcell")
```

Arguments

cells	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

Clustered MAgPIE object on requested resolution

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
calcOutput("calcCellCountryFraction", aggregate = FALSE)

## End(Not run)
```

calcCluster *calcCluster*

Description

This function calculates the aggregation mapping for a given cluster methodology

Usage

```
calcCluster(
  ctype,
  regionscode = madrat::regionscode(),
  seed = 42,
  weight = NULL,
  lpjml = c(natveg = "LPJmL4", crop = "LPJmL5"),
  clusterdata = "yield_airrig"
)
```

Arguments

`ctype` aggregation clustering type, which is a combination of a single letter, indicating the cluster methodology, and a number, indicating the number of resulting clusters. Available methodologies are hierarchical clustering (h), normalized k-means clustering (n), combined hierarchical/normalized k-means clustering (c) and manual setting for clusters per region (m). In the combined clustering hierarchical clustering is used to determine the cluster distribution among regions whereasit is manually set for the m type. Both use normalized k-means for the clustering within a region.

`regionscode` regionscode of the regional mapping to be used. Must agree with the regionscode of the mapping mentioned in the madrat config! Can be retrieved via `regionscode()`.

`seed` Seed for Random Number Generation. If set to NULL it is chosen automatically, if set to an integer it will always return the same pseudo-random numbers (useful to get identical clusters under identical inputs for n and c clustering)

`weight` Should specific regions be resolved with more or less detail? Values > 1 mean higher share, < 1 lower share e.g. cfg$cluster_weight <- c(LAM=2) means that a higher level of detail for region LAM if set to NULL all weights will be assumed to be 1 (examples: c(LAM=1.5,SSA=1.5,OAS=1.5), c(LAM=2,SSA=2,OAS=2))

`lpjml` defines LPJmL version for crop/grass and natveg specific inputs

`clusterdata` similarity data to be used to determine clusters: yield_airrig (current default) or yield_increment

Value

map from cells to clusters as data.frame

Author(s)

Jan Philipp Dietrich

Examples

```
## Not run:
calcOutput("Cluster", ctype = "c200", aggregate = FALSE)

## End(Not run)
```

`calcClusterBase` *calcClusterBase*

Description

Reads a series of MAgPIE files and combines them to a matrix which is then used for calculating a clustering.

Usage

```
calcClusterBase(
  clusterdata = "yield_airrig",
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb")
)
```

Arguments

`clusterdata`	similarity data to be used to determine clusters: yield_airrig (current default) or yield_increment
`lpjml`	defines LPJmL version for crop/grass and natveg specific inputs

Value

A matrix containing the data

Author(s)

Jan Philipp Dietrich, Felicitas Beier

See Also

`calcCluster`

`calcClusterHierarchical`

calcClusterHierarchical

Description

Performs MAgPIE hierarchical clustering and calculates corresponding spam relation matrix

As the creation of a clustering tree is very time consuming the function checks first in the input folder if the corresponding data already exists and if not it stores the tree information in the input folder for later use in the next execution of this function.

Usage

```
calcClusterHierarchical(
  regionscode,
  ncluster,
  lpjml = c(natveg = "LPJmL4", crop = "LPJmL5"),
  clusterdata = "yield_airrig",
  mode = "h",
  weight = NULL
)
```

Arguments

`regionscode` regionscode of the regional mapping to be used. Must agree with the regionscode of the mapping mentioned in the madrat config! Can be retrieved via `regionscode()`.

`ncluster` The desired total number of clusters.

`lpjml` defines LPJmL version for crop/grass and natveg specific inputs

`clusterdata` similarity data to be used to determine clusters: yield_airrig (current default) or yield_increment

`mode` Clustering type. At the moment you can choose between complete linkage clustering (h), single linkage clustering (s) and Ward clustering (w).

`weight` named vector with weighting factors for each region for the cluster distribution, e.g. weight=c(AFR=3,EUR=0.5). weight > 1 will grant more cluster to a region and weight < 1 less cluster than by default.

Value

A mapping between regions and clusters

Author(s)

Jan Philipp Dietrich

See Also

`calcCluster`, `calcClusterKMeans`

`calcClusterKMeans` *calcClusterKMeans*

Description

Performs MAgPIE kmeans clustering and calculates corresponding spam relation matrix

Usage

```
calcClusterKMeans(
  regionscode,
  ncluster,
  weight = NULL,
  cpr = NULL,
  seed = 42,
  lpjml = c(natveg = "LPJmL4", crop = "LPJmL5"),
  clusterdata = "yield_airrig"
)
```

Arguments

`regionscode` regionscode of the regional mapping to be used. Must agree with the regionscode of the mapping mentioned in the madrat config! Can be retrieved via `regionscode()`.

`ncluster` The desired total number of clusters.

`weight` named vector with weighting factors for each region for the cluster distribution, e.g. weight=c(AFR=3,EUR=0.5). weight > 1 will grant more cluster to a region and weight < 1 less cluster than by default.

`cpr` cells-per-region information as returned by toolClusterPerRegionManual. Weight and ncluster are ignored in case that cpr is provided!

`seed` a single value, interpreted as an integer, or NULL, to define seed for random calculations

`lpjml` defines LPJmL version for crop/grass and natveg specific inputs

`clusterdata` similarity data to be used to determine clusters: yield_airrig (current default) or yield_increment

Value

A mapping between regions and clusters

Author(s)

Jan Philipp Dietrich

See Also

`toolClusterPerRegionManual`, `calcClusterHierarchical`

`calcClusterTreeHierarchical` *calcClusterTreeHierarchical*

Description

calculates hierarchical clustering tree

Usage

```
calcClusterTreeHierarchical(
  regionscode,
  mode = "h",
  weight = NULL,
  lpjml = c(natveg = "LPJmL4", crop = "LPJmL5"),
  clusterdata = "yield_airrig"
)
```

Arguments

`regionscode` regionscode of the regional mapping to be used. Must agree with the regionscode of the mapping mentioned in the madrat config! Can be retrieved via `regionscode()`.

`mode` Clustering type. At the moment you can choose between complete linkage clustering (h), single linkage clustering (s) and Ward clustering (w).

`weight` named vector with weighting factors for each region for the cluster distribution, e.g. weight = c(AFR = 3, EUR = 0.5). weight > 1 will grant more cluster to a region and weight < 1 less cluster than by default.

`lpjml` defines LPJmL version for crop/grass and natveg specific inputs

`clusterdata` similarity data to be used to determine clusters: yield_airrig (current default) or yield_increment

Value

A spam relation matrix

Author(s)

Jan Philipp Dietrich

`calcCO2Atmosphere_new` *calcCO2Atmosphere_new*

Description

Disaggregate CO2 global atmospheric concentration to cellular level

Usage

```
calcCO2Atmosphere_new(
  subtype = "ISIMIP3b:ssp126",
  co2Evolution = "rising",
  cells = "lpjcell"
)
```

Arguments

`subtype` specify the version and scenario eg. "ISIMIP3b:ssp126"

`co2Evolution` Define ‘rising‘ for rising CO2 according to the climate scenario selected or ‘static‘ for stable CO2 at the last past time step level.

