Package: magpie4 (via r-universe)

October 8, 2026

Type Package

Title MAgPIE outputs R package for MAgPIE version 4.x

Version 2.83.0

Date 2026-09-30

Description Common output routines for extracting results from the MAgPIE framework (versions 4.x).

License LGPL-3 | file LICENSE

URL https://github.com/pik-piam/magpie4, https://doi.org/10.5281/zenodo.1158582

BugReports https://github.com/pik-piam/magpie4/issues

Depends magclass (>= 2.40), R (>= 3.5.0)

Imports dplyr, gdx2 (>= 0.1.6), gms, luplot, luscale, lusweave, madrat (>= 3.10.0), magpiesets, methods, memoise, mip, mstools (>= 0.12.1), nonparaeff, piamutils, reshape2, rlang, R.utils, rworldmap (>= 1.3.8), stats, utils, tidyr, GDPuc, MASS

Suggests covr, filelock, FRACTION, ncdf4, terra, testthat (>= 3.3.2), withr

Encoding UTF-8

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first fullReport

Config/Needs/website tidyverse/tidytemplate

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake libgdal-dev gdal-bin libgeos-dev libglpk-dev make libicu-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-09-30 12:01:07 UTC

RemoteUrl https://github.com/pik-piam/magpie4

RemoteRef HEAD

RemoteSha dc9ff69fcfdddf8c9b5f30191c04f66998c9a1d2

Contents

magpie4-package . . . 9
addGeometry . . . 9
addToDataChangelog . . . 10
agEmployment . . . 11
AgGDP . . . 12
AgriResearchIntensity . . . 13
anthropometrics . . . 13
BII . . . 15
biochar . . . 16
bioplasticDemand . . . 17
bodyweight . . . 18
carbonHWP . . . 19
carbonLTS . . . 20
carbonstock . . . 22
cellularFit . . . 23
changelogVariables . . . 24
changelogYears . . . 24
clearCacheMagpie4 . . . 25
clusterOutputToTerraVector . . . 25
consumptionValue . . . 26
CostCapital . . . 27
CostOverall . . . 28
costs . . . 29
CostsAEI . . . 30
CostsFertilizer . . . 30
costsMACCS . . . 31
costsPresolve . . . 32
costsWholesale . . . 33
CostsWithoutIncentives . . . 33
CostTC . . . 34
CostTransport . . . 35
croparea . . . 36
CropareaDiversityIndex . . . 37
cropland . . . 38
croplandSCM . . . 39
croplandTreeCover . . . 39
cshare . . . 40
deco . . . 41
demand . . . 42
demandBioenergy . . . 43
disaggregateLandConservation . . . 44
discountRates . . . 45
domesticFootprintTrade . . . 45
embodiedBiodiversity . . . 47
embodiedEmissions . . . 48
embodiedLabor . . . 50

embodiedLand . . . 51
embodiedResourceKastner . . . 52
embodiedWater . . . 53
emisCO2 . . . 55
Emissions . . . 56
EmissionsBeforeTechnicalMitigation . . . 57
emisSOC . . . 58
expectVariablesPresent . . . 59
expenditureIndexFood . . . 60
factorCosts . . . 61
factorCostShares . . . 62
fallow . . . 63
feed . . . 63
FoodDemandModuleConsumerPrices . . . 64
FoodExpenditure . . . 65
FoodExpenditureShare . . . 66
foodmodelstat . . . 67
footprintDemand . . . 68
footprints . . . 69
ForestYield . . . 71
gdxAggregate . . . 72
getReport . . . 73
getReportAgMIP . . . 74
getReportDemandStandalone . . . 76
getReportDietaryIndicators . . . 77
getReportFableScenathon . . . 78
getReportFSECAlessandroPassaro . . . 78
getReportFSECCosts . . . 79
getReportFSECCropDiversityGrid . . . 80
getReportFSECPollution . . . 81
getReportFSECSimonDietz . . . 82
getReportFSECStevenLord . . . 83
getReportGridINMS . . . 83
getReportGridNitrogenPollution . . . 84
getReportINMS . . . 85
getReportIso . . . 86
getReportMAgPIE2GAINS . . . 88
getReportMAgPIE2LPJmL . . . 89
getReportMAgPIE2REMIND . . . 90
getReportPBindicators . . . 91
GrowingStock . . . 93
harvested_area_timber . . . 94
hourlyLaborCosts . . . 95
Hunger . . . 95
income . . . 96
Intake . . . 97
IntakeDetailed . . . 98
IntakeDetailedProtein . . . 99

kayaFractions . . . 100
kayaLaspeyres . . . 101
Kcal . . . 102
laborCostsEndo . . . 104
laborProductivity . . . 105
land . . . 105
land_price . . . 107
landCarbonSink . . . 108
landConservation . . . 109
landForestry . . . 110
landTransitionMatrix . . . 111
lastIter . . . 112
legacyEmissions . . . 113
LivestockDemStructure . . . 115
LivestockShare . . . 116
m_yeardiff . . . 117
malmquist . . . 118
ManureExcretion . . . 118
mappingToLongFormat . . . 119
metadata_comments . . . 120
modelstat . . . 121
NetForestChange . . . 122
NitrogenBudget . . . 122
NitrogenBudgetNonagland . . . 123
NitrogenBudgetPasture . . . 124
NitrogenBudgetWithdrawals . . . 125
OtherLand . . . 125
out . . . 126
outputCheck . . . 127
outputPerWorker . . . 127
PeatlandArea . . . 128
PeatlandEmissions . . . 129
PlantationEstablishment . . . 130
population . . . 131
PriceElasticities . . . 132
PriceGHG . . . 133
priceIndex . . . 134
priceIndexFood . . . 135
prices . . . 136
primaryPerSecondary . . . 138
PrimSecdOtherLand . . . 139
processing . . . 140
productEmissions . . . 141
production . . . 142
productionProfit . . . 143
productionRevenue . . . 144
readGDXBilateral . . . 145
reassignLivestockPathway . . . 145

relativeHourlyLaborCosts . . . 146
reportAAI . . . 147
reportAEI . . . 148
reportAgEmployment . . . 149
reportAgGDP . . . 150
reportAgriResearchIntensity . . . 151
reportAnthropometrics . . . 152
reportBII . . . 153
reportBiochar . . . 154
reportBiogasFeedstock . . . 154
reportBioplasticDemand . . . 155
reportCarbonstock . . . 156
reportConsumVal . . . 157
reportCostCapitalInvestment . . . 158
reportCostCapitalStocks . . . 159
reportCostOverall . . . 160
reportCosts . . . 161
reportCostsAccounting . . . 162
reportCostsAEI . . . 163
reportCostsFertilizer . . . 164
reportCostsInputFactors . . . 165
reportCostsMACCS . . . 166
reportCostsPresolve . . . 167
reportCostsTrade . . . 168
reportCostsWholesale . . . 168
reportCostsWithoutIncentives . . . 169
reportCostTC . . . 170
reportCostTransport . . . 171
reportCroparea . . . 172
reportCropareaGrid . . . 173
reportCropDiversity . . . 174
reportCroplandSCM . . . 175
reportCropResidues2ndBE . . . 176
reportDemand . . . 177
reportDemandBioenergy . . . 178
reportDemandNr . . . 179
reportEmissions . . . 180
reportEmissionsBeforeTechnicalMitigation . . . 184
reportExpenditureFoodIndex . . . 185
reportExtraResidueEmissions . . . 187
reportFactorCostShares . . . 188
reportFeed . . . 189
reportFeedConversion . . . 190
reportFertilizerNitrogen . . . 191
reportFireEmissions . . . 192
reportFit . . . 193
reportFoodExpenditure . . . 194
reportFootprints . . . 195

reportForestYield . . . 198
reportGridCroparea . . . 199
reportGridLand . . . 200
reportGridManureExcretion . . . 201
reportGrowingStock . . . 202
reportharvested_area_timber . . . 203
reportHourlyLaborCosts . . . 204
reportHunger . . . 205
reportIncome . . . 206
reportIntakeDetailed . . . 207
reportKcal . . . 208
reportLaborCostsEmpl . . . 209
reportLaborProductivity . . . 210
reportLandConservation . . . 211
reportLandTransitionMatrix . . . 213
reportLandUse . . . 216
reportLandUseChange . . . 218
reportLandUseForSEALS . . . 219
reportLivestockDemStructure . . . 220
reportLivestockShare . . . 221
reportManure . . . 222
reportManureFuel . . . 223
reportNetForestChange . . . 224
reportNitrogenBudgetCropland . . . 225
reportNitrogenBudgetNonagland . . . 227
reportNitrogenBudgetPasture . . . 228
reportNitrogenEfficiencies . . . 229
reportNitrogenPollution . . . 230
reportOutputPerWorker . . . 231
reportPastSoilCarbon . . . 232
reportPBbiosphere . . . 233
reportPBland . . . 234
reportPBnitrogen . . . 235
reportPBwater . . . 236
reportPeatland . . . 237
reportPlantationEstablishment . . . 238
reportPopulation . . . 239
reportPriceAgriculture . . . 240
reportPriceBioenergy . . . 241
reportPriceElasticities . . . 242
reportPriceFoodIndex . . . 243
reportPriceGHG . . . 244
reportPriceLand . . . 245
reportPriceShock . . . 246
reportPriceWater . . . 247
reportPriceWoodyBiomass . . . 248
reportProcessing . . . 249
reportProcessingResiduesForestry . . . 250

reportProcessingWoodResidues . . . 253
reportProducerPriceIndex . . . 253
reportProduction . . . 254
reportProductionBioenergy . . . 255
reportProductionGrowth . . . 257
reportProductionNr . . . 258
reportProtein . . . 259
reportRelativeHourlyLaborCosts . . . 260
reportResidues . . . 261
reportRotationLength . . . 262
reportRuralDemandShares . . . 263
reportSDG1 . . . 264
reportSDG12 . . . 265
reportSDG15 . . . 266
reportSDG2 . . . 267
reportSDG3 . . . 268
reportSDG6 . . . 269
reportSDG9 . . . 270
reportSOM . . . 270
reportSOM2 . . . 273
reportTau . . . 275
reportTc . . . 276
reportTimber . . . 277
reportTimberDemand . . . 278
reportTotalHoursWorked . . . 279
reportTrade . . . 280
reportTradeGrowth . . . 282
reportValueMaterialDemand . . . 283
reportValueTrade . . . 284
reportVegfruitShare . . . 285
reportWageDevelopment . . . 286
reportWaterAvailability . . . 287
reportWaterIndicators . . . 288
reportWaterUsage . . . 289
reportWoodFuel . . . 290
reportWorkingAgePopulation . . . 291
reportYields . . . 292
reportYieldsCropCalib . . . 293
reportYieldsCropRaw . . . 294
reportYieldShifter . . . 295
ResidueBiomass . . . 297
Residues . . . 298
ResidueUsage . . . 299
RotationLength . . . 300
ruralDemandShares . . . 301
Seed . . . 302
SOM . . . 303
SOM2 . . . 304

submitCalibration . . . 304
superAggregateX . . . 305
surplusChange . . . 306
tau . . . 307
taxRevenueRotations . . . 308
tc . . . 309
Timber . . . 310
TimberDemand . . . 311
TimberProductionVolumetric . . . 312
timePeriods . . . 313
totalHoursWorked . . . 314
trade . . . 314
tradedPrimaries . . . 316
tradeKastner . . . 318
tradeValue . . . 319
tryList . . . 320
tryReport . . . 321
validation . . . 321
ValueMaterialDemand . . . 323
VegfruitShare . . . 323
wageDevelopment . . . 324
wageRent . . . 325
water_AAI . . . 326
water_AEI . . . 327
water_avail . . . 327
water_efficiency . . . 328
water_price . . . 329
water_usage . . . 330
waterEFR . . . 331
waterEFVarea . . . 332
waterEFViolation . . . 333
waterEFVratio . . . 334
waterStress . . . 334
waterStressedPopulation . . . 335
waterStressRatio . . . 336
woodHarvestArea . . . 337
woodProduction . . . 337
yields . . . 338
YieldsCropCalib . . . 339
YieldsCropRaw . . . 340

Index **341**

magpie4-package *MAgPIE outputs R library for MAgPIE version 4.x*

Description

Common output routines for extracting results from the MAgPIE framework (versions 4.x).

Author(s)

Maintainer: Jan Philipp Dietrich <dietrich@pik-potsdam.de>

See Also

Useful links:

- https://github.com/pik-piam/magpie4
- doi:10.5281/zenodo.1158582
- Report bugs at https://github.com/pik-piam/magpie4/issues

addGeometry *addGeometry*

Description

Enriches land use data on cluster resolution geometry information as required for conversion by magclass::as.SpatVector

Usage

```
addGeometry(x, clustermap)
```

Arguments

x	Landuse data on cluster/cell resolution as a magclass object
clustermap	A dataframe mapping with columns cluster, cell, and optionally country

Value

A magclass object enriched with geometry information

Author(s)

Jan Philipp Dietrich, Pascal Sauer

See Also

Other Spatial: `clusterOutputToTerraVector()`, `gdxAggregate()`, `mappingToLongFormat()`, `superAggregateX()`

Examples

```
## Not run:
landUse <- magpie4::land("fulldata.gdx", level = "cell")
clustermap <- readRDS(Sys.glob("clustermap_*.rds"))
landUseEnriched <- magpie4::addGeometry(landUse, clustermap)
attr(landUseEnriched, "geometry")
attr(landUseEnriched, "crs")

## End(Not run)
```

`addToDataChangelog` *addToDataChangelog*

Description

Prepend data from the given report to the changelog.

Usage

```
addToDataChangelog(
  report,
  changelog,
  versionId,
  years = changelogYears(),
  variables = changelogVariables(),
  ...,
  maxEntries = 15,
  roundDigits = 2
)
```

Arguments

`report`	data.frame as obtained by readRDS("report.rds")
`changelog`	Path to the changelog file
`versionId`	The model version identifier, e.g. a release number like 4.9.1 or a date like 2025-02-01
`years`	For which years the variables should be read and put into the changelog
`variables`	named vector of which variables to read from the report
`...`	Reserved for future expansion.

maxEntries	The maximum number of versionIds to keep in the changelog, the oldest one is removed first.
roundDigits	Numbers are rounded to this many decimal places before being written to the changelog.

Value

Invisibly, the written changelog as data.frame

Author(s)

Pascal Sauer

See Also

Other Data Changelog: changelogVariables(), changelogYears()

---

agEmployment *agEmployment*

Description

returns employment in crop+livestock production from MAgPIE results

Usage

```
agEmployment(gdx, type = "absolute", detail = TRUE, level = "reg", file = NULL)
```

Arguments

gdx	GDX file
type	"absolute" for total number of people employed, "share" for share out of working age population
detail	if TRUE, employment is disaggregated to crop products, livestock products and (if available) mitigation measures, if FALSE only aggregated employment is reported, if byProduct then to all crop products
level	spatial aggregation to report employment ("iso", "reg", "glo" or "regglo", if type is "absolute" also "grid")
file	a file name the output should be written to using write.magpie

Value

employment in agriculture as absolute value or as percentage of working age population

Author(s)

Debbora Leip, David M Chen

Examples

```
## Not run:
x <- agEmployment(gdx)

## End(Not run)
```

AgGDP *AgGDP*

Description

Reads data to calculate the agricultural GDP

Usage

```
AgGDP(gdx, file = NULL, level = "reg")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing values related with overall value of production [million US$17]

Author(s)

Edna Molina Bacca

Examples

```
## Not run:
  x <- AgGDP(gdx)

## End(Not run)
```

AgriResearchIntensity *AgriResearchIntensity*

Description

calculates Agricultural Research Intensity (Investment in AgR&D/Total GDP) from a MAgPIE gdx file

Usage

```
AgriResearchIntensity(gdx, file = NULL, level = "reg")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	aggregation level, reg, glo or regglo, cell or grid

Author(s)

David M Chen

Examples

```
## Not run:
x <- AgriResearchIntensity(gdx)

## End(Not run)
```

anthropometrics *anthropometrics*

Description

Calculates anthropometic indicators from the food demand model

Usage

```
anthropometrics(
  gdx,
  indicator = "bodyheight",
  age = "adults",
  sex = FALSE,
  bmi_groups = FALSE,
  level = "iso",
```

```
  final = TRUE,
  file = NULL,
  calibrated = TRUE
)
```

Arguments

gdx	GDX file
indicator	bodyheight, bodyweight, bodyweight_healthy, BMI(Body Mass Index) or PAL (physical activity level)
age	if TRUE, demand is scaled down to age-groups and sex using food requirements
sex	if FALSE, female and male are aggregated, if sex, results are divided into males and females
bmi_groups	if TRUE, data is provided by BMI group
level	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
final	final results or preliminary results (the latter are the ones magpie uses for optimization before last iteration with demand model)
file	a file name the output should be written to using write.magpie
calibrated	if TRUE, uses the calibrated intake estimates for bodyweight estimation

Details

Demand definitions are equivalent to FAO Food supply categories

Value

bodyweight (kg), bodyheight (cm), BMI or PAL as magpie objects

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- anthropometrics(gdx)

## End(Not run)
```

BII *Biodiversity intactness index*

Description

calculates the area weighted biodiversity intactness index (BII) out of a MAgPIE gdx file

Usage

```
BII(
  gdx,
  file = NULL,
  level = "glo",
  mode = "auto",
  landClass = "sum",
  spatialWeight = NULL,
  adjusted = FALSE,
  bii_coeff = NULL,
  side_layers = NULL
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	level of regional aggregation; "cell" (magpie cluster level), "reg" (regional), "glo" (global), "regglo" (regional and global), "iso" (country level), "grid" (0.5 degree grid cell level).
mode	"auto" (default), "from_grid", "MAgPIE" or "postprocessing". • "MAgPIE" reports the BV based on values from the MAgPIE biodiversity module. • "postprocessing" calculates the BV based on land information from MAgPIE (for versions where biodiversity module was not available yet). • "auto" uses "MAgPIE" if available and falls back to "postprocessing" otherwise. • "from_grid" calculates BII values from BII output and returns aggregated values at the aggregation level specified.
landClass	"all" returns average BII values for all land classes of ov_bv, "sum" returns the weighted BII over all land classes of ov44_bv_weighted.
spatialWeight	Spatial weight for aggregating BII values. Only relevant if mode is "from_grid", adjusted is TRUE, or level is either "grid" or "iso".
adjusted	if "TRUE", function returns adjusted BII values (results have been adjusted for primary and secondary other land).
bii_coeff	file containing BII coefficients. Only needed for mode = "postprocessing". NULL tries to automatically detected the file.
side_layers	file containing LUH2 side layers. NULL tries to automatically detected the file.

Details

Calculates global, regional and cluster-level biodiversity intactness index (BII)

Value

Biodiversity intactness index (unitless)

Author(s)

Patrick v. Jeetze, Florian Humpenoeder, Felicitas Beier

Examples

```
## Not run:
x <- BII(gdx)

## End(Not run)
```

`biochar` *biochar*

Description

Calculates biochar-related indicators from a MAgPIE gdx file. This function provides all indicators for biochar-related material flows and conversion, including biochar production, feedstock demand, and long-term stable carbon storage in soils.

Usage

```
biochar(
  gdx,
  indicator,
  level = "reg",
  feedstockAggr = FALSE,
  systemAggr = FALSE,
  attributes = "c",
  file = NULL
)
```

Arguments

`gdx`	GDX file
`indicator`	Indicator types: bc_production, bc_feedstock_dem or bc_stable_carbon
`level`	Spatial aggregation: "reg", "glo", or "regglo"
`feedstockAggr`	If TRUE, aggregates over feedstock types. If not applicable for the selected indicator, set to FALSE (default).

systemAggr	If TRUE, aggregates over bc_sys63 or biopyr_all63. If not applicable for the selected indicator, set to FALSE (default).
attributes	Available output attributes: dry matter: Mt ("dm"), gross energy: PJ ("ge"), carbon: Mt C ("c"). Can also be a vector. The availability of attributes depends on the selected indicator.
file	File name the output should be written to using write.magpie

Value

Selected biochar-related indicator as MAgPIE object:

- bc_production: Biochar production (unit depends on attributes). Available attributes are "dm", "ge", and "c".
- bc_feedstock_dem: Biomass feedstock demand for biochar (unit depends on attributes). Available attributes are "dm", "ge", and "c".
- bc_stable_carbon: Stable C in soil from biochar after 100 years (mio. tC per yr). Available attribute is "c".