`cells` "magpiecell" or "lpjcell"

Value

magpie object in cellular resolution

Author(s)

Marcos Alves, Kristine Karstens

Examples

```
## Not run:
calcOutput("CO2Atmosphere_new", aggregate = FALSE, subtype, co2Evolution)

## End(Not run)
```

---

calcCollectEnvironmentData_new

calcCollectEnvironmentData_new

Description

Calculate climate, CO2 and soil environmental conditions on cellular level

Usage

```
calcCollectEnvironmentData_new(
  subtype = "ISIMIP3b:IPSL-CM6A-LR:ssp126:1965-2100",
  sar = 20,
  sel_feat = c("tas", "pr", "lwnet", "rsds", "CO2", "Ks", "Sf", "w_pwp", "w_fc", "w_sat",
    "hsg", "wet")
)
```

Arguments

`subtype`	Switch between different climate scenarios (default: "CRU_4") eg. "ISIMIP3b:IPSL-CM6A-LR:ssp126:1965-2100"
`sar`	Average range for smoothing annual variations
`sel_feat`	features names to be included in the output file

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("CollectEnvironmentData_new", subtype, sar = 20, sel_feat = "temp")

## End(Not run)
```

`calcCollectSoilCarbonLSU` *calcCollectSoilCarbonLSU*

Description

Calculate soil carbon stocks for different LSU and climate conditions

Usage

```
calcCollectSoilCarbonLSU(
  lsu_levels = c(seq(0, 2, 0.2), 2.5),
  lpjml = "LPJML5.2_pasture",
  climatemodel = "IPSL_CM6A_LR",
  scenario = "ssp126_co2_limN",
  sar = 20
)
```

Arguments

`lsu_levels`	Livestock unit levels in the source folder
`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`climatemodel`	Switch between different climate scenarios
`scenario`	scenario specifications (eg. ssp126_co2_limN)
`sar`	Average range for smoothing annual variations

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("CollectSoilCarbonLSU", lsu_levels = c(seq(0, 2, 0.2), 2.5), scenario)

## End(Not run)
```

calcCollectSoilCarbonPastr *calcCollectSoilCarbonPastr*

Description

calculates soil carbon content for pasture areas

Usage

```
calcCollectSoilCarbonPastr(
  past_mngmt = "me2",
  lpjml = "lpjml5p2_pasture",
  climatemodel = "IPSL_CM6A_LR",
  scenario = "ssp126_co2_limN",
  sar = 1
)
```

Arguments

past_mngmt	pasture areas management option
lpjml	Defines LPJmL version for crop/grass and natveg specific inputs
climatemodel	Switch between different climate scenarios (default: "CRU_4")
scenario	scenario specifications (eg. ssp126_co2_limN)
sar	Average range for smoothing annual variations

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("CollectSoilCarbonPastr", past_mngmt = "me2")

## End(Not run)
```

calcDegradationYieldReduction *calcDegradationYieldReduction*

Description

Function creates dummy file for including yield reduction coefficients to represent land degradation

Usage

```
calcDegradationYieldReduction(cells = "lpjcell")
```

Arguments

`cells`	number of halfdegree grid cells to be returned. Options: "magpiecell" (59199), "lpjcell" (67420)

Value

magpie object in cellular resolution

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
calcOutput("DegradationYieldReduction", aggregate = FALSE)

## End(Not run)
```

calcEFRRockstroem *calcEFRRockstroem*

Description

This function calculates environmental flow requirements (EFR) for MAgPIE retrieved from LPJmL monthly discharge and water availability following the definition of the planetary boundary in Rockström et al. 2023

Usage

```
calcEFRRockstroem(
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb"),
  climatetype = "GSWP3-W5E5:historical",
  stage = "harmonized2020",
  seasonality = "grper"
)
```

Arguments

`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`climatetype`	Switch between different climate scenarios
`stage`	Degree of processing: raw, smoothed, harmonized, harmonized2020
`seasonality`	grper (default): EFR in growing period per year; total: EFR throughout the year; monthly: monthly EFRs

Value

magpie object in cellular resolution

Author(s)

Felicitas Beier, Jens Heinke

Examples

```
## Not run:
calcOutput("EFRRockstroem", aggregate = FALSE)

## End(Not run)
```

calcEFRSmakthin *calcEFRSmakthin*

Description

This function calculates environmental flow requirements (EFR) for MAgPIE retrieved from LPJmL monthly discharge and water availability using the method of Smakthin et al. (2004)

Usage

```
calcEFRSmakthin(
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb"),
  climatetype = "GSWP3-W5E5:historical",
  stage = "harmonized2020",
  LFR_val = 0.1,
  HFR_LFR_less10 = 0.2,
  HFR_LFR_10_20 = 0.15,
  HFR_LFR_20_30 = 0.07,
  HFR_LFR_more30 = 0,
  seasonality = "grper",
  cells = "lpjcell"
)
```

Arguments

`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`climatetype`	Switch between different climate scenarios
`stage`	Degree of processing: raw, smoothed, harmonized, harmonized2020
`LFR_val`	Strictness of environmental flow requirements
`HFR_LFR_less10`	High flow requirements (share of total water for cells) with LFR<10percent of total water
`HFR_LFR_10_20`	High flow requirements (share of total water for cells) with 10percent < LFR < 20percent of total water
`HFR_LFR_20_30`	High flow requirements (share of total water for cells) with 20percent < LFR < 30percent of total water
`HFR_LFR_more30`	High flow requirements (share of total water for cells) with LFR>30percent of total water
`seasonality`	grper (default): EFR in growing period per year; total: EFR throughout the year; monthly: monthly EFRs
`cells`	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

magpie object in cellular resolution

Author(s)

Felicitas Beier, Abhijeet Mishra

Examples

```
## Not run:
calcOutput("EFRSmakthin", aggregate = FALSE)

## End(Not run)
```

calcEnvmtlFlow *calcEnvmtlFlow*

Description

This function calculates environmental flow requirements (EFR) for MAgPIE retrieved from LPJmL monthly discharge and water availability

Usage

```
calcEnvmtlFlow(
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb"),
  climatetype = "GSWP3-W5E5:historical",
  stage = "harmonized2020",
  seasonality = "grper"
)
```

Arguments

lpjml	Defines LPJmL version for crop/grass and natveg specific inputs
climatetype	Switch between different climate scenarios
stage	Degree of processing: raw, smoothed, harmonized, harmonized2020
seasonality	grper (default): EFR in growing period per year; total: EFR throughout the year; monthly: monthly EFRs

Value

magpie object in cellular resolution

Author(s)

Felicitas Beier

Examples

```
## Not run:
calcOutput("EnvmtlFlow", aggregate = FALSE)

## End(Not run)
```

calcFoodDemandGridded *calcFoodDemandGridded*

Description

Calculates grid-level food demand, note also includes food and feed

Usage

```
calcFoodDemandGridded(
  attribute = "dm",
  prod = "k",
  feed = TRUE,
  cells = "lpjcell"
)
```

Arguments

attribute	dm or calories ("ge") or other massbalance attribute
prod	for memory reasons
feed	whether to include feed demand in the gridded demand
cells	magpiecell or lpjcell (default)

Value

Gridded magpie object of food demand disaggregated by rural urban pop

Author(s)

David M Chen

Examples

```
## Not run:
calcOutput("FoodDemandGridded")

## End(Not run)
```

calcGCMClimate *calcGCMClimate*

Description

Disaggregate CO2 global atmospheric concentration to cellular level NOTE: This function will be depreciate soon, please use mrland::calcLPJmLClimate

Usage

```
calcGCMClimate(
  subtype = "ISIMIP3bv2:IPSL-CM6A-LR:ssp126:1850-2100:tas:annual_mean",
  smooth = 0,
  cells = "lpjcell"
)
```

Arguments

subtype	type of climate data to collect, consisting of data source, GDM, RCP, time period, variable and time resolution separated by ":"
smooth	set averaging value for smoothing trajectories
cells	number of halfdegree grid cells to be returned. Options: "magpiecell" (59199), "lpjcell" (67420)

Value

magpie object in cellular resolution

Author(s)

Marcos Alves, Kristine Karstens, Felicitas Beier

Examples

```
## Not run:
calcOutput("GCMClimate", subtype = "ISIMIP3b:IPSL-CM6A-LR:ssp126:1850-2100:tas:annual_mean")

## End(Not run)
```

calcGrasslandBiomass *calcGrasslandBiomass*

Description

Calculates pasture biomass demand for the historical period split between rangelands andmanaged pastures.

Usage

```
calcGrasslandBiomass(cells = "lpjcell")
```

Arguments

`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

Regional biomass demand

Author(s)

Marcos Alves

See Also

`calcOutput`, `calcFAOmassbalance`, `readSource`

Examples

```
## Not run:
calcOutput("GrasslandBiomass")

## End(Not run)
```

calcGrasslandsYields *calcGrasslandsYields*

Description

Calculates rangelands maximum output and managed pastures yields

Usage

```
calcGrasslandsYields(
  lpjml = "lpjml5p2_pasture",
  climatetype = "MRI-ESM2-0:ssp370",
  cells = "lpjcell",
  subtype = "/co2/Nreturn0p5",
  lsu_levels = c(seq(0, 2, 0.2), 2.5),
  past_mngmt = "mdef"
)
```

Arguments

lpjml	Defines LPJmL version for crop/grass and natveg specific inputs
climatetype	Global Circulation Model to be used
cells	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells
subtype	Switch between different climate scenarios
lsu_levels	Livestock unit levels in the source folder
past_mngmt	pasture areas management option

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("GrasslandsYields", lsu_levels, past_mngmt = "me2", subtype)

## End(Not run)
```

calcGrassPastureShare *calcGrassPastureShare*

Description

Calculate glassland shareas os pasture managed lands.

Usage

```
calcGrassPastureShare()
```

Value

List of magpie object with results on cluster level

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("GrassPastureShare")

## End(Not run)
```

`calcGrassSoilEmu` *calcGrassSoilEmu*

Description

Read files related to the training and optimization of the LPJml emulators.

Usage

```
calcGrassSoilEmu(
  subtype = "ISIMIP3b:IPSL_CM6A_LR:ssp126:1965_2100",
  model = "5f5fa2",
  mfile = "weights"
)
```

Arguments

`subtype`	Subtype of file to be opened. Subtypes available: 'weights', 'inputs', 'stddevs' and 'means'.
`model`	trained model ID
`mfile`	model file name

Value

Magpie objects with a diverse inforamtion

Author(s)

Marcos Alves

Examples

```
## Not run:
readSource("GrassSoilEmu",
  subtype = "ISIMIP3b:IPSL_CM6A_LR:ssp126:1965_2100",
  model = "5f5fa2", mfile = "weights"
)

## End(Not run)
```

calcGridPop *calcGridPop*

Description

Past and future (SSP1-5) population based on HYDE3.2 and Jones & O'Neill (2016) Data is scaled to match WDI data from calcPopulation NOTE that some scaling factors for the future (for small countries Gambia and Djibouti) are off, data read in is 50

Usage

```
calcGridPop(
  source = "ISIMIP",
  subtype = "all",
  cellular = TRUE,
  cells = "lpjcell",
  FiveYear = TRUE,
  scale = TRUE,
  harmonize_until = 2015,
  urban = FALSE
)
```

Arguments

`source` default source (ISIMIP) or Gao data (readGridPopGao) which is split into urban and rural.

`subtype` time horizon to be returned. Options: past (1965-2005), future (2005-2010) or all (divergence starts at year in harmonize_until)

`cellular` if true: half degree grid cell data returned

`cells` number of halfdegree grid cells to be returned. Options: "magpiecell" (59199), "lpjcell" (67420)

`FiveYear` TRUE for 5 year time steps, otherwise yearly from source

`scale` if true: scales sum of gridded values to match country level totals

`harmonize_until` harmonization year until which SSPs diverge (default: 2015)

`urban` TRUE to return only urban gridded population based on iso share

Value

Population in millions.

Author(s)

David Chen, Felicitas Beier

Examples

```
## Not run:
calcOutput("GridPop", aggregate = FALSE)

## End(Not run)
```

`calcInitialLsu` *calcInitialLsu*

Description

Loads the LSU that provides the maximum grass harvest as a initial values for MAgPIE

Usage

```
calcInitialLsu(model = "f41f19be67")
```

Arguments

`model`	Grass harvest machine learning model ID

Value

MAgPIE objects with optimal lsu on a cellular level.

Author(s)

Marcos Alves

Examples

```
## Not run:
calOutput("InitialLsu", model = "f41f19be67")

## End(Not run)
```

calcIrrigation *calcIrrigation*

Description

This function extracts irrigation water (airrig: water applied additionally to rainfall) from LPJmL for MAgPIE

Usage

```
calcIrrigation(
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb"),
  climatetype = "GSWP3-W5E5:historical",
  cells = "lpjcell",
  rainfedweight = 0.01
)
```

Arguments

lpjml	Defines LPJmL version for crop/grass and natveg specific inputs
climatetype	Switch between different climate scenarios
cells	Number of cells to be returned: "magpiecell" for 59199 cells or "lpjcell" for 67420 cells
rainfedweight	For clustering airrig is weighted with cropland_irrigated + rainfedweight * cropland_rainfed (default: 0.01)

Value

magpie object in cellular resolution

Author(s)

Felicitas Beier, Abhijeet Mishra

Examples

```
## Not run:
calcOutput("Irrigation", aggregate = FALSE)

## End(Not run)
```

calcLabourProdImpact *calcLabourProdImpact*

Description

Labour productivity impacts

Usage

```
calcLabourProdImpact(
  timestep = "5year",
  subtype = "Orlov",
  cellular = TRUE,
  cells = "lpjcell"
)
```

Arguments

timestep	5year or yearly
subtype	data source comes from
cellular	cellular is true
cells	"magpiecell" or "lpjcell"

Value

List of magpie objects with results on 0.5deg grid level, weights based on production value, unit (ratio) and description.

Author(s)

David Chen

calcLabourProdImpactEmu *calcLabourProdImpactEmu*

Description

Spatial and temporal aggr. of labour productivity impacts from climate change emulated by LAMACLIMA based on method of Orlov et al. 2019. Economics of Disasters and Climate Change, 3(3), 191-211.

Usage

```
calcLabourProdImpactEmu(
  timestep = "5year",
  cellular = TRUE,
  subtype = "impact",
  cells = "lpjcell"
)
```

Arguments

`timestep`	5-year or yearly
`cellular`	cellular is true
`subtype`	impact for rcp based laborprod decrease, relief for LCLM based relief of impact
`cells`	"magpiecell" or "lpjcell"

Value

List of magpie object of gridded (0.5) labour productivity as percentage of full labour prod 1

Author(s)

Michael Windisch, Florian Humpenöder

`calcLivestockDistribution` *calcLivestockDistribution*

Description

Disaggregate Livestock estimates based on the GLW3 dataset.

Usage

```
calcLivestockDistribution(cells = "lpjcell")
```

Arguments

`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

MAgPIE objects with livestock numbers on a cellular level.

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("LivestockDistribution")

## End(Not run)
```

---

calcLsuDensityHist *calcLsuDensityHist*

Description

Calculate livestock historical livestock densities

Usage

```
calcLsuDensityHist(disagg_type = "grassland", cells = "lpjcell")
```

Arguments

`disagg_type`	select the disaggregaton weights for biomass production (can be either grassland or livestock)
`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

List of magpie object with results on cluster level

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("LsuDensityHist")

## End(Not run)
```

`calcLuh2SideLayers` *calcLuh2SideLayers*

Description

Function extracts biodiversity data for LUH2 land cover types

Usage

```
calcLuh2SideLayers(cells = "lpjcell")
```

Arguments

`cells`	number of cells to be returned: magpiecell (59199), lpjcell (67420)

Value

magpie object in cellular resolution

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
calcOutput("Luh2SideLayers", aggregate = FALSE)

## End(Not run)
```

`calcMAPSPAM` *calcMAPSPAM*

Description

MAPSPAM data

Usage

```
calcMAPSPAM(subtype = "physical")
```

Arguments

`subtype`	it can be either "physical" or "harvested" area

Value

magpie object in cellular resolution

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
calcOutput("MAPSPAM", subtype = "physical", aggregate = FALSE)

## End(Not run)
```

`calcMaxPastureSuit` *calcMaxPastureSuit*

Description

Calculate maximum glassland suitable for pasture management based on population and aridity criteria.