Author(s)

Isabelle Weindl

Examples

```
## Not run:
  x <- biochar(gdx, indicator = "bc_production", level = "regglo", feedstockAggr = TRUE)

## End(Not run)
```

---

bioplasticDemand *bioplasticDemand*

Description

returns demand for bioplastic or demand for substrate for bioplastic production

Usage

```
bioplasticDemand(
  gdx,
  type = "bioplastic",
  detail = FALSE,
  level = "regglo",
  file = NULL
)
```

Arguments

gdx	GDX file
type	"bioplastic" for bioplastic demand, "substrate" for biomass demand as substrate for bioplastic production
detail	only relevant for type = "substrate". If TRUE, substrate demand is disaggregated by crop type, if FALSE only the aggregated demand is reported.
level	spatial aggregation to report bioplastic/substrate demand (only "reg" or "regglo")
file	a file name the output should be written to using write.magpie

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- bioplasticDemand(gdx)

## End(Not run)
```

---

bodyweight *bodyweight*

Description

Calculates the prevalence of underweight, normalweight, overweight (excluding obesity) and obesity. For more detailed body mass classifications see functions population or anthropometrics.

Usage

```
bodyweight(
  gdx,
  level = "reg",
  age = FALSE,
  sex = FALSE,
  share = FALSE,
  population = NULL
)
```

Arguments

gdx	GDX file
level	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
age	if TRUE, demand is scaled down to age-groups and sex using food requirements

sex	if FALSE, female and male are aggregated, if sex, results are divided into males and females
share	if TRUE, data is provided by BMI group
population	population information from GDX. Can be provided to speed up calculation process. Will be read from GDX, if not provided.

Details

Demand definitions are equivalent to FAO Food supply categories

Value

MAgPIE object with mio people or share of people in each weight category

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- bodyweight(gdx)

## End(Not run)
```

---

carbonHWP *carbonHWP*

Description

reads carbon stored in harvested timber out of a MAgPIE gdx file

Usage

```
carbonHWP(
  gdx,
  file = NULL,
  level = "cell",
  unit = "element",
  half_life = 35,
  cumulative = FALSE,
  baseyear = 1995
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate
unit	element" or "gas"; "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr; "gas": co2_c Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr
half_life	Half life in years for decay in wood products loosing half theor carbon content. (35 yrs is deafault)
cumulative	Logical; Determines if cHWP emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995.
baseyear	Baseyear used for cumulative emissions (default = 1995)

Details

Annual (and cumulative) Carbon stored in harvested wood products as well as slow emissions from half life deacy.

Value

carbon stocks in MtC from harvested timber

Author(s)

Abhijeet Mishra, Florian Humpenoeder

Examples

```
  ## Not run:
    x <- carbonHWP(gdx)

## End(Not run)
```

---

carbonLTS *carbonLTS*

Description

reads carbon stored in harvested timber out of a MAgPIE gdx file

Usage

```
carbonLTS(
  gdx,
  file = NULL,
  level = "cell",
  unit = "element",
  cumulative = FALSE,
  baseyear = 1995
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregateX
unit	element" or "gas"; "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr; "gas": co2_c Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr
cumulative	Logical; Determines if cHWP emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995.
baseyear	Baseyear used for cumulative emissions (default = 1995)

Details

Annual (and cumulative) Carbon stored in harvested wood products, as well as, slow emissions from half life deacy.

Value

carbon stocks in MtC from harvested timber

Author(s)

Abhijeet Mishra, Florian Humpenoeder

Examples

```
## Not run:
x <- carbonLTS(gdx)

## End(Not run)
```

carbonstock *carbonstock*

Description

reads carbon stocks out of a MAgPIE gdx file

Usage

```
carbonstock(
  gdx,
  file = NULL,
  level = "cell",
  sum_cpool = TRUE,
  sum_land = TRUE,
  subcategories = NULL,
  stockType = "actual"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate
sum_cpool	sum over carbon pool dimension (default = TRUE)
sum_land	sum over land type dimension (default = TRUE)
subcategories	NULL or vector of strings. If NULL, no subcategories are returned. Meaningful options are "crop, "forestry" and "other"
stockType	carbon stock type (default = "actual"). Options: "actual", "previousLandPattern" and "previousCarbonDensity".

Details

carbon pools consist of vegetation carbon (vegc), litter carbon (litc) and soil carbon (soilc)

Value

carbon stocks in MtC

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- carbonstock(gdx)

## End(Not run)
```

`cellularFit` *cellular fit*

Description

cellular fit/error/bias calculations at regional and global level

Usage

```
cellularFit(
  gdx,
  file = NULL,
  level = "cell",
  statistic = "MAE",
  variable = "land",
  dataset = "LUH3",
  water_aggr = FALSE
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	level at which the regional and global bias should be calculated. Options "cell" or "grid"
`statistic`	R2, MAE, MPE (mean percentage error - bias), MAPE (mean absolute percentage error)
`variable`	variable to be evaulated: land (land types) or crop (crop types)
`dataset`	dataset to compare with. LUH3 only option for variable land. LUH3 and MAPSPAM for the crop variable.
`water_aggr`	if irrigation types for crops should be agregated or not

Value

returns selected statistic at regglo level for the historical part of the time horizon

Author(s)

Edna J. Molina Bacca, Patrick v. Jeetze

Examples

```
## Not run:
  x <- cellularFit(gdx)

## End(Not run)
```

`changelogVariables` *changelogVariables*

Description

Returns a named vector of variables to put into the data changelog, see `addToDataChangelog`.

Usage

```
changelogVariables()
```

Author(s)

Pascal Sauer

See Also

Other Data Changelog: `addToDataChangelog()`, `changelogYears()`

`changelogYears` *changelogYears*

Description

Returns the years to put into the data changelog, see `addToDataChangelog`.

Usage

```
changelogYears()
```

Author(s)

Pascal Sauer

See Also

Other Data Changelog: `addToDataChangelog()`, `changelogVariables()`

clearCacheMagpie4 *clearCacheMagpie4*

Description

Clear memoise cache for all memoised functions in magpie4

Usage

```
clearCacheMagpie4()
```

Value

no return, but cache cleared

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
x <- clearCacheMagpie4()

## End(Not run)
```

clusterOutputToTerraVector
Convert cluster output to terra vector

Description

Enriches land use data on cluster resolution with explicit spatial information by creating a terra polygon for each cluster according to the given clustermap.

Usage

```
clusterOutputToTerraVector(x, clustermap)
```

Arguments

x	Landuse data on cluster/cell resolution as a magclass object
clustermap	A dataframe mapping with columns cluster, cell, and country

Value

A SpatVector with the following columns: c("clusterId", "country", "region", "year", "landtype", "value")

Author(s)

Pascal Führlich, Patrick v. Jeetze

See Also

Other Spatial: addGeometry(), gdxAggregate(), mappingToLongFormat(), superAggregateX()

Examples

```
## Not run:
landUse <- magpie4::land("fulldata.gdx", level = "cell")
clustermap <- readRDS(Sys.glob("clustermap_*.rds"))
clusterPolygons <- magpie4::clusterOutputToTerraVector(landUse, clustermap)
terra::writeVector(clusterPolygons, "cluster_resolution.shp")

## End(Not run)
```

---

consumptionValue *consumptionValue*

Description

calculates consumption value of different types based on a MAgPIE gdx file.

Usage

```
consumptionValue(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = TRUE,
  type = NULL,
  type_aggr = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not (boolean, default TRUE)
type	Consumption type(s): "food", "feed", "processed", "other_util", "bioenergy", "seed", "waste", "dom_balanceflow; NULL returns all types
type_aggr	aggregate over demand types or not (boolean, default TRUE)

Value

A MAgPIE object containing consumption value in million $US.

Author(s)

Miodrag Stevanovic

Examples

```
  ## Not run:
    x <- consumptionValue(gdx)

## End(Not run)
```

`CostCapital` *CostCapital*

Description

Reads data to calculate capital stocks

Usage

```
CostCapital(gdx, type = "stocks", file = NULL, level = "cell")
```

Arguments

gdx	GDX file
type	either capital stocks ("stocks") or overall capital investment "investment"
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing values related with overall value of production [million US$17]

Author(s)

Edna Molina Bacca

Examples

```
  ## Not run:
    x <- CostCapital(gdx)

## End(Not run)
```

`CostOverall` *CostOverall*

Description

Gross value of productions

Usage

```
CostOverall(gdx, file = NULL, level = "reg")
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing values related with overall value of production [million US$17]

Author(s)

Edna Molina Bacca

Examples

```
## Not run:
x <- CostOverall(gdx)

## End(Not run)
```

costs *costs*

Description

reads costs entering the objective function from a MAgPIE gdx file

Usage

```
costs(gdx, file = NULL, level = "reg", type = "annuity", sum = TRUE)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
type	either "annuity" (as it enters the objetive function) or "investment" (investment)
sum	total costs (TRUE) or detailed costs (FALSE)

Value

A MAgPIE object containing the goal function costs including investments [million US$17]

Author(s)

Jan Philipp Dietrich, Markus Bonsch, Misko Stevanovic, Florian Humpenoeder, Edna J. Molina Bacca, Michael Crawford

Examples

```
## Not run:
x <- costs(gdx)

## End(Not run)
```

CostsAEI *CostsAEI*

Description

reads AEI costs entering the objective function from a MAgPIE gdx file

Usage

```
CostsAEI(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation ("reg", "glo", "regglo")

Value

MAgPIE object containing costs for AEI [million US$17]

Author(s)

Felicitas Beier

Examples

```
  ## Not run:
    x <- CostsAEI(gdx)

## End(Not run)
```

CostsFertilizer *CostsFertilizer*

Description

reads costs entering the objective function from a MAgPIE gdx file

Usage

```
CostsFertilizer(gdx, file = NULL, level = "regglo", disagg = TRUE)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation ("reg", "glo", "regglo")
disagg	whether costs should be disaggregated into the different crop types

Value

MAgPIE object containing fertilizer costs [million US$17]

Author(s)

Debbora Leip

Examples

```
  ## Not run:
    x <- CostsFertilizer(gdx)

## End(Not run)
```

---

costsMACCS *costsMACCS*

Description

reads costs entering the objective function from a MAgPIE gdx file

Usage

```
costsMACCS(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation ("reg", "glo", "regglo")

Value

MAgPIE object containing mitigation costs [million US$17]

Author(s)

Debbora Leip

Examples

```
  ## Not run:
    x <- costsMACCS(gdx)

## End(Not run)
```

`costsPresolve` *costsPresolve*

Description

reads presovle costs (i.e. without bioenergy demand) entering the objective function from a MAgPIE gdx file

Usage

```
costsPresolve(gdx, file = NULL, level = "reg")
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Details

Presolve is without bioenergy demand. Hence costs from a MAgPIE run with bioenergy demand minus costs from presolve reflect costs that can be attributed to bioenergy production

Value

A MAgPIE object containing the goal function costs in presolve mode [million US$17]

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- costsPresolve(gdx)

## End(Not run)
```

costsWholesale *costsWholesale*

Description

Reads data to calculate wholesale costs

Usage

```
costsWholesale(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregateX

Value

A MAgPIE object containing values related with costs wholesale trade [million US$17/tDM]

Author(s)

David M Chen

Examples

```
## Not run:
x <- costsWholesale(gdx)

## End(Not run)
```

CostsWithoutIncentives *CostsWithoutIncentives*

Description

calculates agricultural costs without taxes, incentives and technical penalty costs (i.e. GHG taxes and BII incentives)

Usage

```
CostsWithoutIncentives(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	aggregation level, reg, glo or regglo

Value

A MAgPIE object containing the costs without taxes, incentives and technical penalty costs [million US$17]

Author(s)

David M Chen

Examples

```
## Not run:
  x <- CostsWithoutIncentives(gdx)

## End(Not run)
```

CostTC *CostTC*

Description

Reads data on TC costs

Usage

```
CostTC(gdx, file = NULL, level = "reg")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing values related with overall value of production [million US$17]

Author(s)

David Chen

Examples

```
  ## Not run:
    x <- CostTC(gdx)

## End(Not run)
```

`CostTransport` *CostTransport*

Description

reads costs entering the objective function from a MAgPIE gdx file

Usage

```
CostTransport(gdx, file = NULL, level = "cell", sum = FALSE)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate.
`sum`	total costs (TRUE) or detailed costs (FALSE)

Value

A MAgPIE object containing the transport costs [million US$17]

Author(s)

David Chen

Examples

```
  ## Not run:
    x <- CostTransport(gdx)

## End(Not run)
```

croparea *croparea*

Description

reads croparea out of a MAgPIE gdx file. Croparea excludes fallow land.

Usage

```
croparea(
  gdx,
  file = NULL,
  level = "reg",
  products = "kcr",
  product_aggr = TRUE,
  water_aggr = TRUE,
  physical = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not (boolean)
water_aggr	aggregate irrigated and non-irriagted production or not (boolean).
physical	if TRUE (default) physical (cultivated) area is returned; if FALSE harvested area, i.e. physical area scaled by the multicropping ratio.

Value

production as MAgPIE object (unit depends on attributes)

Author(s)

Jan Philipp Dietrich, Florian Humpenoeder

See Also

reportCroparea

Examples

```
## Not run:
x <- croparea(gdx)

## End(Not run)
```

`CropareaDiversityIndex` *CropareaDiversityIndex*

Description

calculates an index that measures the croparea diversity

Usage

```
CropareaDiversityIndex(
  gdx,
  index = "shannon",
  level = "reg",
  measurelevel = "cell",
  groupdiv = "agg1"
)
```

Arguments

`gdx`	GDX file
`index`	can be "shannon", "gini" or "invsimpson" for different types of diversitiy indices
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
`measurelevel`	level at which diversity is measured. "cell" means diversity
`groupdiv`	should crop groups be split up into several individual items or not? Choose either FALSE or different (dis)aggregation methods "agg1", "agg2"

Value

MAgPIE object (unit depends on attributes)

Author(s)

Benjamin Leon Bodirsky

See Also

`CropareaDiversityIndex`

Examples

```
## Not run:
x <- CropareaDiversityIndex(gdx)

## End(Not run)
```

`cropland` *cropland*

Description

reads cropland out of a MAgPIE gdx file. Cropland includes croparea plus fallow plus cropland with tree cover

Usage

```
cropland(gdx, level = "reg", types = NULL, sum = FALSE)
```

Arguments

`gdx`	GDX file
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
`types`	NULL or a vector of strings. If NULL, all land types are used. Options are "crop_area", "crop_fallow", "crop_treecover"
`sum`	determines whether output should be land-type-specific (FALSE) or aggregated over all types (TRUE).

Value

land as MAgPIE object (Mha)

Author(s)

Benjamin Leon Bodirsky

See Also

`reportLandUse`

Examples

```
## Not run:
x <- cropland(gdx)

## End(Not run)
```

croplandSCM *croplandSCM*

Description

reads cropland area under soil carbon management (SCM) out of a MAgPIE gdx file (Mha)

Usage

```
croplandSCM(gdx, level = "reg", crop_aggr = TRUE)
```

Arguments

gdx	GDX file
level	aggregation level, "reg", "glo", "regglo", or "cell"
crop_aggr	aggregate over crop types (TRUE) or report by crop (FALSE)

Value

SCM area as MAgPIE object (mio. ha)

Author(s)

Kristine Karstens

Examples

```
## Not run:
x <- croplandSCM(gdx)

## End(Not run)
```

croplandTreeCover *croplandTreeCover*

Description

calculates tree cover on cropland (Mha) from a MAgPIE gdx file

Usage

```
croplandTreeCover(gdx, level = "reg", sum_ac = TRUE, debugMode = FALSE)
```

Arguments

gdx	GDX file
level	aggregation level, reg, glo or regglo, cell or grid
sum_ac	sum over age classes TRUE / FALSE
debugMode	debug mode TRUE makes some consistency checks between estimates for different resolutions

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- fallow(gdx)

## End(Not run)
```

---

cshare *cshare*

Description

Calculates soil carbon share in relation to potential natural vegetation based on a MAgPIE gdx file

Usage

```
cshare(
  gdx,
  file = NULL,
  level = "reg",
  reference = "actual",
  noncrop_aggr = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
reference	default is "actual" (cshare in actual carbon stocks). Other option is "target" (cshare in target carbon stocks).
noncrop_aggr	aggregate non cropland types to 'noncropland' (if FALSE all land types of pools59 will be reported)

Value

A MAgPIE object containing som values

Author(s)

Kristine Karstens

Examples

```
## Not run:
x <- cshare(gdx)

## End(Not run)
```

---

deco *deco*

Description

Function that quantifies the influences of the underlying drivers to a dependent output variable. It attributes the changes of the output variable (A) to changes of several drivers (B, B/C, C/A). The output must be the product of the drivers.

Usage

```
deco(data, names_factor = NULL, plot = FALSE)
```

Arguments

data	Decomposition Data as a magpie object. The first column of the third dimension has to be the output (A), while the subsequent columns are the coefficients of the drivers (B,C,...). Example: Area = Population x Supply/Population x Area/Supply. 3rd-dimension column order then has to be: Area, Population, Supply.
names_factor	Names of the output (A) and the Decomposition-Factors (B,B/C,C/A), if names_factor=NULL the names for the third column will be generated like the factors for decomposition (above example: Area, Population, Supply/Population, Area/Supply)
plot	TRUE or FALSE

Details

Use function deco_plot in library luplot to make a plot out of this. It is only usable for the decomposition of 5 or less drivers. For documentation, see paper Huber, Veronika, Ina Neher, Benjamin L. Bodirsky, Kathrin Hoefner, and Hans Joachim Schellnhuber. 2014. "Will the World Run out of Land? A Kaya-Type Decomposition to Study Past Trends of Cropland Expansion." Environmental Research Letters 9 (2): 024011. https://doi.org/10.1088/1748-9326/9/2/024011. Or see master Thesis of Ina Neher (2013)

Value

Decomposes the impact of certain drivers to an output (A) value.

Author(s)

Ina Neher, Benjamin Leon Bodirsky

Examples

```
Data<-array(c(1,1.1,1.15,1,1.05,1.1,1,1.05,1.15),c(3,3))
dimnames(Data)<-list(paste("y",2000:2002,sep=""),c("Area","Population","Supply"))
Data <- as.magpie(Data)
deco(Data)
```

---

demand *demand*

Description

Calculates MAgPIE demand out of a gdx file

Usage

```
demand(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = FALSE,
  attributes = "dm",
  type = NULL,
  type_aggr = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation ("reg", "glo", "regglo")
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not (boolean, default FALSE)
attributes	dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector.

type	Demand type(s): "food", "feed", "processed", "other_util", "bioenergy", "seed", "waste", "dom_balanceflow; NULL returns all types
type_aggr	aggregate over demand types or not (boolean, default FALSE)

Details

Demand definitions are equivalent to FAO CBS categories

Value

demand as MAgPIE object (Unit depends on attributes)

Author(s)

Benjamin Leon Bodirsky, Abhijeet Mishra, Miodrag Stevanovic

Examples

```
  ## Not run:
    x <- demand(level="regglo", products="kcr")

## End(Not run)
```

`demandBioenergy` *demandBioenergy*

Description

reads bioenergy demand from a MAgPIE gdx file

Usage

```
demandBioenergy(gdx, file = NULL, level = "reg", sum = FALSE, round = NULL)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
sum	1st and 2nd generation bioenergy demand (FALSE) or total bioenergy demand (TRUE)
round	NULL or number of digits

Value

A MAgPIE object containing bioenergy demand in EJ/yr

Author(s)

Jan Philipp Dietrich, Florian Humpenoeder

Examples

```
  ## Not run:
    x <- demandBioenergy(gdx)

## End(Not run)
```

`disaggregateLandConservation` *disaggregateLandConservation*

Description

Read land conservation data from a fulldata.gdx and disaggregate to high resolution (0.5 deg).

Usage

```
disaggregateLandConservation(gdx, cfg, mapping, wdpaHr, conservationPrioHr)
```

Arguments

`gdx`	character, path to a fulldata.gdx
`cfg`	list, config data usually obtained with gms::loadConfig("config.yml")
`mapping`	dataframe, must include columns cell and cluster
`wdpaHr`	magclass, World Database on Protected Areas data in high resolution
`conservationPrioHr`	magclass, high resolution data on conservation priority areas

Value

magclass, high resolution land conservation area

Author(s)

Patrick v. Jeetze, Pascal Sauer

discountRates *discountRates*

Description

reads discount rates from a MAgPIE gdx file

Usage

```
discountRates(gdx, file = NULL, level = "reg")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing discount rates used in the model

Author(s)

Xiaoxi Wang

Examples

```
  ## Not run:
    x <- discountRates(gdx)

## End(Not run)
```

domesticFootprintTrade *domesticFootprintTrade*

Description

Calculates embodied land in trade using the domestic technology assumption.

Usage

```
domesticFootprintTrade(
  gdx,
  file = NULL,
  level = "reg",
  type = "net-trade",
  productsAggr = TRUE,
  maxIntensityRatio = 5
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg", "glo", "regglo"
type	"exports", "imports", or "net-trade" (default)
productsAggr	Aggregation of primary products; default is TRUE
maxIntensityRatio	Maximum allowed ratio of regional to historical (1995) land intensity. Regional land intensities exceeding this factor times the 1995 value are capped. This same value is used for filling NAs. Use NULL for no capping. Default is 5 (caps at 5× 1995 values).

Details

The domestic technology assumption values all traded goods using each region's own processing shares, feed baskets, and land intensity (yields). For exports this represents the actual land footprint. For imports this is a counterfactual: how much land would the region need if it produced its imports domestically?

Trade flows are decomposed into primary product equivalents via three pathways: (1) direct primary trade, (2) primaries embodied in secondary products, (3) primaries needed as livestock feed. Land intensity (ha/tDM) is then applied per primary product.

Regional land intensities are filled and capped using historical (1995) FAO yields as reference. This ensures comparability across different model runs while accounting for regional differences in agricultural potential.

This is a consistent approach with net trade data (no bilateral flows). With bilateral trade data, different basis for land intensity could be applied.

Value

MAgPIE object with embodied land in trade (Mha)

Author(s)

Kristine Karstens

See Also

tradedPrimaries

Examples

```
## Not run:
  x <- domesticFootprintTrade(gdx, type = "net-trade")

## End(Not run)
```

`embodiedBiodiversity` *embodiedBiodiversity*

Description

Calculates production-based and consumption-based (embodied) biodiversity impact accounting using bilateral trade flows. Biodiversity values (BII-weighted area) are allocated to traded products based on cropland and pasture area shares, similar to how CO2 LUC emissions are allocated.

Usage

```
embodiedBiodiversity(
  gdx,
  file = NULL,
  level = "reg",
  type = "all",
  indicator = "bv",
  bilateral = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate. Only used when bilateral=FALSE.
type	Type of accounting: "production" (production-based), "consumption" (consumption-based), "trade" (export, import, and net-trade), "all" (all five), or "flows" (bilateral flows, requires bilateral=TRUE)
indicator	Which biodiversity indicator to use: "bv" (biodiversity value, BII-weighted area in Mha), "bii_loss" (1-BII, representing biodiversity loss)
bilateral	Logical; if TRUE, returns bilateral flows with dimensions (exporter.importer, year, product) instead of regional totals (default FALSE)

Details

Biodiversity is measured via the Biodiversity Intactness Index (BII) which ranges from 0 to 1. BII is calculated at the land cover class level (crop_ann, crop_per, manpast, rangeland, etc.) and not directly per product. This function allocates the biodiversity impact from cropland to individual crop products based on their area shares.