Usage

```
calcMaxPastureSuit(
  climatetype = "MRI-ESM2-0:ssp126",
  lpjml = "LPJmL4_for_MAgPIE_44ac93de",
  cells = "lpjcell"
)
```

Arguments

`climatetype`	Switch between different climate scenarios
`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`cells`	number of halfdegree grid cells to be returned. Options: "magpiecell" (59199), "lpjcell" (67420)

Value

List of magpie object with results on cluster level

Author(s)

Marcos Alves, Kristine Karstens, Alexandre Köberle

Examples

```
## Not run:
calcOutput("MaxPastureSuit")

## End(Not run)
```

---

calcNonLocalProduction *calcNonLocalProduction*

Description

Calculates grid-level amount of food that would need to be transported, assuming that food produced in the grid cell is first consumed by local population i.e. amount of food greater than local rural demand, split into that which feeds the local urban population, and that which exceeds total local demand and is available to export

Usage

```
calcNonLocalProduction(cells = "lpjcell")
```

Arguments

cells	magpiecell or lpjcell (default)

Author(s)

David M Chen

Examples

```
## Not run:
calcOutput("NonLocalTransport")

## End(Not run)
```

---

calcNpiNdcAdAolcPol *calcNpiNdcAdAolcPol*

Description

Function creates dummy NPI/NDC policies

Usage

```
calcNpiNdcAdAolcPol(cells = "lpjcell")
```

Arguments

`cells`	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

magpie object in cellular resolution

Author(s)

Patrick v. Jeetze, Michael Windisch

Examples

```
## Not run:
calcOutput("NpiNdcAdAolcPol", aggregate = FALSE)

## End(Not run)
```

`calcNpiNdcAffPol` *calcNpiNdcAffPol*

Description

Function creates dummy NPI/NDC policies

Usage

```
calcNpiNdcAffPol(cells = "lpjcell")
```

Arguments

`cells`	lpjcell for 67420 cells or magpiecell for 59199 cells

Value

magpie object in cellular resolution

Author(s)

Patrick v. Jeetze, Michael Windisch

Examples

```
## Not run:
calcOutput("NpiNdcAffPol", aggregate = FALSE)

## End(Not run)
```

calcPackagingMarketingCosts *calcPackagingMarketingCosts*

Description

calculates per-ton marketing and packaging costs for food that leaves a cell Currnetly assume expert guess 50 USD / ton of packaging/marketing costs (100 USD/t in model, of which half is already in GTAP)

Usage

```
calcPackagingMarketingCosts()
```

Value

List of magpie objects with results on country level, weight on country level, unit and description.

Author(s)

David M Chen

calcPastrMngtLevels *calcPastrMngtLevels*

Description

Calculates managed pasture potential yields for different combinations of SSP+RCP and grassland management options

Usage

```
calcPastrMngtLevels(
  climatetype = "MRI-ESM2-0:ssp370",
  options = c("brazil_1", "brazil_2", "brazil_4"),
  cost_level = c(1, 2, 3),
  cells = "lpjcell"
)
```

Arguments

climatetype	SSP+RCP combination
options	Management options simulated by LPJml
cost_level	level cost for different past management options
cells	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

magpie object in 0.5 degree resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("PastrMngtLevels", ssps, options)

## End(Not run)
```

`calcPastrTauHist` *calcPastrTauHist*

Description

Calculates managed pastures Tau based on FAO yield trends for 1995.

Usage

```
calcPastrTauHist(past_mngmt = "mdef", cells = "lpjcell")
```

Arguments

`past_mngmt`	Pasture management reference yield
`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

List of magpie objects with results on country level, weight on country level, unit and description.

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("PastrTauHist", past_mngmt)

## End(Not run)
```

calcPastr_new *calcPastr_new*

Description

Calculates managed pasture yields

Usage

```
calcPastr_new(
  past_mngmt = "me2",
  lpjml = "lpjml5p2_pasture",
  climatetype = "MRI-ESM2-0:ssp370",
  scenario = "/co2/Nreturn0p5/limN",
  cells = "lpjcell"
)
```

Arguments

past_mngmt	pasture areas management option
lpjml	Defines LPJmL version for crop/grass and natveg specific inputs
climatetype	Switch between different climate scenarios (default: "CRU_4")
scenario	specify ssp scenario
cells	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("Pastr_new", past_mngmt = "me2", lpjml = "LPJml_pastr", climatetype)

## End(Not run)
```

calcPeatland *calcPeatland*

Description

This function calculates degraded and intact peatland area at cell level. The function takes degraded and intact peatland area from the Global Peatland Database (GPD) at the national level and downscales the peatland area to grid cell level using gridded potential peatland area. The GPD has been provided by Alexandra Barthelmes. The potential peatland area has been provided by Leifeld_2018 (DOI 10.1038/s41467-018-03406-6).

Usage

```
calcPeatland(subtype = "degraded", cells = "lpjcell")
```

Arguments

subtype	degraded (default) or intact
cells	"magpiecell" or "lpjcell"

Value

magpie object in cellular resolution

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
calcOutput("Peatland", aggregate = FALSE)

## End(Not run)
```

calcPeatland2 *calcPeatland2*

Description

This function calculates degraded and intact peatland area at cell level. The function takes degraded and intact peatland area from the Global Peatland Database 2022 (GPD2022) at the national level and downscales the peatland area to grid cell level using gridded peatland area from the Global Peatland Map 2.0 (GPM2) The data has been provided by Alexandra Barthelmes.

Usage

```
calcPeatland2(cells = "magpiecell", countryLevel = FALSE)
```

Arguments

`cells`	number of cells to be returned: magpiecell (59199), lpjcell (67420)
`countryLevel`	Whether output shall be at country level. Requires aggregate=FALSE in calcOutput.

Value

magpie object in cellular resolution

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
calcOutput("Peatland2", aggregate = FALSE)

## End(Not run)
```

---

calcPotentialForestArea

calcPotentialForestArea

Description

Calculates the area than can be potentially covered by forests, based on environmental conditions.

Usage

```
calcPotentialForestArea(
  refData = "lpj",
  countryLevel = FALSE,
  cells = "lpjcell",
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de"),
  climatetype = "MRI-ESM2-0:ssp370"
)
```

Arguments

`refData` Determines the reference data that the estimated potential forest area is derived from (currently only "lpj")

`countryLevel` Whether output shall be at country level. Requires aggregate=FALSE in calcOutput.

`cells` magpiecell (59199 cells) or lpjcell (67420 cells)

`lpjml` Defines LPJmL version for crop/grass and natveg specific inputs. Only relevant, if refData = "lpj".

`climatetype` Switch between different GCM climate scenarios. Only relevant, if refData = "lpj".

Value

magpie object in cellular resolution

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
calcOutput("PotentialForestArea", aggregate = FALSE)

## End(Not run)
```

`calcRangelandsMaxNew` *calcRangelandsMaxNew*

Description

Calculates rangelands maximum output

Usage

```
calcRangelandsMaxNew(
  lsuLevels = c(seq(0, 2.2, 0.2), 2.5),
  lpjml = "lpjml5p2_pasture",
  climatetype = "MRI-ESM2-0:ssp370",
  scenario = "/co2/Nreturn0p5/limN",
  report = "harvest",
  cells = "lpjcell"
)
```

Arguments

`lsuLevels`	Livestock unit levels in the source folder
`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`climatetype`	Switch between different climate scenarios (default: "CRU_4")
`scenario`	specify ssp scenario
`report`	Either 'harvest' or 'lsu/ha' controlling what values are output by the function.
`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("ContGrazMax_new", lsuLevels = 0, lpjml, climatetype, report)

## End(Not run)
```

`calcRangeSoilCarbonHist` *calcRangeSoilCarbonHist*

Description

calculates soil carbon for rangelands

Usage

```
calcRangeSoilCarbonHist(
  subtype = "ISIMIP3b:IPSL-CM6A-LR:ssp126:1965-2100",
  lpjml,
  model = "9eaf9b"
)
```

Arguments

`subtype`	subtypes
`lpjml`	lpjml version
`model`	trained model ID

Value

List of magpie objects with results on country level, weight on country level, unit and description.

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("GrassSoilCarbonHist ", subtype, model)

## End(Not run)
```

---

calcRRLayer *calcRRLayer*

Description

Function extracts range-rarity as used for biodiversity loss

Usage

```
calcRRLayer(cells = "lpjcell")
```

Arguments

cells	number of cells to be returned: magpiecell (59199), lpjcell (67420)

Value

magpie object in cellular resolution

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
calcOutput("RRLayer", aggregate = FALSE)

## End(Not run)
```

calcScaledPastSoilCarbon

calcScaledPastSoilCarbon

Description

calculates the mean and sd of the scaled pasture soil carbon dataset

Usage

```
calcScaledPastSoilCarbon(
  lsu_levels = c(seq(0, 2, 0.2), 2.5),
  lpjml = "LPJML5.2_pasture",
  climatetype = "IPSL_CM6A_LR",
  scenario = "ssp126_co2_limN",
  sar = 20,
  aggr = FALSE
)
```

Arguments

lsu_levels	Livestock unit levels in the source folder
lpjml	Defines LPJmL version for crop/grass and natveg specific inputs
climatetype	Switch between different climate scenarios (default: "CRU_4")
scenario	scenario specifications (eg. ssp126_co2_limN)
sar	Average range for smoothing annual variations
aggr	aggregation level

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("ScaledPastSoilCarbon", lsu_levels = c(seq(0, 2, 0.2), 2.5), scenario)

## End(Not run)
```

calcScaleEnvironmentData_new
calcScaleEnvironmentData_new

Description

Scale climate, CO2 and soil environmental conditions on cellular level

Usage

```
calcScaleEnvironmentData_new(
  subtype = "ISIMIP3b:IPSL-CM6A-LR:ssp126:1965-2100",
  aggr = FALSE,
  sar = 20,
 sel_feat = c("tas", "pr", "lwnet", "rsds", "CO2", "Ks", "Sf", "w_pwp", "w_fc", "w_sat",
    "hsg")
)
```

Arguments

subtype	Switch between different climate scenarios
aggr	aggregation level
sar	Average range for smoothing annual variations
sel_feat	features names to be included in the output file

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("ScaleEnvironmentData_new", climatetype = "HadGEM2_ES:rcp8p5:co2", sar = 20, sel_feat)

## End(Not run)
```

calcSCScalingFactors *calcSCScalingFactors*

Description

calculates the mean and sd of the scaled pasture soil carbon dataset

Usage

```
calcSCScalingFactors(
  lsu_levels = c(seq(0, 2, 0.2), 2.5),
  lpjml = "LPJML5.2_pasture",
  climatetype = "IPSL_CM6A_LR",
  scenario = "ssp126_co2_limN",
  sar = 20
)
```

Arguments

lsu_levels	Livestock unit levels in the source folder
lpjml	Defines LPJmL version for crop/grass and natveg specific inputs
climatetype	Switch between different climate scenarios (default: "CRU_4")
scenario	scenario specifications (eg. ssp126_co2_limN)
sar	Average range for smoothing annual variations

Value

magpie object in cellular resolution

Author(s)

Marcos Alves

Examples

```
## Not run:
calcOutput("SCScalingFactors", lsu_levels = c(seq(0, 2, 0.2), 2.5), scenario)

## End(Not run)
```

calcSoilCharacteristics

calcSoilCharacteristics

Description

Calculate Soil Characteristics based on a HWDS soil classification map

Usage

```
calcSoilCharacteristics()
```

Value

Magpie objects with results on cellular level.

Author(s)

Marcos Alves

See Also

readSoilClassification,

Examples

```
## Not run:
readSource("SoilClassification", subtype = "HWSD.soil", convert = "onlycorrect")

## End(Not run)
```

calcSOMinitialsiationPools

calcSOMinitialsiationPools

Description

calculates Soil Organic Matter Pool, accounting for the management history as initialisation to magpie

Usage

```
calcSOMinitialsiationPools(cells = "lpjcell")
```

Arguments

`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells

Value

List of magpie object with results on country or cellular level, weight on cellular level, unit and description.

Author(s)

Benjamin Leon Bodirsky, Kristine Karstens

Examples

```
## Not run:
calcOutput("SOMinitialsiationPools")

## End(Not run)
```

calcTopsoilCarbon *calcTopsoilCarbon*

Description

This function extracts topsoil carbon densities from LPJ to MAgPIE

Usage

```
calcTopsoilCarbon(
  cells = "lpjcell",
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb"),
  climatetype = "GSWP3-W5E5:historical"
)
```

Arguments

`cells`	"magpiecell" for 59199 cells or "lpjcell" for 67420 cells
`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`climatetype`	Switch between different GCM climate scenarios

Value

magpie object in cellular resolution

Author(s)

Kristine Karstens

Examples

```
## Not run:
calcOutput("TopsoilCarbon", aggregate = FALSE)

## End(Not run)
```

---

calcTransportCosts *calcTransportCosts*

Description

calculates country-level transport costs from GTAP total transport costs, cellular production, and cellular travel time

Usage

```
calcTransportCosts(transport = "all", gtapVersion = "9")
```

Arguments

`transport`	"all" or "nonlocal". "all" means all production incurs transport costs, while "nonlocal" sees only production greater than local rural consumption with transport costs
`gtapVersion`	"9" or "81"

Value

List of magpie objects with results on country level, weight on country level, unit and description.

Author(s)

David M Chen

See Also

[calcTransportTime()], [calcGTAPTotalTransportCosts()]

Examples

```
## Not run:
calcOutput("TransportCosts_new")

## End(Not run)
```

calcTransportDistance *calcTransportDistance*

Description

Function extracts travel time to major cities in minutes This function now deprecated - use calcTransportTime instead

Usage

```
calcTransportDistance()
```

Value

magpie object in cellular resolution

Author(s)

David Chen

Examples

```
## Not run:
calcOutput("TransportDistance", aggregate = FALSE)

## End(Not run)
```

calcTransportTime *calcTransportTime*

Description

Function extracts travel time to major cities in minutes

Usage

```
calcTransportTime(subtype = "cities50", cells = "lpjcell")
```

Arguments

`subtype`	currently only cities of 5, 20, or 50 thousand people ("cities5", "cities20", "cities50") or ports of various sizes ("portsLarge|Medium|Small|VerySmall|Any")
`cells`	number of cells to be returned: magpiecell (59199), lpjcell (67420)

Value

magpie object in cellular resolution

Author(s)

David Chen

Examples

```
## Not run:
calcOutput("TransportTime", aggregate = FALSE)

## End(Not run)
```

`convertGPD` *convertGPD*

Description

convert GPD

Usage

```
convertGPD(x)
```

Arguments

`x`	magpie object provided by the read function

Value

List of magpie objects with results on iso level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
readSource("GPD", convert = TRUE)

## End(Not run)
```

convertGPD2022 *convertGPD2022*

Description

convert GPD2022

Usage

```
convertGPD2022(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects with results on iso level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
readSource("GPD2022", convert = TRUE)

## End(Not run)
```

correctAvlLandSi *correctAvlLandSi*

Description

Read Available Land Si

Usage

```
correctAvlLandSi(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

David Chen

See Also

readAvlLandSi

Examples

```
## Not run:
readSource("AvlLandSi", convert = "onlycorrect")

## End(Not run)
```

---

correctBendingTheCurve *readBendingTheCurve*

Description

Read bending the curve data

Usage

```
correctBendingTheCurve(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Patrick v. Jeetze, Michael Windisch

Examples

```
## Not run:
readSource("BendingTheCurve", subtype = "rr_layer", convert = "onlycorrect")

## End(Not run)
```

correctGCMClimate *correctGCMClimate*

Description

Correct GCMs climate variables NOTE: This function will be depreciate soon, please use mrland::correctLPJmLClimate

Usage

```
correctGCMClimate(x)
```

Arguments

x	magpie object provided by the read function

Value

Magpie objects with results on cellular level, weight, unit and description.

Author(s)

Marcos Alves, Felicitas Beier

See Also

readGCMClimate

Examples

```
## Not run:
readSource("GCMClimate", subtype, convert="onlycorrect")

## End(Not run)
```

correctGFAD *correctGFAD*

Description

Correct Global Forest Age Dataset

Usage

```
correctGFAD(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Abhijeet Mishra, Felicitas Beier

See Also

readGFAD

Examples

```
## Not run:
  readSource("GFAD", convert="onlycorrect")

## End(Not run)
```

---

correctGPM2 *correctGPM2*

Description

correct peatland area

Usage

```
correctGPM2(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
  readSource("GPM2", convert="onlycorrect")

## End(Not run)
```

---

correctGrassYldEmu *correctGrassYldEmu*

Description

Correct files related to the training and optimization of the LPJml emulators

Usage

```
correctGrassYldEmu(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects.

Author(s)

Marcos Alves

See Also

readGrassYldEmu

Examples

```
## Not run:
 readSource("GrassYldEmu", subtype = "GrassYldEmu:20f33a2280.weights", convert="onlycorrect")

## End(Not run)
```

correctLabourProdImpactEmu

correctLabourProdImpactEmu

Description

correct labour productivity impacts from climate change emulated by the LAMACLIMA project based on method of Orlov et al. 2019. Economics of Disasters and Climate Change, 3(3), 191-211.