The indicator "bv" returns the BII-weighted area (higher = more biodiversity preserved), allocated to each crop by its share of total cropland area.

The indicator "bii_loss" returns the biodiversity loss ((1-BII) * area) for aggregate cropland/pasture, then allocates to individual products by area share. This represents "share of biodiversity loss attributable to this crop based on its area share."

Value

Embodied biodiversity impact as MAgPIE object. When bilateral=FALSE: dimensions are (region, year, accounting.product). When bilateral=TRUE: dimensions are (exporter.importer, year, product).

Author(s)

David M Chen

See Also

BII, land, croparea, trade

Examples

```
## Not run:
  x <- embodiedBiodiversity(gdx, type = "all")
  # Bilateral flows
  xBilat <- embodiedBiodiversity(gdx, type = "flows", bilateral = TRUE)

## End(Not run)
```

---

embodiedEmissions *embodiedEmissions*

Description

Consumption-based (embodied) emissions footprint using the column-normalised Kastner allocation in embodiedResourceKastner. Non-negative and closes globally to total emissions. Pollutants are aggregated to a single CO2-equivalent value per product (via unit); crop emissions are allocated through primary-equivalent trade, livestock emissions through direct trade.

Usage

```
embodiedEmissions(
  gdx,
  file = NULL,
  level = "reg",
  type = "all",
  unit = "GWP100AR6",
  bilateral = FALSE,
  secdToFeed = TRUE,
  reassignLivestock = TRUE
)
```

Arguments

gdx	GDX file
file	optional file name to write the result with `write.magpie`
level	regional aggregation level (only "reg" supported)
type	"production", "consumption", "trade", or "all" (default)
unit	GWP metric passed to `productEmissions` (default "GWP100AR6")
bilateral	logical; if TRUE return bilateral (exporter.importer) flows
secdToFeed	logical; if TRUE (default) move the processed-then-fed share (e.g. soybean -> oilcake -> feed) from the secd pathway to the feed pathway, so the Livestock pathway captures all crop products that end up as feed. See `embodiedResourceKastner`.
reassignLivestock	logical; if TRUE (default) move every livestock product's whole footprint into the feed (Livestock) pathway. See `embodiedResourceKastner`.

Value

MAgPIE object (region, year, accounting.pathway.product) in Mt CO2eq.

Author(s)

David M Chen

See Also

`embodiedResourceKastner`, `embodiedEmissions`

`embodiedLabor` *embodiedLabor*

Description

Consumption-based (embodied) agricultural-labour footprint using the column-normalised Kastner allocation in `embodiedResourceKastner`. Non-negative and closes globally to total agricultural employment. Crop/pasture labour is allocated through primary-equivalent trade, livestock labour through direct trade.

Usage

```
embodiedLabor(
  gdx,
  file = NULL,
  level = "reg",
  type = "all",
  bilateral = FALSE,
  secdToFeed = TRUE,
  reassignLivestock = TRUE
)
```

Arguments

`gdx`	GDX file
`file`	optional file name to write the result with `write.magpie`
`level`	regional aggregation level (only "reg" supported)
`type`	"production", "consumption", "trade", or "all" (default)
`bilateral`	logical; if TRUE return bilateral (exporter.importer) flows
`secdToFeed`	logical; if TRUE (default) move the processed-then-fed share (e.g. soybean -> oilcake -> feed) from the secd pathway to the feed pathway, so the Livestock pathway captures all crop products that end up as feed. See `embodiedResourceKastner`.
`reassignLivestock`	logical; if TRUE (default) move every livestock product's whole footprint into the feed (Livestock) pathway. See `embodiedResourceKastner`.

Value

MAgPIE object (region, year, accounting.pathway.product) in million people.

Author(s)

David M Chen

See Also

`embodiedResourceKastner`, `embodiedLabor`

embodiedLand *embodiedLand*

Description

Consumption-based (embodied) land footprint using the column-normalised Kastner allocation in `embodiedResourceKastner`. Unlike `embodiedLand` (which uses production + net-trade and can produce local negatives), this distributes each region's actual cropland and pasture to consumers, so the consumption footprint is non-negative and closes globally to total agricultural land.

Usage

```
embodiedLand(
  gdx,
  file = NULL,
  level = "reg",
  type = "all",
  landType = "all",
  bilateral = FALSE,
  secdToFeed = TRUE,
  reassignLivestock = TRUE
)
```

Arguments

`gdx`	GDX file
`file`	optional file name to write the result with `write.magpie`
`level`	regional aggregation level (only "reg" supported)
`type`	"production", "consumption", "trade", or "all" (default)
`landType`	"all" (crop + pasture), "crop", or "past"
`bilateral`	logical; if TRUE return bilateral (exporter.importer) flows
`secdToFeed`	logical; if TRUE (default) move the processed-then-fed share (e.g. soybean -> oilcake -> feed) from the secd pathway to the feed pathway, so the Livestock pathway captures all crop products that end up as feed. See `embodiedResourceKastner`.
`reassignLivestock`	logical; if TRUE (default) move every livestock product's whole footprint into the feed (Livestock) pathway. See `embodiedResourceKastner`. A no-op for land (no kli land).

Value

MAgPIE object in Mha. When bilateral=FALSE: (region, year, accounting.pathway.product). When bilateral=TRUE: (exporter.importer, year, pathway.product).

Author(s)

David M Chen

See Also

embodiedResourceKastner, embodiedLand

`embodiedResourceKastner` *embodiedResourceKastner*

Description

Generic consumption-based (embodied) resource footprint using a column-normalised Kastner (2011) allocation. Each producing region's resource total (e.g. cropland, water, emissions, labour) is distributed to consuming regions using the bilateral commodity consumption shares from the Kastner $(I - A)^{-1}$ solution, normalised so that each origin's column sums exactly to its actual resource. This guarantees, by construction:

- non-negativity of the consumption footprint (if the resource is non-negative);
- production[r] == resource[r];
- additivity: production - exports + imports == consumption;
- global closure: sum(consumption) == sum(resource).

Crop/pasture products are allocated through primary-equivalent (feed-traced) trade; livestock products (`kli`) through direct bilateral trade. The prim/secd/feed pathway split is derived from demand-category shares (processed -> secd, feed -> feed, remainder -> prim), as in embodiedLand/embodiedWater.

Usage

```
embodiedResourceKastner(
  gdx,
  resource,
  file = NULL,
  level = "reg",
  type = "all",
  bilateral = FALSE,
  secdToFeed = TRUE,
  reassignLivestock = TRUE
)
```

Arguments

gdx	GDX file
resource	MAgPIE object (region, year, product) of the resource total per product (production basis). Products may include crop/pasture and livestock (kli) products; the function routes them to the correct trade matrix automatically.
file	optional file name to write the result with `write.magpie`
level	regional aggregation level (only "reg" supported for now)
type	"production", "consumption", "trade", or "all" (default); ignored when `bilateral = TRUE`
bilateral	logical; if TRUE, returns the bilateral consumption allocation with dimensions (exporter.importer, year, pathway.product) where exporter = production origin and importer = consumer. Default FALSE.
secdToFeed	logical; if TRUE, the share of each primary product that is processed and then fed to livestock (e.g. soybean -> oilcake -> feed) is moved from the secd pathway to the feed pathway. The processed-then-fed share is sSecd * phi, where phi[primary] is the feed fraction of the secondary products derived from that primary (mass allocation via `primaryPerSecondary`). Default TRUE (processed soybean etc. is moved to feed, keyed on its immediate demand category).
reassignLivestock	logical; if TRUE (default) every livestock (`kli`) product's whole footprint (all pathways) is moved into the `feed` (Livestock) pathway via `reassignLivestockPathway`, so the feed pathway carries the FULL livestock footprint (feed crops PLUS the livestock products' own enteric/manure/labour/water footprint) rather than only the feed crops. Orthogonal to `secdToFeed` (which re-routes a crop's processed-then-fed share); a no-op for resources without `kli` products (e.g. land). Per-product totals are conserved either way.

Value

MAgPIE object. When `bilateral = FALSE`: (region, year, accounting.pathway.product) with accounting in {production, consumption, export, import, net-trade} and pathway in {prim, secd, feed}. When `bilateral = TRUE`: (exporter.importer, year, pathway.product).

Author(s)

David M Chen

`embodiedWater` *embodiedWater*

Description

Consumption-based (embodied) water footprint using the column-normalised Kastner allocation in `embodiedResourceKastner`. Non-negative and closes globally to total water use. Crop water is allocated through primary-equivalent (feed-traced) trade; livestock water through direct trade.

Usage

```
embodiedWater(
  gdx,
  file = NULL,
  level = "reg",
  type = "all",
  waterType = "consumption",
  bilateral = FALSE,
  secdToFeed = TRUE,
  reassignLivestock = TRUE
)
```

Arguments

gdx	GDX file
file	optional file name to write the result with `write.magpie`
level	regional aggregation level (only "reg" supported)
type	"production", "consumption", "trade", or "all" (default)
waterType	"consumption" (default) or "withdrawal"
bilateral	logical; if TRUE return bilateral (exporter.importer) flows
secdToFeed	logical; if TRUE (default) move the processed-then-fed share (e.g. soybean -> oilcake -> feed) from the secd pathway to the feed pathway, so the Livestock pathway captures all crop products that end up as feed. See `embodiedResourceKastner`.
reassignLivestock	logical; if TRUE (default) move every livestock product's whole footprint into the feed (Livestock) pathway. See `embodiedResourceKastner`.

Value

MAgPIE object (region, year, accounting.pathway.product), or bilateral.

Author(s)

David M Chen

See Also

`embodiedResourceKastner`, `embodiedWater`

emisCO2 *emisCO2*

Description

reads detailed CO2 emissions out of a MAgPIE gdx file

Usage

```
emisCO2(
  gdx,
  file = NULL,
  level = "cell",
  unit = "gas",
  sum_cpool = TRUE,
  sum_land = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregateX
unit	"element" or "gas"; "element": co2_c in Mt C/yr "gas": co2_c Mt CO2/yr
sum_cpool	aggregate carbon pools (TRUE), below ground (soilc) and above ground (vegc and litc) will be reported, if FALSE
sum_land	TRUE (default) or FALSE. Sum over land types (TRUE) or report land-type specific emissions (FALSE).

Value

CO2 emissions as MAgPIE object (unit depends on unit)

Author(s)

Florian Humpenoeder, Michael Crawford

Examples

```
## Not run:
x <- emisCO2(gdx)

## End(Not run)
```

`Emissions` *Emissions*

Description

reads GHG emissions out of a MAgPIE gdx file

Usage

```
Emissions(
  gdx,
  file = NULL,
  level = "reg",
  type = "co2_c",
  unit = "element",
  subcategories = TRUE,
  cumulative = FALSE,
  lowpass = NULL,
  inorg_fert_split = TRUE
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
`type`	emission type(s): "co2_c", "n2o_n" or "ch4"
`unit`	"element", "gas", "GWP100AR5", "GWP100AR6", "GWP*AR5", or "GWP*AR6" "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr "gas": co2_c in Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr "GWP": co2_c in Mt CO2/yr, n2o_n in Mt CO2eq/yr, ch4 in Mt CO2eq/yr
`subcategories`	FALSE (default) or TRUE
`cumulative`	Logical; Determines if emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995.
`lowpass`	number of lowpass filter iterations
`inorg_fert_split`	if TRUE then inorganic fertilizer emissions are further disaggregated into pasture- and cropland-related emissions. Both the aggregated ("inorg_fert") and disaggregated values ("inorg_fert_crop", "inorg_fert_past)" are reported

Value

emissions as MAgPIE object (unit depends on `unit`)

Author(s)

Florian Humpenoeder, Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- Emissions(gdx)

## End(Not run)
```

EmissionsBeforeTechnicalMitigation *EmissionsBeforeTechnicalMitigation*

Description

reads GHG emissions before technical abatement out of a MAgPIE gdx file. Technical abatement includes all abatement done in the MACC curves, but exclude endogenous mitigation. These emissions are NOT the standard reporting emissions, but used for special purposes like remind-magpie coupling.

Usage

```
EmissionsBeforeTechnicalMitigation(
  gdx,
  file = NULL,
  level = "reg",
  type = "co2_c",
  unit = "element",
  subcategories = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
type	emission type(s): "co2_c", "n2o_n" or "ch4" and in the case of unit="gas" "co2" and "n2o"
unit	"element", "gas" or "co2eq"; "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr; "gas": co2_c Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr; "co2eq": co2_c in Mt CO2/yr, n2o_n in Mt CO2eq/yr, ch4 in Mt CO2eq/yr
subcategories	FALSE (default) or TRUE

Value

emissions as MAgPIE object (unit depends on `unit`)

Author(s)

Florian Humpenoeder; Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- EmissionsBeforeTechnicalMitigation(gdx)

## End(Not run)
```

`emisSOC` *emisSOC*

Description

Reads detailed CO2 soil emissions out of a MAgPIE gdx file using Shapley-style decomposition with interaction terms

Usage

```
emisSOC(gdx, file = NULL, sumLand = FALSE)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`sumLand`	TRUE (default) or FALSE. Sum over land types (TRUE) or report land-type specific emissions (FALSE).

Details

This function uses a structural decomposition approach (Shapley decomposition) to attribute soil carbon emissions to different drivers and their interactions. The framework is based on counterfactual analysis where all possible combinations of driver states are evaluated following the inclusion-exclusion principle.

For four drivers, the total emission is decomposed as:

totEmissions = main_effects + first_order_interactions + second_order_interactions + third_order_interactions + residual

This creates 16 counterfactual scenarios (2^4) to isolate individual and combined effects. Main effects capture changes when one driver varies while others stay constant. Interaction terms capture synergies where driver combinations amplify or dampen effects beyond simple addition.

Drivers included: - Climate (C): Changes in reference soil carbon density - Land-use (LU): Transitions between land types - Management (M): Changes in crop type shares, fallow, and treecover * Can be split into treecover vs other management in attribution - Soil Carbon Management (SCM): Enhanced soil carbon practices

Legacy effects are incorporated using exponential decay (85 over 100 years. Interaction terms are proportionally attributed back to main drivers in the final output.

Value

CO2 emissions as MAgPIE object

Author(s)

Kristine Karstens, Florian Humpenoeder

Examples

```
## Not run:
x <- emisSOC(gdx)

## End(Not run)
```

`expectVariablesPresent` *expectVariablesPresent*

Description

Checks whether a set of expected variable names is present in a MAgPIE report object. Variable names are normalized in that summation symbols are removed.

Usage

```
expectVariablesPresent(report, variableNames)
```

Arguments

`report`	A MAgPIE object containing the report to check.
`variableNames`	A character vector of variable names that are expected to be present in `report`.

Value

`NULL` invisibly. Issues a warning if expected variables are not present.

Author(s)

Patrick Rein

See Also

Other Infrastructure: `metadata_comments()`, `out()`, `tryList()`, `tryReport()`

`expenditureIndexFood` *expenditureIndexFood*

Description

calculates food expenditure index (baseyear = 100) corrected for ghg emission costs based on a MAgPIE gdx file

Usage

```
expenditureIndexFood(
  gdx,
  file = NULL,
  level = "reg",
  products = "kfo",
  basketyear = "y2010",
  baseyear = "y2010",
  round = TRUE,
  ghgtax = TRUE,
  valueAdded = FALSE
)
```

Arguments

`gdx`	GDX file
`file`	File the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in mapping
`products`	Selection of products (either by naming products, e.g. "tece", or naming a set, e.g."kcr")
`basketyear`	Year of reference food basket (should be in the past for comparison of different runs to have identical and comparable food basket)
`baseyear`	Baseyear of the price index
`round`	Rounded result (TRUE or FALSE)
`ghgtax`	Correction of food price expenditure for ghg emission costs (TRUE or FALSE)
`valueAdded`	whether to include value added

Value

A MAgPIE object containing food price expenditure index, in 2017Int$PPP

Author(s)

Felicitas Beier, David Chen

Examples

```
## Not run:
x <- expenditureIndexFood(gdx)

## End(Not run)
```

`factorCosts` *factorCosts*

Description

reads factor costs for crops, livestock, residues or pasture entering the objective function from a MAgPIE gdx file. Depending on the product and the MAgPIE version (and factor cost realization), factor costs are either already split into labor and capital, will be split in this function, or are kept as the aggregate

Usage

```
factorCosts(gdx, products = "kli", file = NULL, level = "regglo")
```

Arguments

`gdx`	GDX file
`products`	products for which factor costs should be reported ("kcr", "kli", "kres", "fish", or "pasture")
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation ("reg", "glo", "regglo")

Value

MAgPIE object containing factor costs [million US$17]

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- factorCosts(gdx)

## End(Not run)
```

`factorCostShares` *factorCostShares*

Description

returns labor and capital cost share out of factor costs (i.e. labor + capital)

Usage

```
factorCostShares(
  gdx,
  type = "optimization",
  products = "kcr",
  level = "reg",
  file = NULL
)
```

Arguments

gdx	GDX file
type	• "requirements": shares from factor requirements • "optimization": cost shares between labor and capital costs in optimization • "accounting": cost shares based on accounting of labor and capital costs
products	products for which cost shares should be reported, kcr or kli
level	spatial aggregation to report employment ("reg", "glo" or "regglo")
file	a file name the output should be written to using write.magpie

Value

labor and capital cost share out of factor costs

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- factorCostShares(gdx)

## End(Not run)
```

fallow *fallow*

Description

calculates fallow land (Mha) from a MAgPIE gdx file

Usage

```
fallow(gdx, level = "reg", debugMode = FALSE)
```

Arguments

gdx	GDX file
level	aggregation level, reg, glo or regglo, cell or grid
debugMode	debug mode TRUE makes some consistency checks between estimates for different resolutions

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
x <- fallow(gdx)

## End(Not run)
```

feed *feed*

Description

calculates feed demand by animal type out of a gdx file

Usage

```
feed(
  gdx,
  file = NULL,
  level = "reg",
  detail = TRUE,
  nutrient = "dm",
  balanceflow = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
detail	if FALSE, only total feed demand per animal type is calculated without details on the type of feed
nutrient	The nutrient in which the results shall be calculated
balanceflow	If true, feed includes the calibration balanceflow

Value

feed demand by animal type as MAgPIE object (unit depends on selected nutrient attributes)

Author(s)

Isabelle Weindl

Examples

```
  ## Not run:
    x <- feed(gdx)

## End(Not run)
```

FoodDemandModuleConsumerPrices

FoodDemandModuleConsumerPrices

Description

Calculates food prices that enter demand module

Usage

```
FoodDemandModuleConsumerPrices(gdx, level = "iso", valueAdded = FALSE)
```

Arguments

gdx	GDX file
level	reg or iso
valueAdded	whether to add the value-added marketing margin to the total expenditures

Value

magpie object

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- FoodDemandModuleConsumerPrices(gdx)

## End(Not run)
```

`FoodExpenditure` *FoodExpenditure*

Description

Calculates the food expenditure in USD per year

Usage

```
FoodExpenditure(
  gdx,
  level = "reg",
  after_shock = TRUE,
  products = "kfo",
  product_aggr = TRUE,
  per_capita = TRUE,
  valueAdded = FALSE
)
```

Arguments

`gdx`	GDX file
`level`	spatial aggregation. can be "iso","reg","regglo","glo"
`after_shock`	FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income, "after_price_before_demand" takes into account price changes on real income, but assumes no demand adjustment
`products`	selected products or sets of products
`product_aggr`	if true, aggregation over products
`per_capita`	per capita or total population
`valueAdded`	whether to add the value-added marketing margin to the total expenditures

Value

magpie object with per capita consumption

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- FoodExpenditure(gdx)

## End(Not run)
```

`FoodExpenditureShare` *FoodExpenditureShare*

Description

Calculates the livestock share from the food demand model

Usage

```
FoodExpenditureShare(
  gdx,
  level = "reg",
  after_shock = TRUE,
  products = "kfo",
  product_aggr = TRUE,
  valueAdded = FALSE
)
```

Arguments

`gdx`	GDX file
`level`	spatial aggregation. can be "iso","reg","regglo","glo"
`after_shock`	FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income
`products`	selected products or sets of products
`product_aggr`	if true, aggregation over products
`valueAdded`	TRUE to include post farmgate value added in the expenditure share

Value

magpie object with per capita consumption

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- FoodExpenditureShare(gdx)

## End(Not run)
```

`foodmodelstat` *foodmodelstat*

Description

MAgPIE food model statistics with information about convergence and number of iterations

Usage

```
foodmodelstat(gdx, file = NULL)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie

Value

A MAgPIE object containing number of iterations and convergence information for each time step

Author(s)

Jan Philipp Dietrich

Examples

```
## Not run:
x <- foodmodelstat(gdx)

## End(Not run)
```

`footprintDemand` *footprintDemand*

Description

Per-tonne denominator for consumption-based (embodied) resource footprints: the tonnes of the FINAL product consumed in each prim/secd/feed pathway. Used by `footprints(type = "perTonne")` and `reportFootprints` to turn an absolute footprint into a per-tonne intensity that is comparable across resources.