Usage

```
correctLabourProdImpactEmu(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Michael Windisch

See Also

readLabourProdImpactEmu

Examples

```
## Not run:
  readSource("LabourProdImpactEmu", convert="onlycorrect")

## End(Not run)
```

correctLeifeld2018 *correctLeifeld2018*

Description

correct potential peatland area from Leifeld2018

Usage

```
correctLeifeld2018(x)
```

Arguments

x	magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
  readSource("Leifeld2018", convert="onlycorrect")

## End(Not run)
```

correctMehta2024 *correctMehta2024*

Description

correct Global Area Equipped for Irrigation Dataset 1900-2015 from Mehta et al., 2024

Usage

```
correctMehta2024(x)
```

Arguments

x	magpie object provided by the read function

Value

magpie object in cellular resolution

Author(s)

Felicitas Beier

Examples

```
## Not run:
  readSource("Mehta2024", convert="onlycorrect")

## End(Not run)
```

`correctRamankutty` *correctRamankutty*

Description

Read Available Land Si

Usage

```
correctRamankutty(x)
```

Arguments

`x`	magpie object provided by the read function

Value

magpie object

Author(s)

Felicitas Beier

See Also

`readRamankutty`

Examples

```
## Not run:
  readSource("Ramankutty", convert="onlycorrect")

## End(Not run)
```

`correctSoilClassification`

correctSoilClassification

Description

Correct soil classification

Usage

```
correctSoilClassification(x)
```

Arguments

x	Magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Marcos Alves, Kristine Karstens

See Also

`readSoilClassification`,

Examples

```
## Not run:
readSource("SoilClassification", subtype = "HWSD.soil", convert = "onlycorrect")

## End(Not run)
```

`correctTransportDistance`

correctTransportDistance

Description

Read transport distance file

Usage

```
correctTransportDistance(x)
```

Arguments

`x` magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

David Chen

See Also

`readTransportDistance`

Examples

```
## Not run:
readSource("TransportDistance", convert = "onlycorrect")

## End(Not run)
```

`correctWindisch2021` *correctWindisch2021*

Description

correct data to calculate BphEffect, BphTCRE or BphMask depending on the chosen subtype. BphEffect: Biogeophysical temperature change of afforestation (degree C). (File is based on observation datasets of Bright et al. 2017 and Duveiller et al. 2018). BphMask: Mask of Datapoints of biogeophysical temperature change of afforestation (degree C) to be used as weight. (File is based on observation datasets of Bright et al. 2017 and Duveiller et al. 2018). BphTCRE: Transient Climate Response to accumulated doubling of CO2. (File is based on CMIP5 +1perc CO2 per year experiment. To be used in the translation to carbon equivalents of BphEffect)

Usage

```
correctWindisch2021(x)
```

Arguments

`x` magpie object provided by the read function

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Felicitas Beier, Michael Windisch

See Also

readWindisch2021

Examples

```
## Not run:
  readSource("Windisch2021", convert="onlycorrect")

## End(Not run)
```

downloadCO2Atmosphere_new *downloadCO2Atmosphere_new*

Description

Download CO2 atm. inputs used for Lpjml runs

Usage

```
downloadCO2Atmosphere_new(subtype = "ISIMIP3b:ssp126")
```

Arguments

subtype	Switch between different inputs (eg. "ISIMIP3b:IPSL-CM6A-LR:historical:1850-2014:tas") It consists of GCM version, climate model, scenario and variable.

Value

metadata entry

Author(s)

Marcos Alves

Examples

```
## Not run: readSource("CO2Atmosphere_new",convert="onlycorrect")
```

`downloadGCMClimate` *downloadGCMClimate*

Description

Download GCM climate input used for Lpjml runs NOTE: This function will be depreciate soon, please use mrland::downloadLPJmLClimate

Usage

```
downloadGCMClimate(subtype = "ISIMIP3b:IPSL-CM6A-LR:ssp126:2015-2100:tas")
```

Arguments

`subtype`	Switch between different inputs (e.g. "ISIMIP3b:IPSL-CM6A-LR:historical:1850-2014:tas") Argument consists of GCM version, climate model, scenario and variable, separated by ":"

Value

metadata entry

Author(s)

Marcos Alves

Examples

```
## Not run:
readSource("GCMClimate", convert = "onlycorrect")

## End(Not run)
```

`downloadMAPSPAM` *downloadMAPSPAM*

Description

Downloads the MAP-SPAM (SPAM) data set for harvested and physical croparea

Usage

```
downloadMAPSPAM()
```

Value

raw files for MAPSPAM

Author(s)

Edna J. Molina Bacca

See Also

[downloadSource()]

Examples

```
## Not run:
a <- download("downloadMAPSPAM")

## End(Not run)
```

---

downloadMehta2024 *downloadMehta2024*

Description

download Global Area Equipped for Irrigation Dataset 1900-2015 from Mehta et al. (2024). Gridded dataset is created based on (sub-)national statistics from FAOSTAT, AQUASTAT, EUROSTAT and country's census data downscaled using two alternative gridded irrigation maps (GMIA from Siebert et al. 2013 and Meier et al. 2018)

Usage

```
downloadMehta2024(subtype = "GMIA")
```

Arguments

subtype	data subtype to be downloaded. Subtypes available: 'GMIA': gridded base map for downscaling from Stefan et al. (2013). Global Map of Irrigation Areas version 5. 'Meier2018': gridded base map for downscaling from Meier, et al. (2018). Global Irrigated Areas.

Author(s)

Felicitas Beier

See Also

[downloadSource()] [readMehta2024()]

Examples

```
## Not run:
a <- downloadSource()

## End(Not run)
```

`downloadRamankutty` *downloadRamankutty*

Description

download Ramankutty available land si (Source: Ramankutty N, Foley JA, Norman J and Mc-Sweeney K (2002) The global distribution of cultivable lands: current patterns and sensitivity to possible climate change. Global Ecology and Biogeography, 11, 377-392.)

Usage

```
downloadRamankutty()
```

Author(s)

Felicitas Beier

See Also

`downloadSource` `readRamankutty`

Examples

```
## Not run:  a <- downloadSource()
```

`downloadTravelTimeNelson2019` *downloadTravelTimeNelson2019*

Description

download Nelson 2019 paper

Usage

```
downloadTravelTimeNelson2019()
```

Author(s)

David M Chen

fullCELLULARMAGPIE *fullCELLULARMAGPIE*

Description

Function that produces the complete cellular data set required for running the MAgPIE model.

Usage

```
fullCELLULARMAGPIE(
  rev = numeric_version("0.1"),
  dev = "",
  ctype = "c200",
  climatetype = "MRI-ESM2-0:ssp370",
  lpjml = c(natveg = "LPJmL4_for_MAgPIE_44ac93de", crop =
    "ggcmi_phase3_nchecks_9ca735cb", grass = "lpjml5p2_pasture"),
  isimip = NULL,
  clusterweight = NULL,
  emu_id = NULL
)
```

Arguments

`rev`	data revision which should be used as input (numeric_version).
`dev`	development suffix to distinguish development versions for the same data revision. This can be useful to distinguish parallel lines of development.
`ctype`	aggregation clustering type, which is a combination of a single letter, indicating the cluster methodology, and a number, indicating the number of resulting clusters. Available methodologies are - hierarchical clustering (h), - normalized k-means clustering (n) and - combined hierarchical/normalized k-means clustering (c). In the latter hierarchical clustering is used to determine the cluster distribution among regions whereas normalized k-means is used for the clustering within a region.
`climatetype`	Global Circulation Model to be used
`lpjml`	Defines LPJmL version for crop/grass and natveg specific inputs
`isimip`	Defines isimip crop model input which replace maiz, tece, rice_pro and soybean
`clusterweight`	Should specific regions be resolved with more or less detail? Values > 1 mean higher share, < 1 lower share e.g. cfg$clusterweight <- c(LAM=2) means that a higher level of detail for region LAM if set to NULL all weights will be assumed to be 1. Examples: c(LAM=1.5,SSA=1.5,OAS=1.5) or c(LAM=2,SSA=2,OAS=2) `setConfig` (e.g. for setting the mainfolder if not already set properly).
`emu_id`	Pasture Soil carbon emulator ID

Author(s)

Kristine Karstens, Jan Philipp Dietrich

See Also

readSource,getCalculations,calcOutput,setConfig

Examples

```
## Not run:
retrieveData("CELLULARMAGPIE", rev = numeric_version("12"),
             mainfolder = "pathtowhereallfilesarestored")

## End(Not run)
```

---

readAvlLandSi *readAvl_Land_Si*

Description

Read si0 and nsi0 areas based on Ramankutty dataset"

Usage

```
readAvlLandSi()
```

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

David Chen

Examples

```
## Not run:
  readSource("AvlLandSi", convert="onlycorrect")

## End(Not run)
```

readBendingTheCurve *readBendingTheCurve*

Description

Read bending the curve data

Usage

```
readBendingTheCurve(subtype)
```

Arguments

subtype	Data used in the Bending the Curve initiative. Type "rr_layer" for the range-size rarity layer and "luh2_side_layers" for the LUH2 Side Layers.

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
  readSource("BendingTheCurve", subtype="rr_layer", convert="onlycorrect")

## End(Not run)
```

readCO2Atmosphere_new *readCO2Atmosphere*

Description

Read CO2 global atmospheric concentration

Usage

```
readCO2Atmosphere_new(subtype = "ISIMIP3b:ssp126")
```

Arguments

subtype	Switch between different inputs

Value

Magpie objects with results on global level

Author(s)

Marcos Alves, Kristine Karstens

Examples

```
## Not run:
readSource("CO2Atmosphere_new", subtype = "ISIMIP3b:ssp126", convert = FALSE)

## End(Not run)
```

readFishCatches *readFishCatches*

Description

Read soil classification data used as input for lpjml

Usage

```
readFishCatches()
```

Value

Magpie object with results on cellular level for soil types

Author(s)

Marcos Alves, Kristine Karstens

Examples

```
## Not run:
readSource("SoilClassification")

## End(Not run)
```

`readGCMClimate` *readGCMClimate*

Description

Read Climate data used as LPJmL inputs into MAgPIE objects NOTE: This function will be depreciate soon, please use mrland::readLPJmLClimate

Usage

```
readGCMClimate(
  subtype = "ISIMIP3bv2:IPSL-CM6A-LR:historical:1850-2014:tas",
  subset = "annual_mean"
)
```

Arguments

`subtype` Switch between different inputs, e.g. "ISIMIP3b:IPSL-CM6A-LR:historical:1850-2014:tas" Available variables are: * tas - * wet - * per -

`subset` Switch between different subsets of the same subtype Available options are: "annual_mean", "annual_sum", "monthly_mean", "monthly_sum", "wet"

Value

MAgPIE objects with results on cellular level.

Author(s)

Marcos Alves, Kristine Karstens, Felicitas Beier

See Also

`readGCMClimate`

Examples

```
## Not run:
readSource("GCMClimate", subtype, convert = "onlycorrect")

## End(Not run)
```

readGFAD *readGFAD*

Description

Read GLobal Forest Age Dataset derived from MODIS and COPENICUS satellite data

Usage

```
readGFAD()
```

Value

magpie object in cellular resolution

Author(s)

Abhijeet Mishra, Felicitas Beier

Examples

```
## Not run:
readSource("GFAD", convert = "onlycorrect")

## End(Not run)
```

readGPD *readGPD*

Description

read GPD Data from the Global Peatland Database provided by Alexandra Barthelmes. The original xls file has been clean-up manually (country names). Turkey had two identical entries in the original xls file. Sources: "Inventory Reports and National Communications UNFCC 2014", "soil and peatland science", "European Mires Book" , "own estimates (incl. GIS data)",

Usage

```
readGPD()
```

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
readSource("GPD", convert = "onlycorrect")

## End(Not run)
```

---

readGPD2022 *readGPD2022*

Description

read x Data from the Global Peatland Database provided by Alexandra Barthelmes. The original xls file has been clean-up manually (country names). Turkey had two identical entries in the original xls file. Sources: "Inventory Reports and National Communications UNFCC 2014", "soil and peatland science", "European Mires Book" , "own estimates (incl. GIS data)",

Usage

```
readGPD2022()
```

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
readSource("x", convert = "onlycorrect")

## End(Not run)
```

---

readGPM2 *readGPM2*

Description

read peatland area from GPM2

Usage

```
readGPM2(subtype = "1km")
```

Arguments

`subtype`	resolution ("1km" or "500m")

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
readSource("GPM2", convert = "onlycorrect")

## End(Not run)
```

`readGrassSoilEmu` *readGrassSoilEmu*

Description

Read files related to the training and optimization of the LPJml emulators.

Usage

```
readGrassSoilEmu(
  subtype = "ISIMIP3b:IPSL_CM6A_LR:ssp126:1965_2100:5f5fa2:stddevs_lab"
)
```

Arguments

`subtype`	Subtype of file to be opened. Subtypes available: ’weights’, ’inputs’, ’stddevs’ and ’means’.

Value

Magpie objects with a diverse inforamtion

Author(s)

Marcos Alves

Examples

```
## Not run:
readSource("GrassSoilEmu",
  subtype =
    "ISIMIP3b:IPSL_CM6A_LR:ssp126:1965_2100:5f5fa2:weights", convert = F
)

## End(Not run)
```

---

readGrassYldEmu *readGrassYldEmu*

Description

Read files related to the training and optimization of the LPJml emulators.

Usage

```
readGrassYldEmu(subtype = "109325f71e.inputs")
```

Arguments

subtype	Subtype of file to be opened. Subtypes available: 'max_harvest', 'weights', 'inputs', 'stddevs' and 'means'.

Value

Magpie objects with a diverse inforamtion

Author(s)

Marcos Alves

Examples

```
## Not run:
readSource("GrassYldEmu", subtype = "109325f71e.inputs", convert="onlycorrect")

## End(Not run)
```

`readGridPopGao` *readGridPopGao*

Description

Read gridded population, by urban and rural, from Gao O'Neill and JOnes dataset, see https://www.cgd.ucar.edu/iam/modelin population-scenarios.html https://doi.org/10.7927/m30p-j498

Usage

```
readGridPopGao(subtype = "future")
```

Arguments

`subtype`	only "future" post-2000 available for this source

Author(s)

David Chen, Felicitas Beier

`readGridPopIsimip` *readGridPopIsimip*

Description

Reads in past and future (SSP1-5) gridded population data, from ISIMIP database, Past data is based on HYDE3.2, while future SSPs are based on projections from Jones & O'Neill 2016

Usage

```
readGridPopIsimip(subtype)
```

Arguments

`subtype`	past (1965-2005) or future (2010-2100)

Value

A MAgPIE object, cellular 0.5deg resolution, of population (millions)

Author(s)

David Chen, Marcos Alves, Felicitas Beier

readLabourProdImpactEmu

readLabourProdImpactEmu

Description

read in labour productivity impacts from climate change emulated by the LAMACLIMA project based on method of Orlov et al. 2019. Economics of Disasters and Climate Change, 3(3), 191-211.

Usage

```
readLabourProdImpactEmu()
```

Value

magpie object of gridded productivity loss in percent (0-100)

Author(s)

Michael Windisch, Florian Humpenöder, Felicitas Beier

See Also

readSource

readLabourProdImpactOrlov

readLabourProdImpactOrlov

Description

read in labour productivity impacts from climate change from Orlov (see Orlov et al. 2019. Economic Losses of Heat-Induced Reductions in Outdoor Worker Productivity: a Case Study of Europe. Economics of Disasters and Climate Change, 3(3), 191-211.)