- `prim` = primary products eaten directly (all demand categories except the processed and feed categories);
- `secd` = tonnes of SECONDARY (ksd) products consumed (the processing OUTPUT, not the primary input); with `secdToFeed = TRUE` the fed-secondary share is excluded so it reads per tonne of secondary eaten as food/non-feed;
- `feed` = tonnes of LIVESTOCK (kli) products consumed.

Usage

```
footprintDemand(dem, prods, kli, ksd, secdToFeed = TRUE)
```

Arguments

`dem`	demand object (region, year, demand.product) as returned by `demand(gdx, level)`, with the `dom_balanceflow` category already removed.
`prods`	products present in the footprint (used to restrict the primary set); typically `getItems(<footprint>, dim = "product")`.
`kli`	livestock product set (`readGDX(gdx, "kli")`).
`ksd`	secondary (processed) product set (`readGDX(gdx, "ksd")`).
`secdToFeed`	logical; if TRUE (default) drop the fed share of secondary products from the secd denominator so it reads per tonne of secondary eaten as food. Should match the `secdToFeed` used for the numerator. See `embodiedResourceKastner`.

Value

MAgPIE object (region, year, pathway) with pathway in {prim, secd, feed}, in tonnes (Mt DM) of the final product consumed.

Author(s)

David M Chen

See Also

`footprints`, `reportFootprints`

Examples

```
## Not run:
  dem   <- demand(gdx, level = "reg")[, , "dom_balanceflow", invert = TRUE]
  denom <- footprintDemand(dem, getItems(dem, dim = 3.2),
                           kli = readGDX(gdx, "kli"), ksd = readGDX(gdx, "ksd"))

## End(Not run)
```

`footprints` *footprints*

Description

Consumption-based (embodied) resource footprints as ready-to-use MAgPIE objects, on top of the column-normalised Kastner allocation in `embodiedResourceKastner`. A single entry point that (a) chooses the resource (`land`, `water`, `emissions` or `labor`), (b) derives regional production / consumption / import / export / net-trade by prim/secd/feed pathway and product from the bilateral Kastner allocation, and (c) returns the footprint as an absolute total, per capita, or per tonne of the final product consumed in each pathway.

This is the reusable computation extracted from the bilateral footprint plotting scripts, so any MAgPIE run can obtain the same numbers without the plotting layer. Regional accounting is derived from the bilateral object by aggregation: production = sum over importer; consumption = sum over exporter; domestic = the self-trade diagonal; imports = consumption - domestic; exports = production - domestic. The pathway split of every accounting component is therefore keyed on the CONSUMER's demand mix (consistent bilateral basis).

Usage

```
footprints(
  gdx,
  resource = "land",
  type = "total",
  level = "reg",
  file = NULL,
  reassignLivestock = TRUE,
  secdToFeed = TRUE,
  bil = NULL,
  dem = NULL,
  ...
)
```

Arguments

`gdx`	GDX file (must contain bilateral trade, i.e. a bilateral MAgPIE run).
`resource`	which resource footprint: "land", "water", "emissions" or "labor" (alias "labour").
`type`	the footprint metric:

- "total" (default): absolute embodied footprint (region, year, accounting.pathway.product), in the resource's own unit (Mha, km3, Mt CO2eq, mio people);
- "perCapita": total divided by population (region, year, accounting.pathway.product); unit is resource-per-person (e.g. Mha / mio people = ha / capita);
- "perTonne": footprint per tonne of the FINAL product consumed in each pathway (region, year, accounting.pathway), products summed within each pathway. The denominator is primary demand (prim), secondary (ksd) demand consumed as food (secd) and livestock (kli) demand (feed); see secdToFeed. Per-tonne is a pathway-level quantity, so the product dimension is collapsed.

level — regional aggregation level (only "reg" supported by the underlying embodied* functions).

file — optional file name to write the result with write.magpie.

reassignLivestock — logical; if TRUE (default) move every livestock product's whole footprint into the feed (Livestock) pathway, so the feed pathway carries the full livestock footprint. See reassignLivestockPathway.

secdToFeed — logical; if TRUE (default) route each crop's processed-then-fed share from secd to feed, and (for type = "perTonne") drop the fed share of secondary products from the secd denominator so it reads per tonne of secondary eaten as food. See embodiedResourceKastner.

bil — optional precomputed RAW bilateral embodied object (exporter.importer, year, pathway.product) from embodied*Kastner(bilateral = TRUE,reassignLivestock = FALSE). When supplied the expensive Kastner computation is skipped and this object is used directly (reassignment, if requested, is applied here). Enables reuse of a cached bilateral allocation.

dem — optional precomputed demand object (region, year, demand.product), as returned by demand(gdx, level) WITHOUT the dom_balanceflow category; only used for type = "perTonne". Read from gdx if not supplied.

... — further resource-specific arguments passed to the chosen embodied*Kastner wrapper (e.g. landType, waterType, unit); ignored when bil is supplied.

Value

MAgPIE object; layout depends on type (see above).

Author(s)

David M Chen

See Also

embodiedResourceKastner, reportFootprints, footprintDemand, reassignLivestockPathway

Examples

```
## Not run:
  land   <- footprints(gdx, "land", type = "total")       # Mha, by product & pathway
  emisPC <- footprints(gdx, "emissions", type = "perCapita")
  waterT <- footprints(gdx, "water", type = "perTonne")   # by pathway

## End(Not run)
```

`ForestYield` *ForestYield*

Description

Calculates the average growing stock density of harvested forest areas (m3/ha). This is the ratio of timber production volume to harvested area, i.e. the average standing volume removed per hectare harvested. Values are comparable to growing stock (typically 100-300 m3/ha) and should NOT be confused with forestry yield in the MAI sense (5-20 m3/ha/yr).

Usage

```
ForestYield(gdx, file = NULL, level = "cell")
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate

Details

Average growing stock of harvested areas = timber production (m3) / harvested area (ha). Differences to overall growing stock arise from selective harvesting of specific age classes.

Value

Average growing stock density of harvested areas in m3/ha

Author(s)

Abhijeet Mishra, Florian Humpenoeder

Examples

```
## Not run:
  x <- ForestYield(gdx)

## End(Not run)
```

gdxAggregate *gdxAggregate*

Description

aggregates and disaggregates on spatial scales using mappings from the gdx files.

Usage

```
gdxAggregate(gdx, x, weight = NULL, to, absolute = TRUE, ...)
```

Arguments

gdx	gdx file
x	object to be aggrgeagted or disaggregated
weight	weight can be either an object or a functionname in "", where the function provides the weight
to	either a fixed target aggregation level (grid, cell, iso, reg, glo, regglo) or the name of a mapping based on regions
absolute	is it a absolute or a relative value (absolute: tons, relative: tons per hectare)
...	further parameters handed on to weight function.

Value

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

Author(s)

Benjamin Leon Bodirsky, Edna J. Molina Bacca, Florian Humpenoeder

See Also

Other Spatial: addGeometry(), clusterOutputToTerraVector(), mappingToLongFormat(), superAggregateX()

Examples

```
## Not run:
gdp_pc <- income(gdx, level = "reg")
is.function(population)
gdp_pc_iso <- gdxAggregate(gdx = gdx, x = gdp_pc, weight = "population", to = "iso",
                           absolute = FALSE)
gdp_pc_glo <- gdxAggregate(gdx = gdx, x = gdp_pc, weight = "population", to = "glo",
                           absolute = FALSE)
gdp <- income(gdx, level = "reg", per_capita = FALSE)
gdp_iso <- gdxAggregate(gdx = gdx, x = gdp, weight = "population", to = "iso", absolute = TRUE)
gdp_glo <- gdxAggregate(gdx = gdx, x = gdp, weight = "population", to = "glo", absolute = TRUE)

## End(Not run)
```

getReport *getReport*

Description

Puts together a report based on a MAgPIE gdx file

Usage

```
getReport(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  detail = TRUE,
  level = "regglo",
  legacyEmis = TRUE,
  ...
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
detail	Crop specific (TRUE) or aggregated outputs (FALSE)
level	An aggregation level (currently "regglo" or "iso") or the name of a mapping that should be used by default to aggregate the report. The mapping can only map from reg to other regions. Not all parts of the report will necessarily adhere to this default aggregation level.
legacyEmis	Logical (default TRUE). Forwarded to reportEmissions and reportCarbonstock: TRUE reports the legacy-clearing reframe (the shipped default); FALSE delivers the raw instantaneous accounting and skips the pool pre-warm.
...	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Details

Reports are organize with '|' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in '|' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in '|' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

`getReportAgMIP` *getReportAgMIP*

Description

Puts together a report for the Agricultural Model Intercom- parison and Improvement Project (AgMIP) based on a MAgPIE gdx file

Usage

```
getReportAgMIP(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  detail = TRUE,
  ...
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
detail	Crop specific (TRUE) or aggregated outputs (FALSE)
...	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Details

Reports are organize with 'l' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in 'l' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in 'l' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
```

```
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Florian Humpenoeder, Isabelle Weindl, Felicitas Beier

Examples

```
## Not run:
x <- getReportAgMIP(gdx)

## End(Not run)
```

`getReportDemandStandalone` *getReportDemandStandalone*

Description

Puts together a report based on a MAgPIE gdx file

Usage

```
getReportDemandStandalone(
  gdx,
  file = NULL,
  scenario = NULL,
  detail = FALSE,
  ...
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
`scenario`	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
`detail`	Crop specific (TRUE) or aggregated outputs (FALSE)
`...`	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- getReportDemandStandalone(gdx)

## End(Not run)
```

`getReportDietaryIndicators` *getReportDietaryIndicators*

Description

reports dietary indicators on the country level. These are formatted as data.frames describing: 1. population, anthropometrics, and intake 2. caloric intake by food category (without food waste)

Usage

```
getReportDietaryIndicators(gdx, scenario)
```

Arguments

`gdx`	filepath of the GDX file
`scenario`	character string describing the scenario configuration

Value

list of data.frames for the dietary indicators

Author(s)

Michael Crawford, Felicitas Beier, Benjamin Bodirsky

Examples

```
  ## Not run:
    getReportDietaryIndicators(gdx, scenario)

## End(Not run)
```

getReportFableScenathon *getReportFableScenathon*

Description

Collects outputs from MAgPIE runs for FABLE Scenathon.

Usage

```
getReportFableScenathon(gdx, file = NULL, iso = NULL)
```

Arguments

gdx	a GDX file
file	a file name the output should be written to using write.report. If 'NULL' the report is returned instead as a MAgPIE object. For the easier reporting in Scenathon tabs, a .csv file extension is recommenended.
iso	country/region selection. Default 'NULL', i.e. all 'regglo' reporting

Author(s)

Miodrag Stevanovic

Examples

```
  ## Not run:
    x <- getReportFableScenathon(gdx, file = "magpie2scenathon.csv", iso = "IND")

## End(Not run)
```

getReportFSECAlessandroPassaro *getReportFSECAlessandroPassaro*

Description

Collects reports for Alessandro Passaro's analysis

Usage

```
getReportFSECAlessandroPassaro(gdx, reportOutputDir = NULL, scenario = NULL)
```

Arguments

gdx	gdx file
reportOutputDir	a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function.
scenario	the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function.

Value

A list of reports

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- getReportFSECAlessandroPassaro()

## End(Not run)
```

---

getReportFSECCosts *getReportFSECCosts*

Description

Reports cost indicators for the FSEC project

Usage

```
getReportFSECCosts(gdx, reportOutputDir = NULL, scenario = NULL)
```

Arguments

gdx	a GDX file
reportOutputDir	a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function.
scenario	the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function.

Value

A .csv containing the summed output of reportCostsAccounting on the region level

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- getReportFSECCosts(gdx)

## End(Not run)
```

---

getReportFSECCropDiversityGrid

getReportFSECCropDiversityGrid

Description

Reports grid cell level crop diversity for the FSEC project

Usage

```
getReportFSECCropDiversityGrid(gdx, reportOutputDir = NULL, scenario = NULL)
```

Arguments

gdx	a GDX file
reportOutputDir	a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function.
scenario	the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function.

Value

A list of MAgPIE objects containing the reports

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
x <- getReportFSECCropDiversityGrid(gdx)

## End(Not run)
```

getReportFSECPollution *getReportFSECPollution*

Description

Reports nutrient surplus indicators for the FSEC project

Usage

```
getReportFSECPollution(gdx, reportOutputDir = NULL, scenario = NULL)
```

Arguments

gdx	GDX file
reportOutputDir	a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function.
scenario	the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function.

Value

A list of MAgPIE objects containing the reports

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- getReportFSECPollution(gdx)

## End(Not run)
```

getReportFSECSimonDietz *getReportFSECSimonDietz*

Description

Collects reports for Simon Dietz' social welfare function analysis

Usage

```
getReportFSECSimonDietz(gdx, reportOutputDir = NULL, scenario = NULL)
```

Arguments

gdx	GDX file
reportOutputDir	a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function.
scenario	the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function.

Value

A list of reports

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- getReportFSECSimonDietz()

## End(Not run)
```

getReportFSECStevenLord *getReportFSECStevenLord*

Description

Collects reports for Steven Lord's cost of action / cost of inaction analysis.

Usage

```
getReportFSECStevenLord(gdx, reportOutputDir, scenario)
```

Arguments

gdx	gdx file
reportOutputDir	a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function.
scenario	the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function.

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- getReportFSECStevenLord()

## End(Not run)
```

getReportGridINMS *getReportGridINMS*

Description

Generates and saves a list of reports relevant to the INMS context

Usage

```
getReportGridINMS(
  gdx,
  reportOutputDir = NULL,
  scenario = NULL,
  filter = c(2, 7),
  version = "v13"
)
```

Arguments

gdx	GDX file
reportOutputDir	Directory in which the reports are to be saved. If NULL, a list of reports (MAgPIE objects) is returned instead
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL a list of reports (MAgPIE objects) is returned instead.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
version	Version number for this analysis

Value

A list of reports (MAgPIE objects)

Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder, Michael Crawford

Examples

```
## Not run:
x <- getReportGridINMS(gdx)

## End(Not run)
```

---

getReportGridNitrogenPollution

getReportGridNitrogenPollution

Description

Reports nutrient surplus indicators as well as exceedance of the critical nitrogen surplus at the grid level

Usage

```
getReportGridNitrogenPollution(gdx, reportOutputDir = NULL, scenario = NULL)
```

Arguments

gdx	gdx file
reportOutputDir	a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function.
scenario	the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function.

Value

A list of MAgPIE objects containing the reports

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- getReportGridNitrogenPollution()

## End(Not run)
```

---

getReportINMS *getReportINMS*

Description

Puts together a report for the INMS project based on a MAgPIE gdx file

Usage

```
getReportINMS(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(2, 7),
  detail = TRUE,
  ...
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
detail	Crop specific (TRUE) or aggregated outputs (FALSE)
...	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Benjamin Bodirsky, Florian Humpenoeder

Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

---

getReportIso *getReportIso*

Description

Puts together a report based on a MAgPIE gdx file

Usage

```
getReportIso(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  detail = FALSE,
  ...
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
detail	Crop specific (TRUE) or aggregated outputs (FALSE)
...	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Details

Reports are organize with 'l' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in 'l' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in 'l' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

---

getReportMAgPIE2GAINS *getReportMAgPIE2GAINS*

Description

Puts together a report for the IIASA GAINS model based on a MAgPIE gdx file

Usage

```
getReportMAgPIE2GAINS(
  gdx,
  folder = NULL,
  scenario = NULL,
  filter = c(2, 7),
  ...
)
```

Arguments

gdx	GDX file
folder	a folder name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
...	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder

getReportMAgPIE2LPJmL *getReportMAgPIE2LPJmL*

Description

Puts together a report for LPJmL or other biophysical models based on a MAgPIE gdx file

Usage

```
getReportMAgPIE2LPJmL(
  gdx,
  folder = NULL,
  scenario = NULL,
  filter = c(2, 7),
  ...
)
```

Arguments

gdx	GDX file
folder	a folder name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
...	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder

Examples

```
## Not run:
x <- getReportMAgPIE2LPJmL(gdx)

## End(Not run)
```

getReportMAgPIE2REMIND

getReportMAgPIE2REMIND

Description

Based on a MAgPIE gdx file, a report is generated containing only the variables relevant for the coupling with REMIND. Basically a copy of getReport, but calling less 'reportXY()' functions. Land-use-Change CO2 uses the legacy-clearing reframe (the reportEmissions default).

Usage

```
getReportMAgPIE2REMIND(gdx, file = NULL, scenario = NULL)
```

Arguments

gdx	GDX file
file	A file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.

Details

Reports are organized with '|' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in '|' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will be silently ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in '|' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

Value

A MAgPIE object containing the report.

Author(s)

Florian Humpenoeder, David Klein

Examples

```
## Not run:
x <- getReportMAgPIE2REMIND(gdx)

## End(Not run)
```

---

getReportPBindicators *getReportPBindicators*

Description

Puts together all reporting variables for planetary boundary indicators of MAgPIE

Usage

```
getReportPBindicators(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  ...
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object.
scenario	Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object.
filter	Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA.
...	additional arguments for write.report. Will only be taken into account if argument "file" is not NULL.

Details

Reports are organize with 'l' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in 'l' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in 'l' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

Value

A MAgPIE object containing the report in the case that "file" is NULL.

Author(s)

Felicitas Beier

Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

GrowingStock *GrowingStock*

Description

reads woody growing stock out of a MAgPIE gdx file

Usage

```
GrowingStock(gdx, file = NULL, level = "regglo", indicator = "relative")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate
indicator	If the reported numbers are relative (mio m3/ha) or absolute (mio. m3). Default is relative.

Details

Growing stock for producing woody materials consist of growing stock from plantations (forestry), secondary and primary forest as well as other land (natveg)

Value

Growing stock in m3 per ha

Author(s)

Abhijeet Mishra

Examples

```
## Not run:
x <- GrowingStock(gdx)

## End(Not run)
```

harvested_area_timber *harvested_area_timber*

Description

Reads wood harvest area separated by source (primforest, secdforest, forestry, other) and age classes from a gdx. The data is on cluster level and the unit is Mha per year.

Usage

```
harvested_area_timber(
  gdx,
  file = NULL,
  level = "cell",
  aggregateAgeClasses = TRUE,
  annualized = TRUE
)
```

Arguments

gdx	A fulldata.gdx of a magpie run, usually with endogenous forestry enabled
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate
aggregateAgeClasses	If TRUE, age classes are aggregated
annualized	If TRUE, Mha per year. If FALSE, Mha per time step

Value

Area harvested for wood in Mha per year (annualized = TRUE) or Mha per time step (annualized = FALSE) as a magpie object

Author(s)

Abhijeet Mishra, Pascal Sauer, Florian Humpenoeder

hourlyLaborCosts *hourlyLaborCosts*

Description

returns hourly labor costs in agriculture from MAgPIE results

Usage

```
hourlyLaborCosts(gdx, level = "reg", file = NULL)
```

Arguments

gdx	GDX file
level	spatial aggregation to report employment ("iso", "reg", "glo", or "regglo")
file	a file name the output should be written to using write.magpie

Value

hourly labor costs in agriculture

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- hourlyLaborCosts(gdx)

## End(Not run)
```

Hunger *Hunger*

Description

Calculates the share of people living in hunger.

Usage

```
Hunger(
  gdx,
  level = "reg",
  after_shock = TRUE,
  calibrated = FALSE,
  share = TRUE
)
```

Arguments

gdx	GDX file
level	spatial aggregation. can be "iso","reg","regglo","glo"
after_shock	FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income
calibrated	if calibrated is TRUE, kcal values are calibrated to better match historical years
share	share of population that is undernourished

Value

magpie object with hunger (mio people) or hunger share

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- Hunger(gdx)

## End(Not run)
```

---

income *income*

Description

Calculates income based on a MAgPIE gdx file

Usage

```
income(
  gdx,
  file = NULL,
  level = "reg",
  per_capita = TRUE,
  type = "ppp",
  after_shock = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
per_capita	income per capita or aggregated for the total population
type	ppp for purchase power parity, mer for market exchange rate
after_shock	FALSE is using the exogenous real income, TRUE is using the endogeenous real income that takes into account food price change on real income

Value

annual income as MAgPIE object (unit depends on per_capita: US$2017 MER/cap/yr (TRUE), US$2017 MER/yr (FALSE))

Author(s)

Florian Humpenoeder, Benjamin Bodirsky, Felicitas Beier

Examples

```
## Not run:
x <- income(gdx)

## End(Not run)
```

Intake *Intake*

Description

Calculates the per-capita kcal intake from the food demand model

Usage

```
Intake(
  gdx,
  file = NULL,
  level = "reg",
  calibrated = TRUE,
  pregnancy = FALSE,
  per_capita = TRUE,
  age = FALSE,
  sex = FALSE,
  bmi_groups = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
calibrated	if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used
pregnancy	if TRUE, adding the intake requirements for lactation and pregnancy
per_capita	per capita or aggregated for the population
age	if FALSE age and sex is aggregated
sex	if TRUE, data is provided by sex
bmi_groups	if TRUE data is proided by BMI group

Details

Demand definitions are equivalent to FAO Food supply categories

Value

calories as MAgPIE object (unit depends on per_capita: kcal/cap/day (TRUE), kcal/day (FALSE))

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- Intake(gdx)

## End(Not run)
```

IntakeDetailed *IntakeDetailed*

Description

Calculates detailed or aggregated per-capita kcal intake including exogenous scenarios

Usage

```
IntakeDetailed(gdx, file = NULL, level = "reg", product_aggr = FALSE)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
product_aggr	aggregate over products or not (boolean)

Details

Calculation of kcal food intake is possible for both exogenous diet scenarios and endogenous estimation from food demand model

Value

Calories as MAgPIE object (unit: kcal/cap/day)

Author(s)

Isabelle Weindl

Examples

```
  ## Not run:
    x <- IntakeDetailed(gdx)

## End(Not run)
```

IntakeDetailedProtein *IntakeDetailedProtein*

Description

Calculates food-specific per-capita protein intake from magpie results in grams.

Usage

```
IntakeDetailedProtein(gdx, file = NULL, level = "reg", product_aggr = FALSE)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
product_aggr	aggregate over products or not (boolean)

Value

Protein intake as MAgPIE object (unit: grams/cap/day)

Author(s)

Vartika Singh, Isabelle Weindl

Examples

```
  ## Not run:
    x <- IntakeDetailedProtein(gdx)

## End(Not run)
```

`kayaFractions` *kayaFractions*

Description

Calculates the elements t, d1, d2/d1, ..., dn/dn-1, t/dn in a kaya-like identity of the form t = d1 * d2/d1 * ... * dn/dn-1 * t/dn, based on the variables t, d1, d2, ..., dn.

Usage

```
kayaFractions(data, driverNames = NULL, fixTimeSteps = FALSE)
```

Arguments

`data`	MAgPIE object with target variable and variables to calculate drivers. Needs to have the target variable as first column in the data dimension, and the variables to calculate the drivers in the following columns. I.e. getItems(data, dim = 3) should return c(t, d1, d2, ..., dn). If decomposition is to be calculated for multiple scenarios, the data object can have a scenario dimension in 3.2.
`driverNames`	Names of the drivers in the data object. If NULL, names of the drivers are set based on the names of the variables d1, d2, ..., dn in the data object (i.e. "d2/d1", "d3/d2", ..., "t/dn"). Name of the target variable is always kept as provided in the data object. Default is NULL.
`fixTimeSteps`	Logical. For a consistent decomposition, the time steps in the data object need to be of equal length. If fixTimeSteps is TRUE, the function will check if the time steps are of equal length and if not, will interpolate the data linearly to have equal time steps. If fixTimeSteps is FALSE, the function will only throw a warning if the time steps are not of equal length. Default is TRUE.

Value

The function returns a magpie object containing the target variable t and the drivers of the target variable

Author(s)

Debbora Leip

See Also

kayaLaspeyres

Examples

```
## Not run:
data <- new.magpie(cells_and_regions = c("EUR", "SSA", "USA", "LAM", "IND", "OAS"),
                   years = c(2000, 2005, 2010),
                   names = as.vector(outer(c("Area", "Population", "Supply"),
                                           c("SSP1", "SSP2"), paste, sep = ".")),
                  sets = c("Region", "Year", "Variable", "Scenario"), fill = runif(108))
kayaFractions(data)

## End(Not run)
```

---

kayaLaspeyres *kayaLaspeyres*

Description

Calculates the Laspeyres decomposition of a changes in a target variable t into drivers based on a kaya-like identity. The kaya-like identity needs to be of the form t = d1 * d2/d1 * d3/d2 * ... * dn/dn-1 * t/dn, where d1, d2/d1, ..., dn/dn-1, t/dn are the drivers of change in the target variable t. The Laspeyres decomposition calculates an additive decomposition of the change over time in the target variable based on the changes in the drivers (for formula of the laspeyres decomposition see comment in .laspeyresDriver). The function returns the change in the target variable and the decomposition of the change into the drivers.

Usage

```
kayaLaspeyres(data, driverNames = NULL, type = "relative", fixTimeSteps = TRUE)
```

Arguments

data	MAgPIE object with target variable and variables to calculate drivers. Needs to have the target variable as first column in the data dimension, and the variables to calculate the drivers in the following columns. I.e. getItems(data, dim = 3) should return c(t, d1, d2, ..., dn). If decomposition is to be calculated for multiple scenarios, the data object can have a scenario dimension in 3.2.
driverNames	Names of the drivers in the data object. If NULL, names of the drivers are set based on the names of the variables d1, d2, ..., dn in the data object (i.e. "d2/d1", "d3/d2", ..., "t/dn"). Name of the target variable is always kept as provided in the data object. Default is NULL.

type	Type of the output. "relative" returns the relative change in the target variable from each time step to the next and its decomposition by drivers, "absolute" returns the absolute change over time and its decomposition. Default is "absolute".
fixTimeSteps	Logical. For a consistent decomposition, the time steps in the data object need to be of equal length. If fixTimeSteps is TRUE, the function will check if the time steps are of equal length and if not, will interpolate the data linearily to have equal time steps. If fixTimeSteps is FALSE, the function will only throw a warning if the time steps are not of equal length. Default is TRUE.

Value

The function returns the change in the target variable and the decomposition of the change into the drivers either in percentage points or absolute values (unit of target variable).

Author(s)

Debbora Leip

See Also

kayaFractions

Examples

```
## Not run:
data <- new.magpie(cells_and_regions = c("EUR", "SSA", "USA", "LAM", "IND", "OAS"),
                   years = c(2000, 2005, 2010),
                   names = as.vector(outer(c("Area", "Population", "Supply"),
                                           c("SSP1", "SSP2"), paste, sep = ".")),
                  sets = c("Region", "Year", "Variable", "Scenario"), fill = runif(108))
kayaLaspeyres(data)

## End(Not run)
```

Kcal *Kcal*

Description

Calculates the per-capita kcal consumption from the food demand model

Usage

```
Kcal(
  gdx,
  file = NULL,
  level = "reg",
```

```
  products = "kfo",
  product_aggr = TRUE,
  after_shock = TRUE,
  calibrated = TRUE,
  magpie_input = FALSE,
  attributes = "kcal",
  per_capita = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not (boolean)
after_shock	FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogenous real income that takes into account food price change on real income
calibrated	if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used
magpie_input	TRUE or FALSE. This setting is only activate if arguments "calibrated" and "after_shock" are set to TRUE and else ignored. If set as TRUE, the per-capita kcal consumption values finally entering MAgPIE as input are used, which drive the behaviour of the MAgPIE model, excluding countries not listed in FAO. If set as FALSE, the per-capita kcal consumption values as calculated in the food demand model are used, including countries not listed in FAO.
attributes	unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr").
per_capita	per capita or aggregated for the population

Details

Demand definitions are equivalent to FAO Food supply categories

Value

calories as MAgPIE object (unit depends on per_capita: kcal/cap/day (TRUE), kcal/day (FALSE))

Author(s)

Benjamin Leon Bodirsky, Isabelle Weindl

Examples

```
  ## Not run:
    x <- Kcal(gdx)

## End(Not run)
```

`laborCostsEndo` *laborCostsEndo*

Description

reads MAgPIE endogenous labor costs for crop and livestock production from gdx file

Usage

```
laborCostsEndo(gdx, products = "kcr", file = NULL, level = "grid")
```

Arguments

`gdx`	GDX file
`products`	products for which labor costs should be reported ("kcr" or "kli", for other products use factorCosts())
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation ("grid" or "iso", for regional/global use factorCosts())

Value

MAgPIE object containing labor costs [million US$17]

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- laborCostsEndo(gdx)

## End(Not run)
```

laborProductivity *laborProductivity*

Description

calculates labor productivity in crop sector (kg DM per hour) from a MAgPIE gdx file

Usage

```
laborProductivity(gdx, level = "reg", productAggr = TRUE)
```

Arguments

gdx	GDX file
level	spatial aggregation to report productivity ("cell","reg", "regglo", "glo")
productAggr	Aggregate over products or not (boolean)

Value

labor productivity in crop sector (kg DM per hour)

Author(s)

Xiaoxi Wang, Ruiying Du, Debbora Leip

Examples

```
## Not run:
x <- laborProductivity(gdx)

## End(Not run)
```

land *land*

Description

reads land out of a MAgPIE gdx file

Usage

```
land(
  gdx,
  file = NULL,
  level = "reg",
  types = NULL,
  subcategories = NULL,
  sum = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
types	NULL or a vector of strings. If NULL, all land types are used. Options are "crop", "past", "forestry", "primforest","secdforest, "urban", "other", "primother" and "secdother"
subcategories	NULL or vector of strings. If NULL, no subcategories are returned. Meaningful options are "crop, "forestry" and "other"
sum	determines whether output should be land-type-specific (FALSE) or aggregated over all types (TRUE).

Value

land as MAgPIE object (Mha)

Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Benjamin Leon Bodirsky, Patrick v. Jeetze

See Also

reportLandUse

Examples

```
## Not run:
x <- land(gdx)

## End(Not run)
```

land_price *land_price*

Description

Calculates MAgPIE MAgPIE land shadow prices based on a gdx file

Usage

```
land_price(
  gdx,
  file = NULL,
  level = "reg",
  ignore_lowbound = FALSE,
  absolute = TRUE,
  digits = 4
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
ignore_lowbound	Some shadow prices are positive (see Details), corresponding to a lower bound for that pool. TRUE sets them to 0. Default value: FALSE.
absolute	Should the absolute value of all the marginals be taken into calculations? TRUE (default) of FALSE. See Details.
digits	rounding accuracy for the output

Details

The land price is obtained through marginals of the "oq_cropland" constraint. The majority of these marginals are negative values, and a negligible number of them are positive. This is the consequence of the constraint binding either on upper or lower level. The parameter ignore_lowbound removes all the positive marginals from land price calculation (negligible), and parameter absolute transforms them into negative values (to be all together reported as positive values at the final calculation).

Value

A MAgPIE object containing the land shadow prices (US$2017/ha).

Author(s)

Markus Bonsch, Misko Stevanovic

Examples

```
  ## Not run:
    x <- land_price(level="regglo", products="kcr")

## End(Not run)
```

`landCarbonSink` *Land Carbon Sink Adjustment Factors*

Description

Indirect human-induced emissions in the land use system

Usage

```
landCarbonSink(
  gdx,
  file = NULL,
  level = "reg",
  cumulative = FALSE,
  baseyear = 1995,
  source = "Grassi"
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global).
`cumulative`	Logical; Determines if emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995.
`baseyear`	Baseyear used for cumulative emissions (default = 1995)
`source`	Currently only "Grassi", which uses pre-calculated adjustment factors from Grassi et al 2021 (DOI 10.1038/s41558-021-01033-6). Can be extended in the future to also include "PIK", based on data from LPJmL.

Details

Calculates global and regional Land Carbon Sink Adjustment Factors

Value

Land Carbon Sink Adjustment Factors (Mt CO2 per year or cumulative)

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- landCarbonSink(gdx)

## End(Not run)
```

`landConservation` *landConservation*

Description

reads land conservation information out of a MAgPIE gdx file. Land restoration `'restore'` is reported in Mha/yr by default but can be also reported both over the time step length and cumulatively.

Usage

```
landConservation(
  gdx,
  file = NULL,
  level = "cell",
  cumuRestor = FALSE,
  baseyear = 1995,
  annualRestor = FALSE,
  sum = FALSE
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "cell", "grid", "iso, "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate
`cumuRestor`	Logical; Whether function should report cumulative land restoration.
`baseyear`	Base year used for cumulative land restoration reporting (default = 1995)
`annualRestor`	Logical; Whether function should report annual land restoration.
`sum`	sum over land pools (default = FALSE)

Details

protected areas in primforest, secdforest and other land

Value

protected area in Mha

Author(s)

Florian Humpenoeder, Patrick v. Jeetze

Examples

```
## Not run:
x <- landConservation(gdx)

## End(Not run)
```

`landForestry` *landForestry*

Description

reads and compiles forestry land subcategories from a MAgPIE gdx file

Usage

```
landForestry(gdx, file = NULL, level = "reg")
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

land as MAgPIE object (Mha)

Author(s)

Florian Humpenoeder

See Also

`reportLandUse`

Examples

```
## Not run:
  x <- land(gdx)

## End(Not run)
```

`landTransitionMatrix` *landTransitionMatrix*

Description

reads the land transition matrix out of a MAgPIE gdx file. The transition matrix from module 10 (ov_lu_transitions) only captures optimization-driven land conversions. Two processes in module 35 (natveg) shift area across land types in the presolve (before the optimization), bypassing the transition matrix: (1) disturbance losses transferring primary forest to secondary forest (p35_disturbance_loss_primf), and (2) natural regrowth on other land being reclassified as secondary forest once carbon density exceeds 20 tC/ha (p35_matureseccdf). This function adds both flows to the transition matrix to ensure consistency with land stock changes.

Usage

```
landTransitionMatrix(gdx, file = NULL, level = "reg")
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate

Value

land transition matrix as MAgPIE object (Mha/yr). The full 7x7 matrix including diagonal (persistence) is returned.

Author(s)

Florian Humpenoeder

See Also

`reportLandTransitionMatrix`

Examples

```
## Not run:
x <- landTransitionMatrix(gdx)

## End(Not run)
```

`lastIter` *lastIter*

Description

Returns the value of a parameter in the last iteration

Usage

```
lastIter(gdx, param, MagpieOutput = TRUE)
```

Arguments

`gdx`	GDX file
`param`	Parameter to be returned
`MagpieOutput`	In inelastic model runs, the food demand model is run once in iter1 based on exo prices prescribed in iter0, then the magpie model in iter1. The magpie results therefore appear in iter1. In elastic model runs, iter2 is started by rerunning the food demand model, but if convergence is reached, MAgPIE does not run again. MAgPIE results are therefore still in iter1, even if the iteration is in iter2.

Value

magpie object

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
x <- lastIter(gdx)

## End(Not run)
```

`legacyEmissions` *legacyEmissions*

Description

Reporting-side legacy-clearing tail for land-clearing CO2. MAgPIE books the aboveground carbon of clearing (deforestation + other-land conversion) instantly; bookkeeping models (BLUE, OSCAR, Houghton & Castanheira) spread it over a slash/deadwood decay tail. This re-spreads the pulse over a first-order-decay (FOD) pool, as `carbonLTS` does for harvested wood products (HWP), and returns the net correction to the raw flux, its Storage/Release parts, and the pool carbon stock.

Usage

```
legacyEmissions(
  gdx,
  file = NULL,
  level = "cell",
  unit = "element",
  cumulative = FALSE,
  baseyear = 1995,
  priming = "hist",
  primingStart = 1850,
  harvestSlashFrac = 0,
  a0Biome = c(A = 0.25, B = 0.25, C = 0.2, D = 0.15, E = 0.15),
  halfLifeBiome = c(A = 4, B = 8, C = 10, D = 20, E = 25)
)
```

Arguments

`gdx`	GDX file
`file`	file name to write the output to with write.magpie
`level`	aggregation level: "cell", "reg", "glo", "regglo", or any superAggregateX level
`unit`	"element" (Mt C/yr) or "gas" (Mt CO2/yr)
`cumulative`	report annually (FALSE) or cumulatively from `baseyear` (TRUE)
`baseyear`	baseyear for cumulative emissions (default 1995)
`priming`	assumed pre-firstYear clearing history that sets the pool's initial stock: "hist" (default, the bookkeeping-ensemble LUC-CO2 shape), "ramp" (linear 0.3->1.0), "half" (flat 0.5x) or "peak" (flat 1x). The only non-mass-conserving element - it credits pre-firstYear clearing MAgPIE never modelled (cf. carbonLTS's 1970 seed).
`primingStart`	first year of the priming window (default 1850; results insensitive below ~1950).
`harvestSlashFrac`	fraction of the wood-harvest aboveground flux added as dead slash (default 0, OFF); a coarse sensitivity switch that does NOT net out the HWP carbon carbonLTS already defers.

a0Biome — immediate/combustion fraction per Koeppen main group (A tropical, B arid, C temperate, D boreal, E polar). Default A 0.25 / B 0.25 / C 0.20 / D 0.15 / E 0.15, from Houghton et al. 2012's clearing-year burn partition (doi:10.5194/bg-9-5125-2012).

halfLifeBiome — slash/deadwood decay half-life (years) per Koeppen group. Default A 4 / B 8 / C 10 / D 20 / E 25, from coarse-woody-debris decay data (Chambers et al. 2000, Harmon et al. 2020, Russell et al. 2014).

Details

The pulse is aboveground (vegc+litc) clearing CO2 only - deforestation + other-land conversion; soil carbon and degradation are excluded. It is interpolated from MAgPIE's 5/10-yr steps to annual temporal resolution, split by Koeppen group (fast decay in the tropics, slow in the boreal) with a separate pool per group, then summed; the fraction a0 is released immediately and the rest (1-a0) decays with the group half-life (see the code for the recursion). y1995 has no reported clearing flux, so the pulse starts at the first reported year, anchored to the mean of the first few.

Caveats: (1) reporting reframe only - the model still emits instantaneously and the carbon price is unchanged; raw emissions stay available via legacyEmis = FALSE. (2) not mass-conserving: the priming credit is a one-off initial pool; being a linear decay it is proportional to the priming amplitude, so half = exactly 0.5x peak and the default hist sits near peak. (3) a residual scope gap to bookkeeping remains. The cumulative legacy series is the timestep-weighted integral of the yearly one (both at the model timesteps); carbon still in the pool at 2100 never enters the reported window.

Value

MAgPIE object (region x year x name) with "legacy_net" (delta to add to Land-use Change), "legacy_storage" (<= 0, deferred emission), "legacy_release" (>= 0, decay outflow) and "legacy_stock" (the slash-pool carbon level, incl. the primed initial stock). legacy_net = legacy_storage + legacy_release = -diff(legacy_stock) by construction. Mt C or CO2 (per `unit`); cumulative fluxes are rebased to `baseyear` as in `carbonLTS` (legacy_stock, a level, is returned only for cumulative = FALSE).

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- legacyEmissions(gdx)

## End(Not run)
```

`LivestockDemStructure` *LivestockDemStructure*

Description

Calculates the share of different livestock commodities in total livestock product consumption on the basis of chosen attribute

Usage

```
LivestockDemStructure(
  gdx,
  file = NULL,
  level = "reg",
  after_shock = TRUE,
  calibrated = TRUE,
  attributes = "kcal",
  fish = FALSE
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
`after_shock`	FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income
`calibrated`	if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used
`attributes`	unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr").
`fish`	if true, livestock share includes fish, otherwhise not

Value

magpie object with the livestock demand structure in a region or country. Unit is dimensionsless, but value depends on chosen attribute

Author(s)

Isabelle Weindl

Examples

```
  ## Not run:
    x <- LivestockDemStructure(gdx)

## End(Not run)
```

`LivestockShare` *LivestockShare*

Description

Calculates the livestock share from the food demand model

Usage

```
LivestockShare(
  gdx,
  file = NULL,
  level = "reg",
  after_shock = TRUE,
  calibrated = TRUE,
  attributes = "kcal",
  fish = TRUE
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
`after_shock`	FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income
`calibrated`	if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used
`attributes`	unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr").
`fish`	if true, livestock share includes fish, otherwhise not

Value

magpie object with the livestock share in a region or country. Unit is dimensionsless, but value depends on chosen attribute

Author(s)

Benjamin Bodirsky

Examples

```
  ## Not run:
    x <- LivestockShare(gdx)

## End(Not run)
```

`m_yeardiff` *m_yeardiff*

Description

Calculates the parameter m_yeardiff, which is a macro within MAgPIE.

Usage

```
m_yeardiff(gdx)
```

Arguments

`gdx`	GDX file

Value

a magpie object with the length of each timestep

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- m_yeardiff(gdx)

## End(Not run)
```

malmquist *malmquist*

Description

calcluates malmquist index based on a MAgPIE gdx file

Usage

```
malmquist(gdx, file = NULL)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie

Value

A MAgPIE object containing the malmquist index

Author(s)

Xiaoxi Wang

Examples

```
  ## Not run:
    x <- malmquist(gdx)

## End(Not run)
```

ManureExcretion *ManureExcretion*

Description

downscales Manure Excretion

Usage

```
ManureExcretion(
  gdx,
  level = "reg",
  products = "kli",
  awms = c("grazing", "stubble_grazing", "fuel", "confinement"),
  agg = TRUE
)
```

Arguments

gdx	GDX file
level	aggregation level: glo, reg, cell, grid, iso
products	livestock products
awms	large animal waste management categories: "grazing","stubble_grazing","fuel","confinement"),
agg	aggregation over "awms" or over "products".