Usage

```
readLabourProdImpactOrlov(
  subtype = "IPSL-CM5A-LR_rcp85_wbgtod_hothaps_400W.nc"
)
```

Arguments

subtype	subtype of choice between indoor outdoor work, GCM, work intesnsity (300W medium, 400W high, rcp)

Value

magpie object of gridded productivity as share of 1 (full productivity)

Author(s)

David Chen

See Also

readSource

`readLeifeld2018` *readLeifeld2018*

Description

read potential peatland area from Leifeld2018

Usage

```
readLeifeld2018()
```

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
readSource("Leifeld2018", convert = "onlycorrect")

## End(Not run)
```

readMAPSPAM *readMAPSPAM*

Description

Reads the MAP-SPAM crop data per year (mapping each year different)

Usage

```
readMAPSPAM(subtype = "harvested")
```

Arguments

subtype	It can be either "harvested" or "physical" area

Value

magpie object with croparea data in ha

Author(s)

Edna J. Molina Bacca, Felicitas Beier

See Also

[readSource()]

Examples

```
## Not run:
a <- readSource("MAPSPAM")

## End(Not run)
```

readMehta2024 *readMehta2024*

Description

reads in Global Area Equipped for Irrigation for years 1900-2015 from Mehta et al. (2022)

Usage

```
readMehta2024(subtype = "GMIA")
```

Arguments

`subtype` data subtype to be downloaded. Subtypes available: 'GMIA': gridded base map for downscaling from Stefan et al. (2013). Global Map of Irrigation Areas version 5. 'Meier2018': gridded base map for downscaling from Meier, et al. (2018). Global Irrigated Areas.

Author(s)

Felicitas Beier

See Also

[correctMehta2024()]

Examples

```
## Not run:
a <- readSource("Mehta2024")

## End(Not run)
```

---

readRamankutty *readRamankutty*

Description

Read in data of Ramankutty dataset (Source: Ramankutty N, Foley JA, Norman J and McSweeney K (2002) The global distribution of cultivable lands: current patterns and sensitivity to possible climate change. Global Ecology and Biogeography, 11, 377-392.). Link to data: https://www.nelson.wisc.edu/sage/data-and-models/global-land-use/grid.php

Usage

```
readRamankutty()
```

Value

magpie object

Author(s)

Felicitas Beier

Examples

```
## Not run:
readSource("Ramankutty", convert = "onlycorrect")

## End(Not run)
```

readSoilClassification *readSoilClassification*

Description

Read soil classification data used as input for lpjml

Usage

```
readSoilClassification(subtype = "HWSD.soil")
```

Arguments

subtype	Switch between different inputs

Value

Magpie object with results on cellular level for soil types

Author(s)

Marcos Alves, Kristine Karstens

Examples

```
## Not run:
readSource("SoilClassification", subtype="HWSD.soil", convert="onlycorrect")

## End(Not run)
```

readTransportDistance *readTransportDistance*

Description

Read transport distance

Usage

```
readTransportDistance()
```

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

David Chen

Examples

```
## Not run:
  readSource("TransportDistance", convert="onlycorrect")

## End(Not run)
```

`readTravelTimeNelson2019` *readTravelTimeNelson2019*

Description

Read minimum travel time to cities and ports and ports of various size, see metadata file in source folder

Usage

```
readTravelTimeNelson2019(subtype = "cities50")
```

Arguments

`subtype` currently only cities of 5, 20, or 50 thousand people ("cities5", "cities20", "cities50") or ports of various sizes ("portsLarge|Medium|Small|VerySmall|Any")

Value

gridded magpie object for 2015, minimum travel time to cities in minutes

Author(s)

David M Chen

readWindisch2021 *readWindisch2021*

Description

Reads in data to calculate BphEffect, BphTCRE or BphMask depending on the chosen subtype. BphEffect: Biogeophysical temperature change of afforestation (degree C). (File is based on observation datasets of Bright et al. 2017 and Duveiller et al. 2018). BphMask: Mask of Datapoints of biogeophysical temperature change of afforestation (degree C) to be used as weight. (File is based on observation datasets of Bright et al. 2017 and Duveiller et al. 2018). BphTCRE: Transient Climate Response to accumulated doubling of CO2. (File is based on CMIP5 +1perc CO2 per year experiment. To be used in the translation to carbon equivalents of BphEffect)

Usage

```
readWindisch2021(subtype)
```

Arguments

`subtype`	refordefor_BPHonly_05_new, annmean_pertCha_05_EW1, annstd_diff_pertCha_05_EW1

Value

List of magpie objects with results on cellular level, weight, unit and description.

Author(s)

Felicitas Beier, Michael Windisch, Patrick v. Jeetze

Examples

```
## Not run:
  readSource("Windisch2021", convert="onlycorrect")

## End(Not run)
```

toolApplyRegionNames *Apply region names*

Description

This tool function replaces country names with region names in the spatial dimension of the object. To avoid mixing up of cache files with different regional aggregation the regioncode needs to supplied and checked as well. Only if the supplied regions code agrees with the region mapping currently chosen the function will return the data.

Usage

```
toolApplyRegionNames(cdata, regionscode)
```

Arguments

`cdata`	a cluster data file as produced by cluster_base
`regionscode`	regionscode of the regional mapping to be used. Must agree with the regionscode of the mapping mentioned in the madrat config! Can be retrieved via `regionscode()`.

Value

the cluster data file with region names in spatial dimension rather than country names

Author(s)

Jan Philipp Dietrich, Felicitas Beier

See Also

`calcClusterKMeans`, `calcClusterBase`

`toolClusterPerRegion` *toolClusterPerRegion*

Description

This function calculates an appropriate number of clusters per region as it is needed for ClusterKMeans

Usage

```
toolClusterPerRegion(cells, ncluster, weight = NULL)
```

Arguments

`cells`	spatial names as returned by `getCells`
`ncluster`	The desired total number of clusters.
`weight`	named vector with weighting factors for each region for the cluster distribution, e.g. weight=c(AFR=3,EUR=0.5). weight > 1 will grant more cluster to a region and weight < 1 less cluster than by default.

Value

A matrix with regions in rows and number of cells and clusters in columns

Author(s)

Jan Philipp Dietrich

See Also

`calcClusterKMeans`, `calcClusterBase`

---

toolClusterPerRegionManual *toolClusterPerRegionManual*

Description

This function translates weights into number of clusters per region as it is needed for ClusterKMeans. Weights have to sum up to total number of clusters.

Usage

```
toolClusterPerRegionManual(cells, ncluster, ncluster2reg)
```

Arguments

`cells`	spatial names as returned by `getCells`
`ncluster`	The desired total number of clusters.
`ncluster2reg`	named vector with numbers per region

Value

A matrix with regions in rows and number of cells and clusters in columns

Author(s)

Kristine Karstens

See Also

`calcClusterKMeans`, `calcClusterBase`

toolMoveValues *toolMoveValues*

Description

Distances are calculated from the lat and lon coordinates. Therefore, all magpie objects must have location information (see addLocation). Values are only moved within a country. If no suitable cell is available in the same country, the undesirable values are discarded. This function takes only magpie objects with only one time and data dimensions to allow for more flexibility. Whenever more than one dimension is available in the magpie objects, I suggest using a loop (see for and apply).

Usage

```
toolMoveValues(x, y, z, w = NULL)
```

Arguments

`x`	Unidimensional magpie object (one time and one data dimension) with location information caring for the values that must be checked and moved if necessary.
`y`	Unidimensional magpie object (one time and one data dimension) that has a binary or logical mapping (see as.logical) of the unsuitable areas for the values in x
`z`	Unidimensional magpie object (one time and one data dimension) that has a binary or logical (see as.logical) mapping of the areas that can receive the values from x.
`w`	Unidimensional magpie object (one time and one data dimension) that has a binary or logical (see as.logical) mapping of the areas that have to be zeroed. If left empty, the inverse of ‘z‘ is assumed.

Details

Move values in an undesirable cell to the nearest desirable neighbor (Euclidian distance).

Value

Unidimensional magpie object with summed values of the moved values to the nearest suitale neighbor. All the unmoved and discarded values are set to 0.

Author(s)

Marcos Alves

toolNeuralNet *Neural Network Reconstruction*

Description

Reconstructs and evaluate a neural network from the weights and biases provided as arguments

Usage

```
toolNeuralNet(inputsMl, weights, activation)
```

Arguments

`inputsMl`	Neural Network input features properly scaled with the scale and center attributes of the scaled training set in a matrix format.
`weights`	The learned weights and biases in a list format as outputed by the function `keras::get_weights()`.
`activation`	Name of the activation function used for training. Currently implemented functions: ‘relu‘, ‘softplus‘, ‘sigmoid‘. Optionally, a custom activation function can be passed using a "." to indicate where the layer inputs should be piped.

Value

The evaluated result of the neural network for the `input_ml` parameter.

Author(s)

Marcos Alves

toolRefoldWeights *Refold weights from NN training Refold weights into their original configuration.*

Description

Refold weights from NN training Refold weights into their original configuration.

Usage

```
toolRefoldWeights(x)
```

Arguments

`x`	magpie object containing weights.

Author(s)

Marcos Alves

Index