Value

MAgPIE object

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- ManureExcretion(gdx)

## End(Not run)
```

---

mappingToLongFormat *mappingToLongFormat*

Description

Derives a report aggregation mapping (sourceRegion to targetRegion) from a country-level region mapping. The input must have at least the columns `countryname`, `country`, and `region`. Any additional columns are treated as definitions of aggregated macro regions: each unique non-empty value defines a target region, and all source regions containing at least one country assigned to that value are mapped to it.

The output mapping has three blocks (each sorted by sourceRegion):

1. each source region mapped to itself (regional resolution)
2. each source region mapped to `"GLO"` (global aggregate)
3. source regions mapped to every additional aggregated region they belong to

Usage

```
mappingToLongFormat(mappingOrFileName)
```

Arguments

`mappingOrFileName`
: Either a data frame containing the country-level region mapping, or a character string giving the path to a CSV-file.

Value

A data frame with columns `sourceRegion` and `targetRegion`.

Note

This function is a workaround as in magpie4 we can not use the same mapping formulas that we could use in madrat functions. If the magpie4 aggregation logic is ever rewritten, this function may become obsolete.

Author(s)

Kristine Karstens, Patrick Rein

See Also

Other Spatial: `addGeometry()`, `clusterOutputToTerraVector()`, `gdxAggregate()`, `superAggregateX()`

`metadata_comments` *metadata_comments*

Description

set metadata comments to magpie4 objects

Usage

```
metadata_comments(x, unit, description, comment, note)
```

Arguments

`x`	magpie object (magpie4)
`unit`	provide unit
`description`	provide short description
`comment`	optional comment
`note`	optional note

Value

vector of comments following order of input (unit, description, comment, note - further: origin, creation data)

Author(s)

Benjamin Bodirsky, Jannes Breier

See Also

Other Infrastructure: `expectVariablesPresent()`, `out()`, `tryList()`, `tryReport()`

Examples

```
## Not run:
  x <- metadata_comments(x,unit,description,comment,note)

## End(Not run)
```

`modelstat` *modelstat*

Description

MAgPIE model stat of all optimizations - main optimization and (if used) presolve optimization.

Usage

```
modelstat(gdx, file = NULL)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie

Value

A MAgPIE object containing the modelstat

Author(s)

Jan Philipp Dietrich

Examples

```
## Not run:
  x <- modelstat(gdx)

## End(Not run)
```

NetForestChange *NetForestChange*

Description

Calculates net and gross forest area change based on a MAgPIE gdx file

Usage

```
NetForestChange(gdx, file = NULL, level = "cell", lowpass = NULL)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregateX
lowpass	number of lowpass filter iterations (default = NULL)

Value

Net Forest Change as MAgPIE object (Mha per year)

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- NetForestChange(gdx)

## End(Not run)
```

NitrogenBudget *NitrogenBudget*

Description

calculates projections of Nitrogen Budgets for Croplands (Tg Nr per) from a MAgPIE gdx file

Usage

```
NitrogenBudget(
  gdx,
  include_emissions = FALSE,
  level = "reg",
  debug = FALSE,
  cropTypes = FALSE,
  threshold = 0.05,
  progress = TRUE
)
```

Arguments

gdx	GDX file
include_emissions	TRUE also divides the N surplus into different emissions
level	aggregation level, reg, glo or regglo, cell, iso or grid
debug	debug mode TRUE makes some consistency checks between estimates for different resolutions.
cropTypes	FALSE for aggregate results; TRUE for crop-specific results
threshold	passed to mstools::toolFertilizerDistribution
progress	passed to mstools::toolFertilizerDistribution

Author(s)

Benjamin Leon Bodirsky, Michael Crawford, Edna J. Molina Bacca, Florian Humpenoeder

Examples

```
## Not run:
x <- NitrogenBudget(gdx)

## End(Not run)
```

NitrogenBudgetNonagland

NitrogenBudgetNonagland

Description

calculates projections of Nitrogen Budgets for non-agricutlural land from a MAgPIE gdx file

Usage

```
NitrogenBudgetNonagland(gdx, level = "reg")
```

Arguments

gdx	GDX file
level	aggregation level, reg, glo or regglo

Author(s)

Benjamin Leon Bodirsky, Edna J. Molina Bacca

Examples

```
  ## Not run:
    x <- NitrogenBudgetNonagland(gdx)

## End(Not run)
```

NitrogenBudgetPasture *NitrogenBudgetPasture*

Description

calculates projections of Nitrogen Budgets for Croplands from a MAgPIE gdx file

Usage

```
NitrogenBudgetPasture(gdx, include_emissions = FALSE, level = "reg")
```

Arguments

gdx	GDX file
include_emissions	TRUE also divides the N surplus into different emissions
level	aggregation level, reg, glo or regglo, cell, grid, iso

Author(s)

Benjamin Leon Bodirsky, Edna J. Molina Bacca

Examples

```
## Not run:
x <- NitrogenBudgetPasture(gdx)

## End(Not run)
```

NitrogenBudgetWithdrawals *NitrogenBudgetWithdrawals*

Description

calculates projections of Nitrogen Budgets withdrawals for Croplands from a MAgPIE gdx file

Usage

```
NitrogenBudgetWithdrawals(gdx, kcr = "sum", net = TRUE, level = "reg")
```

Arguments

gdx	GDX file
kcr	"sum" provides the totals over all crops, "kcr" provides outputs by kcr
net	TRUE only provides total net-withdrawals, otherwise all categories are returned (fixation and seed are returned positive, not negative)
level	aggregation level, reg, glo or regglo, cell, grid or iso

Author(s)

Benjamin Leon Bodirsky, Michael Crawford

Examples

```
## Not run:
x <- NitrogenBudgetWithdrawals(gdx)

## End(Not run)
```

OtherLand *OtherLand*

Description

Disaggregation of other land into initial, restored and recovered land based on a MAgPIE gdx file

Usage

```
OtherLand(gdx, level = "reg")
```

Arguments

gdx	GDX file
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any aggregation level defined in superAggregateX. In addition "climate" for the 3 climate regions tropical, temperate and boreal is available.

Details

initial, restored and recovered land

Value

Other land area in Mha

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- PeatlandArea(gdx)

## End(Not run)
```

---

out *out*

Description

Function to safely returns parameters. Function returns either the output or writes it to a file. Please use this function when you write own GDX output functions.

Usage

```
out(x,file)
```

Arguments

x	an object that can be converted to a MAgPIE object
file	file name of a file it should be written to. NULL, if x should be returned instead to be written to a file.

Value

NULL or x as MAgPIE object

Author(s)

Jan Philipp Dietrich

See Also

Other Infrastructure: `expectVariablesPresent()`, `metadata_comments()`, `tryList()`, `tryReport()`

`outputCheck` *outputCheck*

Description

Function to check a MAgPIE gdx file for known problems (e.g. non-zero dummy variables). The function will throw warnings for problem found in the outputs.

Usage

```
outputCheck(gdx)
```

Arguments

gdx	GDX file

Author(s)

Jan Philipp Dietrich

Examples

```
## Not run:
outputCheck(gdx)

## End(Not run)
```

`outputPerWorker` *outputPerWorker*

Description

returns output per worker in crop+livestock production

Usage

```
outputPerWorker(gdx, level = "reg", file = NULL)
```

Arguments

gdx	GDX file
level	spatial aggregation to report employment ("reg", "glo", "regglo", or custom region aggregation)
file	a file name the output should be written to using write.magpie

Value

output per worker as magpie object

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- outputPerWorker(gdx)

## End(Not run)
```

PeatlandArea *PeatlandArea*

Description

reads peatland area out of a MAgPIE gdx file

Usage

```
PeatlandArea(gdx, file = NULL, level = "cell", sum = TRUE)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any aggregation level defined in superAggregateX. In addition "climate" for the 3 climate regions tropical, temperate and boreal is available.
sum	sum over land types TRUE (default) or FALSE

Details

Intact, degraded and rewettet peatland area

Value

Peatland area in Mha

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- PeatlandArea(gdx)

## End(Not run)
```

PeatlandEmissions *PeatlandEmissions*

Description

reads peatland GHG emissions out of a MAgPIE gdx file

Usage

```
PeatlandEmissions(
  gdx,
  file = NULL,
  level = "cell",
  unit = "gas",
  cumulative = FALSE,
  baseyear = 1995,
  lowpass = 0,
  sum = TRUE,
  intact = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any aggregation level defined in superAggregateX. In addition "climate" for the 3 climate regions tropical, temperate and boreal is available.
unit	global warming potential (GWP100AR6) or gas (gas)
cumulative	FALSE (default) or TRUE
baseyear	Baseyear used for cumulative emissions (default = 1995)

lowpass	number of lowpass filter iterations (default = 0)
sum	sum over land types TRUE (default) or FALSE
intact	report GHG emissions from intact peatlands FALSE (default) or TRUE

Details

Peatland GHG emissions: CO2, DOC, CH4 and N2O

Value

Peatland GHG emissions in Mt CO2eq (if unit="gwp") or Mt of the respective gas (if unit="gas")

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- PeatlandArea(gdx)

## End(Not run)
```

PlantationEstablishment

PlantationEstablishment

Description

reads carbon stocks in harvested timber out of a MAgPIE gdx file

Usage

```
PlantationEstablishment(gdx, file = NULL, level = "cell")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global), custom regation aggregation or any secdforest aggregation level defined in superAggregateX

Details

Area newly established in current time step for future timber production

Value

Area newly for timber production

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- PlantationEstablishment(gdx)

## End(Not run)
```

`population` *population*

Description

reads population out of a MAgPIE gdx file

Usage

```
population(
  gdx,
  file = NULL,
  level = "reg",
  age = FALSE,
  sex = FALSE,
  bmi_groups = FALSE
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
`age`	if TRUE, population is split up by age groups
`sex`	if TRUE, population is split up by sex
`bmi_groups`	if TRUE, the population will be split up in body-mass-index groups.

Value

population as MAgPIE object (million people)

Author(s)

Florian Humpenoeder, Benjamin Bodirsky, Isabelle Weindl

See Also

reportPopulation

Examples

```
## Not run:
x <- population(gdx)

## End(Not run)
```

PriceElasticities *PriceElasticities*

Description

Calculates the physical elasticity for food demand

Usage

```
PriceElasticities(
  gdx,
  file = NULL,
  level = "reg",
  calibrated = TRUE,
  products = "kfo"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
calibrated	if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used
products	set of the products for which the elasticity should be estimated. Please note that this stills remains an elasticity relative to total food expenditure. So its the change in consumption of one good when the prices of all products change according to the scenario.

Value

magpie object with the livestock share in a region or country. Unit is dimensionsless, but value depends on chosen attribute

Author(s)

Benjamin Bodirsky

Examples

```
## Not run:
  x <- PriceElasticities(gdx)

## End(Not run)
```

PriceGHG *PriceGHG*

Description

reads GHG emission prices out of a MAgPIE gdx file

Usage

```
PriceGHG(gdx, file = NULL, level = "reg", aggr = "weight")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
aggr	aggregation used, currently only "weight" (weighted by population) (max is deprecated)

Value

GHG emission prices as MAgPIE object (US$2017/tCO2, US$2017/tN2O, US$2017/tCH4)

Author(s)

Florian Humpenoeder, Amsalu W. Yalew

See Also

reportPriceGHG

Examples

```
  ## Not run:
    x <- PriceGHG(gdx)

## End(Not run)
```

---

priceIndex *priceIndex*

Description

calcluates price indicies based on a MAgPIE gdx file

Usage

```
priceIndex(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  index = "lasp",
  chain = FALSE,
  baseyear = "y2005",
  round = TRUE,
  type = "consumer",
  product_aggr = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
index	"lasp" (Laspeyres-Index: baseyear weighting), "paas" (Paasche-Index: current weighting), "fish" (Fisher-Index: geometric mean of "lasp" and "paas")
chain	Chain Index: if true, the base period for each time period is the immediately preceding time period. Can be combined with all of the above indices
baseyear	baseyear of the price index
round	shall the results be rounded?

type	For whom are the prices important? "producer" are the prices that farmer face, as they also produce intermediate products (seed, feed). "consumer" are the prices for the end consumer faces (supermarket, bioenergy plant). Currently, the only difference is the basket composition (ideally, also prices should differ between regions)
product_aggr	aggregate over products or not (boolean)

Value

A MAgPIE object containing price indices for consumers or producers (depending on type)

Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Benjamin Bodirsky

Examples

```
  ## Not run:
    x <- priceIndex(gdx)

## End(Not run)
```

`priceIndexFood` *priceIndexFood*

Description

calcluates price indicies based on a MAgPIE gdx file

Usage

```
priceIndexFood(
  gdx,
  file = NULL,
  level = "reg",
  index = "lasp",
  chain = FALSE,
  baseyear = "y2005",
  round = TRUE,
  product_aggr = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
index	"lasp" (Laspeyres-Index: baseyear weighting), "paas" (Paasche-Index: current weighting), "fish" (Fisher-Index: geometric mean of "lasp" and "paas")
chain	Chain Index: if true, the base period for each time period is the immediately preceding time period. Can be combined with all of the above indices
baseyear	baseyear of the price index. type model to take baseyear 2010 with literature prices
round	shall the results be rounded?
product_aggr	aggregate over products or not (boolean)

Value

A MAgPIE object containing price indices for consumers or producers (depending on type)

Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Benjamin Bodirsky

Examples

```
  ## Not run:
    x <- priceIndexFood(gdx)

## End(Not run)
```

---

prices *prices*

Description

calcluates prices based on a MAgPIE gdx file

Usage

```
prices(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = FALSE,
```

```
  attributes = "dm",
  type = "consumer",
  glo_weight = "production"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not (boolean)
attributes	USD17MER per ton X (dm,nr,p,k,wm) except gross energy (ge) where it is USD17MER per GJ
type	"consumer" or "producer" prices. Producers' prices are calculated on the regional level as a sum of regional trade equation marginal values and respective global trade equation marginal values.For the non traded commodities, both global and regional producers prices are set to zero instead of NaN.
glo_weight	Decides the calculation of global prices. Weighting schemes are applied for estimation of global producer price. If `"export"` prices are calculated as average of regional exporters' prices, weighted by the export volumes. If `"production"` (default), prices are calculated as average of regional prices weighted by regional production. If `"free_trade"`, the global prices are directly taken from the shadow prices of the global trade constraint, and no averaging is performed.

Value

A MAgPIE object containing the consumer's or producers' prices (unit depends on attributes)

Author(s)

Misko Stevanovic, Florian Humpenoeder, Jan Philipp Dietrich, Xiaoxi Wang, Edna J. Molina Bacca

Examples

```
## Not run:
x <- prices(gdx)

## End(Not run)
```

primaryPerSecondary *primaryPerSecondary*

Description

Calculates the amount of primary products needed per unit of secondary (and tertiary) product, accounting for processing chains and co-product allocation. This function traces back from secondary/tertiary products through processing stages to the original primary crop inputs.

Usage

```
primaryPerSecondary(gdx, file = NULL, level = "reg", allocation = "mass")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
allocation	Method for allocating primary product use when co-products are produced. Options: "mass" (default): Allocation based on dry matter content of outputs "value": Allocation based on economic value of outputs "none": No allocation - full primary input attributed to each output (sum of allocations > 100 percent when co-products exist)

Value

MAgPIE object containing the amount of primary product needed per unit of secondary product (tDM/tDM)

Author(s)

Kristine Karstens, David M Chen

See Also

production, trade

Examples

```
## Not run:
x <- primaryPerSecondary(gdx)

## End(Not run)
```

PrimSecdOtherLand *PrimSecdOtherLand*

Description

Calculates share of primary and secondary non-forest vegetation for different aggregation levels based on gridded magpie output and initial shares of primary and secondary non-forest vegetation.

Usage

```
PrimSecdOtherLand(
  x,
  ini_file,
  ini_year = "y1995",
  file = NULL,
  level = "grid",
  unit = "Mha"
)
```

Arguments

x	Time series of land pools (model output) containing only one aggregated class for other land. Can be a file or magclass object.
ini_file	Initialisation file for primary and secondary other land (e.g. based on 1995 MAgPIE land-use initialisation values). Must have the same spatial resolution as x.
ini_year	Reference year for estimating primary and secondary other land shares, must be included in ini_file.
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate. The unit of output for the cases above is Mha. If level "grid" is specified the unit of output can be chosen between "share" and "Mha".
unit	"Mha" or "share". Defines the unit of the gridded output, see also level.

Value

x including land area for primary and secondary non-forested vegetation in MAgPIE (other land) as MAgPIE object; either as unit of area (Mha) or as fraction of total land per grid cell (share).

Author(s)

Patrick v. Jeetze, Kristine Karstens

Examples

```
## Not run:
x <- "./cell.land_0.5.nc"
land <- PrimSecdOtherLand(x)

# direct use of disaggregation output
land <- PrimSecdOtherLand(land_hr)

## End(Not run)
```

`processing` *processing*

Description

Calculates MAgPIE disaggregated processing out of a gdx file

Usage

```
processing(gdx, level = "reg", indicator = "secondary_from_primary")
```

Arguments

`gdx`	GDX file
`level`	Level of regional aggregation ("reg", "glo", "regglo")
`indicator`	process or secondary product output

Details

Demand definitions are equivalent to FAO CBS categories

Value

processing as MAgPIE object (Unit depends on attributes)

Author(s)

David Chen, Benjamin Leon Bodirsky

Examples

```
## Not run:
 x <- processing(gdx = gdx, level="regglo", products="kcr", indicator="primary_to_process")

## End(Not run)
```

productEmissions *productEmissions*

Description

Calculates GHG emissions by product, allocating emissions from different sources (land-use change, enteric fermentation, manure management, etc.) to specific agricultural products based on cropland shares, livestock feed demand, and other factors. This is an improved version with better code style and documentation.

Usage

```
productEmissions(gdx, unit = "GWP100AR6", level = "reg", perTonne = TRUE)
```

Arguments

gdx	GDX file
unit	"element", "gas", "GWP100AR5", "GWP100AR6", "GWP*AR5", or "GWP*AR6" "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr "gas": co2_c in Mt CO2/yr, n2o_n in Mt N2O/yr, ch4 in Mt CH4/yr "GWP": co2_c in Mt CO2eq/yr, n2o_n in Mt CO2eq/yr, ch4 in Mt CO2eq/yr
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global)
perTonne	If TRUE, returns emissions intensity (per tonne of production); if FALSE, returns total emissions

Value

GHG emissions as MAgPIE object (Unit depends on unit parameter)

Author(s)

David M Chen

See Also

Emissions, emisCO2

Examples

```
## Not run:
  x <- productEmissions(gdx, unit = "GWP100AR6", level = "reg")
  xIntensity <- productEmissions(gdx, unit = "GWP100AR6", perTonne = TRUE)

## End(Not run)
```

production *production*

Description

reads production out of a MAgPIE gdx file

Usage

```
production(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = FALSE,
  attributes = "dm",
  water_aggr = TRUE,
  cumulative = FALSE,
  baseyear = 1995
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not (boolean)
attributes	dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector.
water_aggr	aggregate irrigated and non-irriagted production or not (boolean).
cumulative	Logical; Determines if production is reported annually (FALSE, default) or cumulative (TRUE)
baseyear	Baseyear used for cumulative production (default = 1995)

Value

production as MAgPIE object (unit depends on attributes and cumulative)

Author(s)

Benjamin Leon Bodirsky

See Also

reportProduction, demand

Examples

```
## Not run:
x <- production(gdx)

## End(Not run)
```

---

productionProfit *productionProfit*

Description

calcluates aggregate producer profit based on a MAgPIE gdx file.

Usage

```
productionProfit(gdx, file = NULL, level = "reg")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing producers profit in million $US.

Author(s)

Miodrag Stevanovic

Examples

```
## Not run:
  x <- productionProfit(gdx)

## End(Not run)
```

productionRevenue *productionRevenue*

Description

calcluates production revenue based on a MAgPIE gdx file.

Usage

```
productionRevenue(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	ggregate over products or not (boolean, default TRUE)

Value

A MAgPIE object containing prodcution revenues.

Author(s)

Miodrag Stevanovic

Examples

```
## Not run:
  x <- productionRevenue(gdx)

## End(Not run)
```

`readGDXBilateral` *readGDXBilateral*

Description

Internal support function to read bilateral spatial data from GDX files.

Usage

```
readGDXBilateral(gdx, symbol)
```

Arguments

`gdx`	gdx file to report from
`symbol`	name of the symbol to read from gdx

Author(s)

David M Chen

`reassignLivestockPathway` *reassignLivestockPathway*

Description

Collapse every livestock (`kli`) product's prim/secd/feed pathway split into a single `feed` (Livestock) pathway, leaving all non-livestock products untouched. The Kastner prim/secd/feed pathway split in `embodiedResourceKastner` is keyed on each product's demand share, so a livestock product's OWN footprint (enteric/manure emissions, herding labour, drinking water) lands in the prim/secd pathway of the `kli` product. To show the FULL livestock footprint in the Livestock pathway, this moves each `kli` product's whole footprint (all pathways) into `feed`. Non-`kli` feed crops/pasture keep their pathway, so they still count as Livestock via the `feed` pathway.

Works on both layouts produced by `embodiedResourceKastner`: the bilateral object (dim 3 = `pathway.product`) and the regional accounting object (dim 3 = `accounting.pathway.product`); the pathway and product sub-dimensions are located by name/position, so the object must carry a sub-dimension named `pathway` with the product as the last sub-dimension of dim 3. The operation is a no-op when no `kli` product is present (e.g. the land resource, whose livestock land sits in the feed pathway of crops) and is idempotent (re-applying it changes nothing). Each product's total (summed over pathway) is conserved.

Usage

```
reassignLivestockPathway(x, kli = NULL, gdx = NULL)
```

Arguments

x	MAgPIE object with a pathway sub-dimension in dim 3 and the product as the last sub-dimension of dim 3 (bilateral or regional output of embodiedResourceKastner and its embodied*Kastner wrappers).
kli	character vector of livestock product names (the kli set). Alternatively supply gdx to read it.
gdx	optional GDX file to read the kli set from when kli is not given.

Value

MAgPIE object of the same layout as x, with all kli products' footprint moved to the feed pathway.

Author(s)

David M Chen

See Also

embodiedResourceKastner, footprints

---

relativeHourlyLaborCosts *relativeHourlyLaborCosts*

Description

calculates labor costs per ag. worker in relation to GDP pc

Usage

```
relativeHourlyLaborCosts(gdx, level = "reg", file = NULL)
```

Arguments

gdx	GDX file
level	spatial aggregation to report ("iso", "reg", "glo", "regglo", or custom region aggregation)
file	a file name the output should be written to using write.magpie

Value

labor costs per ag. worker in relation to GDP pc

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- relativeHourlyLaborCosts(gdx)

## End(Not run)
```

---

reportAAI *reportAAI*

Description

reports area actually irrigated

Usage

```
reportAAI(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

Area actually irrigated as MAgPIE object. Unit: see names

Area actually irrigated variables

Name	Unit	Meta
Resources|Land Cover|Cropland|Area actually irrigated	million ha	Cropland area actually receiving irrigation

Author(s)

Stephen Wirth, Anne Biewald

Examples

```
## Not run:
x <- reportAEI(gdx)

## End(Not run)
```

reportAEI *reportAEI*

Description

reports Area equipped for Irrigation

Usage

```
reportAEI(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

Area equipped for Irrigation as MAgPIE object. Unit: see names

Area equipped for irrigation variables

Name	Unit	Meta
Resources|Land Cover|Cropland|Area equipped for irrigation	million ha	Cropland area equipped with irrigation infrastruc

Author(s)

Stephen Wirth

Examples

```
  ## Not run:
    x <- reportAEI(gdx)

## End(Not run)
```

reportAgEmployment *reportAgEmployment*

Description

reports employment in crop+livestock production from MAgPIE results

Usage

```
reportAgEmployment(gdx, type = "absolute", detail = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
type	"absolute" for total number of people employed, "share" for share out of working age population
detail	if TRUE, employment is disaggregated to crop and livestock production, if FALSE only aggregated employment is reported
level	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

agricultural employment as MAgPIE object

Agricultural employment variables

Name	Unit	Meta
Labor|Employment|Agricultural employment	mio people	Total agricultural employment
Labor|Employment|Agricultural employment|+|Crop products	mio people	Employment in crop production
Labor|Employment|Agricultural employment|+|Livestock products	mio people	Employment in livestock produc
Labor|Employment|Share of working age population employed in agriculture	%	Agricultural employment as sha

Author(s)

Debbora Leip

Examples

```
## Not run:
  x <- reportAgEmployment(gdx)

## End(Not run)
```

reportAgGDP *reportAgGDP*

Description

reports MAgPIE Agricultural GDP Mio. USD05 MER

Usage

```
reportAgGDP(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

Magpie object

Agricultural GDP variables

Name	Unit	Meta
Value|Agriculture GDP	million US$2017/yr	Agricultural value added (GDP from agriculture sector)

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
x <- reportAgGDP(gdx)

## End(Not run)
```

reportAgriResearchIntensity
reportAgriResearchIntensity

Description

reports Agricultural Research Intensity as % of total GDP

Usage

```
reportAgriResearchIntensity(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

magpie object

Agricultural research intensity variables

Name	Unit	Meta
Agricultural Research Intensity	% of Total GDP	Share of GDP spent on agricultural research

Author(s)

David Chen

Examples

```
## Not run:
x <- reportAgriResearchIntensity(gdx)

## End(Not run)
```

`reportAnthropometrics` *reportAnthropometrics*

Description

reports Underweight, Normalweight, Overweight and Obesity as well as body height for males and females

Usage

```
reportAnthropometrics(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

Magpie object

Anthropometrics variables

Name	Unit	Meta
Nutrition|Anthropometrics|People underweight	million people	Population with underweight BMI
Nutrition|Anthropometrics|People normalweight	million people	Population with normal BMI
Nutrition|Anthropometrics|People overweight	million people	Population with overweight BMI
Nutrition|Anthropometrics|People obese	million people	Population with obesity
Nutrition|Anthropometrics|Body height of female adults	cm/capita	Average body height of adult females
Nutrition|Anthropometrics|Body height of male adults	cm/capita	Average body height of adult males

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
x <- reportBodyweight(gdx)

## End(Not run)
```

reportBII *reportBII*

Description

reports biodiversity intactness index

Usage

```
reportBII(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

Biodiversity intactness index as MAgPIE object

Biodiversity Intactness Index variables

Name	Unit	Meta
Biodiversity|BII	unitless	Terrestrial biodivers
Biodiversity|Cropland Landscapes BII	unitless	Terrestrial biodivers
Biodiversity|BII in 30x30 Landscapes	unitless	Terrestrial biodivers
Biodiversity|Biodiversity Hotspot BII	unitless	Terrestrial biodivers
Biodiversity|Biodiversity Hotspot and Intact Forest Landscapes BII	unitless	Terrestrial biodivers
Biodiversity|BII in areas outside Biodiversity Hotspots, Intact Forest & Cropland Landscapes	unitless	Terrestrial biodivers
Biodiversity|Key Biodiversity Area BII	unitless	Terrestrial biodivers

Author(s)

Patrick v. Jeetze, Florian Humpenoeder

Examples

```
## Not run:
x <- reportBII(gdx)

## End(Not run)
```

reportBiochar *reportBiochar*

Description

reports production and carbon storage of biochar

Usage

```
reportBiochar(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	The aggregation level to be used ("regglo" by default)

Value

Biochar production and carbon storage as MAgPIE object

Author(s)

Kristine Karstens, Isabelle Weindl

Examples

```
  ## Not run:
    x <- reportBiochar(gdx)

## End(Not run)
```

reportBiogasFeedstock *reportBiogasFeedstock*

Description

Reports biogas feedstock from a MAgPIE GDX file at ISO country level. Currently covers two sources: manure allocated to digesters and fodder crops (placeholder, all zeros until MAgPIE tracks foddr/forage crop allocation to biogas). Manure is converted from nitrogen mass (Mt N) to energy (PJ) using livestock-specific heating values and nitrogen contents from Hoyos-Sebá et al. (2024).

Usage

```
reportBiogasFeedstock(gdx, file = NULL)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie

Value

MAgPIE object with biogas feedstock potential ("manure", "fodder") at ISO level in PJ.

Author(s)

Kristine Karstens

Examples

```
## Not run:
  x <- reportBiogasFeedstock(gdx)

## End(Not run)
```

---

reportBioplasticDemand *reportBioplasticDemand*

Description

reports demand for bioplastic and demand for substrate for bioplastic production from MAgPIE results

Usage

```
reportBioplasticDemand(gdx, detail = TRUE, level = "regglo")
```

Arguments

gdx	GDX file
detail	only relevant for substrate demand. If TRUE, substrate demand is disaggregated by crop type, if FALSE only the aggregated demand is reported.
level	spatial aggregation to report bioplastic/substrate demand (only "reg" or "regglo")

Value

bioplastic and bioplastic substrate demand as MAgPIE object

Bioplastic demand variables

Name	Unit	Meta
Demand for bioplastic	Mt/yr	Total bioplastic demand
Demand for bioplastic substrate|Total	Mt DM/yr	Total substrate demand for bioplastic production

Author(s)

Debbora Leip

Examples

```
## Not run:
  x <- reportBioplasticDemand(gdx)

## End(Not run)
```

`reportCarbonstock` *reportCarbonstock*

Description

Reports the carbon stocks for future MAgPIE projections

Usage

```
reportCarbonstock(gdx, level = "regglo", legacyEmis = TRUE)
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)
`legacyEmis`	Logical (default TRUE). If TRUE, add the legacy-clearing slash/deadwood pool (`legacyEmissions`) as an additive `+` child of `Resources|Carbon`, so the reported total carbon includes the carbon still held as slash/deadwood under the reporting reframe. This mirrors the `+|Legacy clearing` child on the emission side in `reportEmissions` and keeps emissions and stocks consistent (the reframed Land-use Change flux equals minus the change in this pool). The soil/litter/ vegetation sub-pools stay model-native; only the aggregate changes (~0.5 percent). legacyEmis=FALSE => `Resources|Carbon` is the model's soil+litter+vegetation (backward compatible).

Carbon stock variables

Name	Unit	Meta
Resources|Carbon	Mt C	Total terrestrial carbon stocks (incl. the legacy-clearing pool when legacy
Resources|Carbon|+|Soil	Mt C	Soil carbon stocks
Resources|Carbon|+|Litter	Mt C	Litter carbon stocks
Resources|Carbon|+|Vegetation	Mt C	Vegetation carbon stocks (above and below ground biomass)
Resources|Carbon|+|Legacy clearing pool	Mt C	Slash/deadwood pool of the legacy-clearing reframe (additive child; only

Author(s)

Kristine Karstens, Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportSOM(gdx)

## End(Not run)
```

`reportConsumVal` *reportConsumVal*

Description

reports MAgPIE consumption value

Usage

```
reportConsumVal(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

Magpie object associated with the consumption value

Consumption value variables

Name	Unit	Meta
Value|Consumption Value	million US$2017/yr	Total value of agricultural consumption

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
x <- reportConsumVal(gdx)

## End(Not run)
```

`reportCostCapitalInvestment`
reportCostCapitalInvestment

Description

reports MAgPIE capital investments

Usage

```
reportCostCapitalInvestment(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

Magpie object associated with overall costs and value of production

Capital investment cost variables

Name	Unit	Meta
Costs|Capital Investments	million US$2017	Capital investments (sticky cost implementation)

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
x <- reportCostCapitalInvestment(gdx)

## End(Not run)
```

`reportCostCapitalStocks`
reportCostCapitalStocks

Description

reports MAgPIE capital stocks

Usage

```
reportCostCapitalStocks(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

Magpie object associated with overall costs and value of production

Capital stock variables

Name	Unit	Meta
Capital Stocks|Arable farm capital	million US$2017	Capital stocks used in cropland (sticky cost implementation)

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
x <- reportCostCapitalStocks(gdx)

## End(Not run)
```

`reportCostOverall` *reportCostOverall*

Description

reports MAgPIE costs

Usage

```
reportCostOverall(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

Magpie object associated with overall costs and value of production

Overall cost variables

Name	Unit	Meta
Costs|Gross value of production	million US$2017/yr	Total gross value of agricultural production

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
  x <- reportCostOverall(gdx)

## End(Not run)
```

reportCosts *reportCosts*

Description

reports MAgPIE costs

Usage

```
reportCosts(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via gdxAggregate.

Value

consumption value as MAgPIE object Unit: see names

Cost variables

Name	Unit	Meta
Costs	million US$2017/yr	Total production costs
Costs|+|Input Factors	million US$2017/yr	Costs for input factors (labor, capital, materials)
Costs|+|Land Conversion	million US$2017/yr	Costs for land conversion
Costs|+|Transport	million US$2017/yr	Transport costs
Costs|+|TC	million US$2017/yr	Technology costs (research and development)
Costs|+|GHG Emissions	million US$2017/yr	Costs from GHG emission pricing
Costs|MainSolve w/o GHG Emissions	million US$2017/yr	Total costs excluding GHG emission costs

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
  x <- reportCosts(gdx)

## End(Not run)
```

`reportCostsAccounting` *reportCostsAccounting*

Description

reports MAgPIE costs including total investments

Usage

```
reportCostsAccounting(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

Costs accounting including total investments

Cost accounting variables

Name	Unit	Meta
Costs Accounting	million US$2017/yr	Total cost accounting including investments
Costs Accounting|+|Land Conversion	million US$2017/yr	Investment costs for land conversion
Costs Accounting|+|Transport	million US$2017/yr	Transport cost investments
Costs Accounting|+|TC	million US$2017/yr	Technological change investment costs

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
x <- reportCostsAccounting(gdx)

## End(Not run)
```

reportCostsAEI *reportCostsAEI*

Description

reports MAgPIE AEI costs

Usage

```
reportCostsAEI(gdx)
```

Arguments

gdx	GDX file

Value

magpie object containing AEI costs

AEI cost variables

Name	Unit	Meta
Costs|AEI	million US$2017/yr	Costs for area equipped for irrigation

Author(s)

Felicitas Beier

Examples

```
## Not run:
  x <- reportCostsAEI(gdx)

## End(Not run)
```

`reportCostsFertilizer` *reportCostsFertilizer*

Description

reports MAgPIE nitrogen fertilizer costs disaggregated to crop categories

Usage

```
reportCostsFertilizer(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

magpie object with fertilizer costs

Nitrogen fertilizer cost variables

Name	Unit	Meta
Costs|N Fertilizer	million US$2017/yr	Total nitrogen fertilizer costs
Costs|N Fertilizer|+|Crops	million US$2017/yr	N fertilizer costs for crops
Costs|N Fertilizer|+|Pasture	million US$2017/yr	N fertilizer costs for pasture

Author(s)

Debbora Leip

Examples

```
## Not run:
  x <- reportCostsFertilizer(gdx)

## End(Not run)
```

`reportCostsInputFactors` *reportFactorCosts*

Description

reports MAgPIE factor costs (split into labor and capital for sticky realization)

Usage

```
reportCostsInputFactors(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

magpie object with factor costs

Input factor cost variables

Name	Unit	Meta
Costs|Input Factors	million US$2017/yr	Total input factor costs (labor + capital)
Costs|Input Factors|+|Crops	million US$2017/yr	Factor costs for crop production
Costs|Input Factors|+|Livestock	million US$2017/yr	Factor costs for livestock production
Costs|Input Factors|+|Fish	million US$2017/yr	Factor costs for fish production
Costs|Input Factors|+|Pasture	million US$2017/yr	Factor costs for pasture management
Costs|Input Factors|+|Residues	million US$2017/yr	Factor costs for residue handling
Costs|Input Factors|++|Labor costs	million US$2017/yr	Total labor costs
Costs|Input Factors|++|Capital costs	million US$2017/yr	Total capital costs

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- reportCostsInputFactors(gdx)

## End(Not run)
```

`reportCostsMACCS` *reportCostsMACCS*

Description

reports MAgPIE mitigation costs disaggregated into labor and capital

Usage

```
reportCostsMACCS(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

magpie object with mitigation costs

MACC cost variables

Name	Unit	Meta
Costs|MACCS|+|Labor costs	million US$2017/yr	Labor costs for MACC implementation
Costs|MACCS|+|Capital costs	million US$2017/yr	Capital costs for MACC implementation

Author(s)

Debbora Leip

Examples

```
## Not run:
  x <- reportCostsMACCS(gdx)

## End(Not run)
```

reportCostsPresolve *reportCostsPresolve*

Description

reports MAgPIE costs

Usage

```
reportCostsPresolve(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

consumption value as MAgPIE object Unit: see names

Presolve cost variables

Name	Unit	Meta
Costs|PreSolve|Total	million US$2017	Cumulative costs from presolve phase

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportCostsPresolve(gdx)

## End(Not run)
```

reportCostsTrade *reportCostsTrade*

Description

Reports MAgPIE bilateral trade cost components

Usage

```
reportCostsTrade(gdx, level = "regglo", sum = FALSE)
```

Arguments

gdx	GDX file
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
sum	whether to sum across subcategories

Value

A MAgPIE object containing values related to trade costs (million US$2017/yr)

Author(s)

David M Chen

reportCostsWholesale *reportCostsWholesale*

Description

Reads data to calculate wholesale costs

Usage

```
reportCostsWholesale(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing values related with costs wholesale trade (million US$2017/yr)

Wholesale cost variables

Name	Unit	Meta
Costs|Wholesale Costs	million US$2017/yr	Total wholesale trade costs

Author(s)

David M Chen

Examples

```
## Not run:
x <- reportCostsWholesale(gdx)

## End(Not run)
```

`reportCostsWithoutIncentives`
reportCostsWithoutIncentives

Description

reports Costs Without Incentives

Usage

```
reportCostsWithoutIncentives(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

magpie object

Costs without incentives variables

Name	Unit	Meta
Costs Accounting|Costs without incentives	million US$2017/yr	Total costs excluding policy incentives

Author(s)

David Chen

Examples

```
## Not run:
x <- reportCostsWithoutIncentives(gdx)

## End(Not run)
```

`reportCostTC` *reportCostTC*

Description

reports MAgPIE TC costs

Usage

```
reportCostTC(gdx)
```

Arguments

gdx	GDX file

Value

magpie object with TC costs

TC cost variables

Name	Unit	Meta
Costs|TC	million US$2017/yr	Technological change investment costs

Author(s)

David Chen

Examples

```
## Not run:
x <- reportCostTC(gdx)

## End(Not run)
```

`reportCostTransport` *reportCostTransport*

Description

reports MAgPIE costs

Usage

```
reportCostTransport(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

consumption value as MAgPIE object Unit: see names

Transport cost variables

Name	Unit	Meta
Costs|Transport	million US$2017/yr	Total transport costs for agricultural commodities
Costs|Transport|+|Crops	million US$2017/yr	Transport costs for crop products
Costs|Transport|+|Livestock products	million US$2017/yr	Transport costs for livestock products

Author(s)

David Chen

Examples

```
  ## Not run:
    x <- reportCostTransport(gdx)

## End(Not run)
```

`reportCroparea` *reportCroparea*

Description

reports croparea

Usage

```
reportCroparea(gdx, detail = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`detail`	if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
`level`	aggregation level of returned data ("regglo" by default)

Value

Croparea as MAgPIE object (million ha/yr)

Croparea variables

Name	Unit	Meta
Resources|Land Cover|Cropland|Croparea	million ha	Total physical cropland area used for crop prod
Resources|Land Cover|Cropland|Croparea|+|Crops	million ha	Cropland area for food and feed crops
Resources|Land Cover|Cropland|Croparea|Crops|+|Cereals	million ha	Cropland for cereals (maize, rice, temperate cer
Resources|Land Cover|Cropland|Croparea|Crops|+|Oil crops	million ha	Cropland for oil crops (cotton seed, groundnuts
Resources|Land Cover|Cropland|Croparea|Crops|+|Sugar crops	million ha	Cropland for sugar crops (sugar beet, sugar can
Resources|Land Cover|Cropland|Croparea|Crops|+|Other crops	million ha	Cropland for other crops (fruits, vegetables, nut
Resources|Land Cover|Cropland|Croparea|+|Bioenergy crops	million ha	Cropland for second-generation bioenergy crop
Resources|Land Cover|Cropland|Croparea|++|Irrigated	million ha	Irrigated cropland (physical area)
Resources|Land Cover|Cropland|Croparea|++|Rainfed	million ha	Rainfed cropland (physical area)

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- reportCroparea(gdx)

## End(Not run)
```

`reportCropareaGrid` *reportCropareaGrid*

Description

reports croparea

Usage

```
reportCropareaGrid(gdx)
```

Arguments

gdx	GDX file

Value

Croparea as MAgPIE object (million ha/yr)

Grid-level croparea

This function produces grid-level (0.5 degree) croparea data for individual crops. Variable names follow the reportingnames mapping (e.g., Cereals, Oilcrops, etc.).

Author(s)

Benjamin Bodirsky

Examples

```
## Not run:
x <- reportCropareaGrid(gdx)

## End(Not run)
```

`reportCropDiversity` *reportCropDiversity*

Description

reports crop diversity

Usage

```
reportCropDiversity(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Supports "reg", "glo", "regglo", "grid" or a custom region aggregation level.

Value

Crop diversity as MAgPIE object

Crop diversity variables

Name	Unit	Meta
Biodiversity|Shannon crop area diversity index	unitless	Crop type diversity based on area shares (higher = more d
Biodiversity|Inverted Simpson crop area diversity index	unitless	Crop type diversity based on area shares (higher = more d

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
x <- reportCropDiversity(gdx)

## End(Not run)
```

reportCroplandSCM *reportCroplandSCM*

Description

Reports cropland area under soil carbon management (SCM)

Usage

```
reportCroplandSCM(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level ("regglo" by default)

Value

SCM area indicators as MAgPIE object (million ha)

SCM area variables

Name	Unit	Meta
Resources|Land Cover|Cropland|Croparea under Soil Carbon Management	million ha	Total cropland ar
Resources|Land Cover|Cropland|Croparea under Soil Carbon Management|Share of Croparea	1	Share of croplan

Author(s)

Kristine Karstens

Examples

```
## Not run:
x <- reportCroplandSCM(gdx)

## End(Not run)
```

`reportCropResidues2ndBE` *reportCropResidues2ndBE*

Description

Reports crop residues available for 2nd generation bioenergy from a MAgPIE GDX file at ISO country level. Applies soil cover constraints (minimum 30% soil cover retained, Lutz et al. 2019) and a collection fraction to estimate sustainably harvestable residue biomass, converted to energy (PJ).

Usage

```
reportCropResidues2ndBE(
  gdx,
  file = NULL,
  collectionFraction = 0.25,
  minDensityForExtraction = 0.1
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`collectionFraction`	fraction of available residues that can be physically collected (default: 0.25)
`minDensityForExtraction`	minimum residue density (tDM/ha) below which extraction is not economic (default: 0.1)

Value

MAgPIE object with crop residue potential for 2nd gen bioenergy (res_cereals, res_fibrous, res_nonfibrous) at ISO level in PJ.

Author(s)

Kristine Karstens

References

Lutz et al. (2019), GMD, https://gmd.copernicus.org/articles/12/2419/2019/

See Also

`ResidueBiomass`, `ResidueUsage`

Examples

```
## Not run:
  x <- reportCropResidues2ndBE(gdx)

## End(Not run)
```

`reportDemand` *reportDemand*

Description

reports Demand for Food, Feed, Processing, Material, Bioenergy, Seed and Supply Chain Loss

Usage

```
reportDemand(gdx, detail = FALSE, agmip = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`detail`	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
`agmip`	if agmip=T, additional sector aggregates required for agmip are reported (e.g. "AGR")
`level`	The level at which the report data should be aggregated.

Value

demand as MAgPIE object (Mt DM)

Demand variables

Name	Unit	Meta
Demand	Mt DM/yr	Total demand for agricultural products including food, feed, processing, mate
Demand|++|Crops	Mt DM/yr	Demand for crops including food, feed products and bioenergy (1st and 2nd g
Demand|++|Livestock products	Mt DM/yr	Demand for livestock products (excluding fish)
Demand|++|Secondary products	Mt DM/yr	Demand for secondary products (processed agricultural goods)
Demand|++|Pasture	Mt DM/yr	Demand for pasture biomass
Demand|++|Bioenergy crops	Mt DM/yr	Demand for second-generation bioenergy crops

Demand by use variables

Name	Unit	Meta
Demand|Food	Mt DM/yr	Demand for food consumption
Demand|Feed	Mt DM/yr	Demand for animal feed
Demand|Processing	Mt DM/yr	Demand for food processing
Demand|Material	Mt DM/yr	Demand for material use (non-food, non-feed)
Demand|Bioenergy	Mt DM/yr	Demand for bioenergy production
Demand|Seed	Mt DM/yr	Demand for seeds
Demand|Supply Chain Loss	Mt DM/yr	Losses in the supply chain

Author(s)

Benjamin Leon Bodirsky, Isabelle Weindl

Examples

```
  ## Not run:
    x <- reportDemand()

## End(Not run)
```

`reportDemandBioenergy` *reportDemandBioenergy*

Description

reports Bioenergy Demand in EJ/yr

Usage

```
reportDemandBioenergy(gdx, detail = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`detail`	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

Bioenergy demand as MAgPIE object (EJ/yr)

Bioenergy demand variables

Name	Unit	Meta
Demand|Bioenergy	EJ/yr	Total bioenergy demand
Demand|Bioenergy|++|2nd generation	EJ/yr	Second generation bioenergy demand (dedicated crops and residues)
Demand|Bioenergy|++|1st generation	EJ/yr	First generation bioenergy demand (oils, ethanol)
Demand|Bioenergy|++|Traditional Burning	EJ/yr	Traditional biomass burning demand

Author(s)

Florian Humpenoeder, Kristine Karstens

Examples

```
  ## Not run:
    x <- reportDemandBioenergy()

## End(Not run)
```

---

reportDemandNr *reportDemandNr*

Description

Similar to reportDemand, but for nitrogen. reports Demand for Food, Feed, Processing, Material, Bioenergy, Seed and Supply Chain Loss

Usage

```
reportDemandNr(gdx, detail = FALSE)
```

Arguments

gdx	GDX file
detail	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")

Value

demand as MAgPIE object (Mt DM)

Nitrogen demand variables

Name	Unit	Meta
Demand|Food	Mt Nr/yr	Nitrogen demand for food
Demand|Feed	Mt Nr/yr	Nitrogen demand for feed
Demand|Processing	Mt Nr/yr	Nitrogen demand for processing
Demand|Bioenergy	Mt Nr/yr	Nitrogen demand for bioenergy
Demand|Material	Mt Nr/yr	Nitrogen demand for materials

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportDemand()

## End(Not run)
```

`reportEmissions` *reportEmissions*

Description

reports GHG emissions

Usage

```
reportEmissions(gdx, level = "regglo", storageWood = TRUE, legacyEmis = TRUE)
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.
`storageWood`	Accounting for long term carbon storage in wood products. Default is TRUE.
`legacyEmis`	Logical (default TRUE). If TRUE, fold a legacy-clearing correction into Land-use Change: the instantaneous aboveground clearing pulse (deforestation + other-land conversion) is re-spread over a first-order-decay slash/deadwood pool via

legacyEmissions, mirroring the HWP convolution. "Legacy" here is the Houghton-style delayed release of carbon from land already cleared (a slash/deadwood decay tail) - it is NOT the front-loaded "committed deforestation emissions" of the trade-footprint literature (which assigns the whole future tail to the clearing year). legacyEmis=FALSE => output is bit-identical to the raw instantaneous accounting. This is a REPORTING reframe only (the model still emits instantaneously and the carbon price is unaffected) and is NOT mass-conserving (the pre-1995 priming credits a ~+6 to +21 percent legacy tail); the kernel and priming are literature-anchored and MUST NOT be tuned to a comparison cloud. See legacyEmissions for details and caveats.

Details

Comparability with historical LUC-CO2 sources has two axes (documented in mrvalidation calcValidGlobalCarbonBudget
(1) PEAT - Emissions|CO2|Land|+|Land-use Change includes peat (separable via its +|Peatland child); the bookkeeping sources (BLUE/OSCAR/H&C/GCB/Gasser) are ex-peat, so compare them to Land-use Change net of the Peatland child. (2) INDIRECT (Grassi) - +|Land-use Change is a direct/bookkeeping flux, whereas net Emissions|CO2|Land adds the managed-land environmental sink (+|Indirect) and is the quantity comparable to NGHGI/inventory sources (FAO/PRIMAP/EDGAR). Note the GCB net-Land / Indirect series in mrvalidation is all-land S_LAND, NOT this managed-land quantity, so do not benchmark net Land against it.

Value

GHG emissions as MAgPIE object (Unit: Mt CO2/yr, Mt N2O/yr, and Mt CH4/yr, for cumulative emissions Gt CO2)

CO2 land-use change emissions (yearly)

Name	Unit	Meta
Emissions|CO2|Land	Mt CO2/yr	Net CO2 flux from land use an
Emissions|CO2|Land|+|Indirect	Mt CO2/yr	Carbon sink on managed land
Emissions|CO2|Land|+|Land-use Change	Mt CO2/yr	Net CO2 flux from land-use ch
Emissions|CO2|Land|Land-use Change|+|Deforestation	Mt CO2/yr	CO2 emissions from permaner
Emissions|CO2|Land|Land-use Change|Deforestation|+|Primary forests	Mt CO2/yr	CO2 emissions from deforesta
Emissions|CO2|Land|Land-use Change|Deforestation|+|Secondary forests	Mt CO2/yr	CO2 emissions from deforesta
Emissions|CO2|Land|Land-use Change|Deforestation|+|Forestry plantations	Mt CO2/yr	CO2 emissions from deforesta
Emissions|CO2|Land|Land-use Change|Deforestation|+|Cropland Tree Cover	Mt CO2/yr	CO2 emissions from removal c
Emissions|CO2|Land|Land-use Change|+|Forest degradation	Mt CO2/yr	CO2 emissions from forest deg
Emissions|CO2|Land|Land-use Change|Forest degradation|+|Primary forests	Mt CO2/yr	CO2 emissions from degradati
Emissions|CO2|Land|Land-use Change|Forest degradation|+|Secondary forests	Mt CO2/yr	CO2 emissions from degradati
Emissions|CO2|Land|Land-use Change|+|Other land conversion	Mt CO2/yr	CO2 emissions from conversio
Emissions|CO2|Land|Land-use Change|+|Regrowth	Mt CO2/yr	CO2 removals from forest regr
Emissions|CO2|Land|Land-use Change|Regrowth|+|CO2-price AR	Mt CO2/yr	CO2 removals from afforestati
Emissions|CO2|Land|Land-use Change|Regrowth|+|NPI_NDC AR	Mt CO2/yr	CO2 removals from afforestati
Emissions|CO2|Land|Land-use Change|Regrowth|+|Timber Plantations	Mt CO2/yr	CO2 removals from timber pla
Emissions|CO2|Land|Land-use Change|Regrowth|+|Secondary Forest	Mt CO2/yr	CO2 removals from secondary
Emissions|CO2|Land|Land-use Change|Regrowth|+|Cropland Tree Cover	Mt CO2/yr	CO2 removals from trees on cr

Emissions|CO2|Land|Land-use Change|Regrowth|+|Other Land	Mt CO2/yr	CO2 removals from regrowth o
Emissions|CO2|Land|Land-use Change|+|Peatland	Mt CO2/yr	Net CO2 flux from managed pe
Emissions|CO2|Land|Land-use Change|Peatland|+|Positive	Mt CO2/yr	CO2 emissions from drained p
Emissions|CO2|Land|Land-use Change|Peatland|+|Negative	Mt CO2/yr	CO2 removals from rewetted p
Emissions|CO2|Land|Land-use Change|Excl Peatland	Mt CO2/yr	Net CO2 flux from land-use ch
Emissions|CO2|Land|Land-use Change|+|Soil	Mt CO2/yr	Net CO2 flux from soil organic
Emissions|CO2|Land|Land-use Change|Soil|++|Emissions	Mt CO2/yr	CO2 emissions from soil carbo
Emissions|CO2|Land|Land-use Change|Soil|++|Withdrawals	Mt CO2/yr	CO2 removals from soil carbon
Emissions|CO2|Land|Land-use Change|Soil|+|Land Conversion	Mt CO2/yr	Net soil CO2 flux from land-us
Emissions|CO2|Land|Land-use Change|Soil|+|Cropland management	Mt CO2/yr	Net soil CO2 flux from croplan
Emissions|CO2|Land|Land-use Change|Soil|+|Soil Carbon Management	Mt CO2/yr	Net soil CO2 flux from explici
Emissions|CO2|Land|Land-use Change|+|Wood Harvest	Mt CO2/yr	CO2 emissions from wood har
Emissions|CO2|Land|Land-use Change|Wood Harvest|+|Timber Plantations	Mt CO2/yr	CO2 emissions from harvest in
Emissions|CO2|Land|Land-use Change|Wood Harvest|+|Primary Forest	Mt CO2/yr	CO2 emissions from harvest in
Emissions|CO2|Land|Land-use Change|Wood Harvest|+|Secondary Forest	Mt CO2/yr	CO2 emissions from harvest in
Emissions|CO2|Land|Land-use Change|Wood Harvest|+|Other Land	Mt CO2/yr	CO2 emissions from harvest on
Emissions|CO2|Land|Land-use Change|+|Timber	Mt CO2/yr	Net CO2 flux from harvested w
Emissions|CO2|Land|Land-use Change|Timber|+|Storage in HWP	Mt CO2/yr	CO2 stored in harvested wood
Emissions|CO2|Land|Land-use Change|Timber|+|Release from HWP	Mt CO2/yr	CO2 released from decay of ha
Emissions|CO2|Land|Land-use Change|+|Legacy clearing	Mt CO2/yr	REPORTING REFRAME: net
Emissions|CO2|Land|Land-use Change|Legacy clearing|+|Storage	Mt CO2/yr	Clearing carbon deferred into t
Emissions|CO2|Land|Land-use Change|Legacy clearing|+|Release	Mt CO2/yr	Clearing carbon released from
Emissions|CO2|Land|Land-use Change|Excl Legacy clearing	Mt CO2/yr	Net CO2 flux from land-use ch
Emissions|CO2|Land|Land-use Change|Gross Positive	Mt CO2/yr	Gross source of the LUC-CO2
Emissions|CO2|Land|Land-use Change|Gross Negative	Mt CO2/yr	Gross sink of the LUC-CO2 flu
Emissions|CO2|Land|Land-use Change|+|Residual	Mt CO2/yr	Residual CO2 flux not capture
Emissions|CO2|Land|Land-use Change|Residual|+|Positive	Mt CO2/yr	Positive residual CO2 flux
Emissions|CO2|Land|Land-use Change|Residual|+|Negative	Mt CO2/yr	Negative residual CO2 flux
Emissions|CO2|Land|++|Above Ground Carbon	Mt CO2/yr	CO2 flux from above ground c
Emissions|CO2|Land|++|Below Ground Carbon	Mt CO2/yr	CO2 flux from below ground c

CO2 land carbon sink (yearly)

Name	Unit	Meta
Emissions|CO2|Land Carbon Sink|Grassi|Managed Land|Managed Forest	Mt CO2/yr	Carbon sink in managed forests foll
Emissions|CO2|Land Carbon Sink|LPJmL	Mt CO2/yr	Total carbon sink from LPJmL vege
Emissions|CO2|Land Carbon Sink|LPJmL|+|Managed Land	Mt CO2/yr	Carbon sink on managed land from
Emissions|CO2|Land Carbon Sink|LPJmL|+|Unmanaged Land	Mt CO2/yr	Carbon sink on unmanaged land fro

CO2 land-use change emissions (cumulative)

Name	Unit	Meta
Emissions|CO2|Land|Cumulative	Gt CO2	Cumulative net CO2 flux from land u
Emissions|CO2|Land|Cumulative|+|Indirect	Gt CO2	Cumulative carbon sink on managed
Emissions|CO2|Land|Cumulative|+|Land-use Change	Gt CO2	Cumulative net CO2 flux from land-u
Emissions|CO2|Land|Cumulative|Land-use Change|+|Deforestation	Gt CO2	Cumulative CO2 emissions from def
Emissions|CO2|Land|Cumulative|Land-use Change|+|Regrowth	Gt CO2	Cumulative CO2 removals from regr
Emissions|CO2|Land|Cumulative|Land-use Change|+|Other land conversion	Gt CO2	Cumulative CO2 emissions from oth
Emissions|CO2|Land|Cumulative|Land-use Change|+|Peatland	Gt CO2	Cumulative net CO2 flux from peatla
Emissions|CO2|Land|Cumulative|Land-use Change|Excl Peatland	Gt CO2	Cumulative net CO2 flux from land-u
Emissions|CO2|Land|Cumulative|Land-use Change|+|Soil	Gt CO2	Cumulative net CO2 flux from soil
Emissions|CO2|Land|Cumulative|Land-use Change|+|Wood Harvest	Gt CO2	Cumulative CO2 emissions from wo
Emissions|CO2|Land|Cumulative|Land-use Change|+|Timber	Gt CO2	Cumulative net CO2 flux from harve
Emissions|CO2|Land|Cumulative|Land-use Change|+|Legacy clearing	Gt CO2	Cumulative legacy clearing correctio
Emissions|CO2|Land|Cumulative|Land-use Change|Excl Legacy clearing	Gt CO2	Cumulative net CO2 flux from land-u
Emissions|CO2|Land|Cumulative|Land-use Change|+|Residual	Gt CO2	Cumulative residual CO2 flux
Emissions|CO2|Land|Cumulative|Land-use Change|Gross Positive	Gt CO2	Cumulative gross source of the LUC
Emissions|CO2|Land|Cumulative|Land-use Change|Gross Negative	Gt CO2	Cumulative gross sink of the LUC-C

N2O emissions variables

Name	Unit	Meta
Emissions|N2O|Land	Mt N2O/yr	Total N2O emissions fro
Emissions|N2O|Land|+|Agriculture	Mt N2O/yr	N2O emissions from the
Emissions|N2O|Land|Agriculture|+|Animal Waste Management	Mt N2O/yr	N2O emissions from ani
Emissions|N2O|Land|Agriculture|+|Agricultural Soils	Mt N2O/yr	N2O emissions from agri
Emissions|N2O|Land|Agriculture|Agricultural Soils|+|Inorganic Fertilizers	Mt N2O/yr	N2O emissions from ino
Emissions|N2O|Land|Agriculture|Agricultural Soils|Inorganic Fertilizers|+|Cropland	Mt N2O/yr	N2O emissions from fert
Emissions|N2O|Land|Agriculture|Agricultural Soils|Inorganic Fertilizers|+|Pasture	Mt N2O/yr	N2O emissions from fert
Emissions|N2O|Land|Agriculture|Agricultural Soils|+|Manure applied to Croplands	Mt N2O/yr	N2O emissions from ma
Emissions|N2O|Land|Agriculture|Agricultural Soils|+|Decay of Crop Residues	Mt N2O/yr	N2O emissions from dec
Emissions|N2O|Land|Agriculture|Agricultural Soils|+|Soil Organic Matter Loss	Mt N2O/yr	N2O emissions from soil
Emissions|N2O|Land|Agriculture|Agricultural Soils|+|Pasture	Mt N2O/yr	N2O emissions from pas
Emissions|N2O|Land|+|Peatland	Mt N2O/yr	N2O emissions from ma
Emissions|N2O|Land|Peatland|+|Managed	Mt N2O/yr	N2O emissions from ma
Emissions|N2O|Land|+|Biomass Burning	Mt N2O/yr	N2O emissions from bio
Emissions|N2O|Land|Biomass Burning|+|Burning of Crop Residues	Mt N2O/yr	N2O emissions from bur

CH4 emissions variables

Name	Unit	Meta
Emissions|CH4|Land	Mt CH4/yr	Total CH4 emissions from agriculture, fo

Emissions|CH4|Land|+|Agriculture	Mt CH4/yr	CH4 emissions from agriculture sector
Emissions|CH4|Land|Agriculture|+|Rice	Mt CH4/yr	CH4 emissions from flooded rice cultivati
Emissions|CH4|Land|Agriculture|+|Animal waste management	Mt CH4/yr	CH4 emissions from animal waste manag
Emissions|CH4|Land|Agriculture|+|Enteric fermentation	Mt CH4/yr	CH4 emissions from enteric fermentation
Emissions|CH4|Land|+|Peatland	Mt CH4/yr	CH4 emissions from managed peatlands
Emissions|CH4|Land|Peatland|+|Managed	Mt CH4/yr	CH4 emissions from managed peatlands (
Emissions|CH4|Land|+|Biomass Burning	Mt CH4/yr	CH4 emissions from biomass burning
Emissions|CH4|Land|Biomass Burning|+|Burning of Crop Residues	Mt CH4/yr	CH4 emissions from burning of crop resi

GWP emissions variables

Name	Unit	Meta
Emissions|GWP100AR6|Land	Gt CO2e/yr	Total GHG emissions from land use in CO2-equivalents using AR
Emissions|GWP100AR6|Land|Cumulative	Gt CO2e	Cumulative total GHG emissions from land use
Emissions|CH4_GWP100AR6|Land	Mt CO2e/yr	CH4 emissions in CO2-equivalents using AR6 GWP100 (factor 2
Emissions|N2O_GWP100AR6|Land	Mt CO2e/yr	N2O emissions in CO2-equivalents using AR6 GWP100 (factor 2

Author(s)

Florian Humpenoeder, Benjamin Leon Bodirsky, Michael Crawford

Examples

```
## Not run:
x <- reportEmissions(gdx)

## End(Not run)
```

`reportEmissionsBeforeTechnicalMitigation`

reportEmissionsBeforeTechnicalMitigation

Description

reports GHG emissions before technical mitigation. Technical abatement includes all abatement done in the MACC curves, but exclude endogenous mitigation. These emissions are NOT the standard reporting emissions, but used for special purposes like remind-magpie coupling.

Usage

```
reportEmissionsBeforeTechnicalMitigation(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

MAgPIE object (Unit: Mt CO2/yr, Mt N2O/yr and Mt CH4/yr)

Emissions before technical mitigation variables

Name	Unit	Meta
Emissions before technical mitigation|N2O|Land|+|Agriculture	Mt N2O/yr	N2O emissions be
Emissions before technical mitigation|N2O|Land|Agriculture|+|Animal Waste Management	Mt N2O/yr	N2O from animal
Emissions before technical mitigation|N2O|Land|Agriculture|+|Agricultural Soils	Mt N2O/yr	N2O from agricul
Emissions before technical mitigation|CH4|Land|+|Agriculture	Mt CH4/yr	CH4 emissions be
Emissions before technical mitigation|CH4|Land|Agriculture|+|Enteric fermentation	Mt CH4/yr	CH4 from enteric

Author(s)

Florian Humpenoeder, Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportEmissionsBeforeTechnicalMitigation(gdx)

## End(Not run)
```

---

reportExpenditureFoodIndex

reportExpenditureFoodIndex

Description

reports food expenditure index and food expenditure index corrected for emission costs

Usage

```
reportExpenditureFoodIndex(
  gdx,
  baseyear = "y2010",
  basketyear = "y2010",
  level = "regglo"
)
```

Arguments

gdx	GDX file
baseyear	Baseyear of the price index
basketyear	Year of reference food basket (should be in the past for comparison of different runs to have identical and comparable food basket)
level	aggregation level of returned data ("regglo" by default)

Value

Food expenditure index as MAgPIE object

Food expenditure index variables

Name	Unit	Meta
Prices|Food Expenditure Index	Index 2010=100	Food expenditure index (all food products)
Prices|Food Expenditure Index|Plant-based food products	Index 2010=100	Food expenditure index for plant-based produc
Prices|Food Expenditure Index|Livestock food products	Index 2010=100	Food expenditure index for livestock products
Prices|Food Expenditure Index corrected for ghg costs	Index 2010=100	Food expenditure index corrected for GHG cos
Prices|Agricultural Primary Products Expenditure Index	Index 2010=100	Agricultural primary products expenditure ind

Author(s)

Felicitas Beier

Examples

```
## Not run:
x <- reportPriceFoodIndex(gdx)

## End(Not run)
```

reportExtraResidueEmissions *reportExtraResidueEmissions.R*

Description

reads residue biomass from a MAgPIE gdx file and multiplies DM by emission factors from GFED

Usage

```
reportExtraResidueEmissions(gdx, level = "reg")
```

Arguments

gdx	GDX file
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global)

Value

emissions as MAgPIE object (unit: Mt X/yr, plus cumulative Mt X/yr)

Residue burning emission variables

Name	Unit	Meta
Emissions|CO2|Land|Biomass Burning|+|Burning of Crop Residues	Mt CO2/yr	CO2 emissions from residue
Emissions|CO|Land|Biomass Burning|+|Burning of Crop Residues	Mt CO/yr	CO emissions from residue b
Emissions|BC|Land|Biomass Burning|+|Burning of Crop Residues	Mt BC/yr	Black carbon from residue bu
Emissions|VOC|Land|Biomass Burning|+|Burning of Crop Residues	Mt VOC/yr	VOC emissions from residue
Emissions|CO2|Land|Cumulative|Biomass Burning|+|Burning of Crop Residues	Mt CO2	Cumulative CO2 from residu

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- reportExtraResidueEmissions(gdx, level = "glo")

## End(Not run)
```

reportFactorCostShares

reportFactorCostShares

Description

reports labor and capital cost share out of factor costs from MAgPIE results

Usage

```
reportFactorCostShares(gdx, type = "optimization", level = "regglo")
```

Arguments

gdx	GDX file
type	• "requirements": shares from factor requirements • "optimization": cost shares between labor and capital costs in optimization • "accounting": cost shares based on accounting of labor and capital costs
level	spatial aggregation: "reg", "glo", "regglo"

Value

labor and capital cost shares as MAgPIE object

Factor cost share variables

Name	Unit	Meta
Factor cost shares optimization|Crop products|+|Labor cost share	%	Labor cost share in crop production
Factor cost shares optimization|Crop products|+|Capital cost share	%	Capital cost share in crop production
Factor cost shares optimization|Livestock products|+|Labor cost share	%	Labor cost share in livestock production
Factor cost shares optimization|Livestock products|+|Capital cost share	%	Capital cost share in livestock production

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- reportFactorCostShares(gdx)

## End(Not run)
```

reportFeed *reportFeed*

Description

reportes feed demand by animal type

Usage

```
reportFeed(gdx, detail = TRUE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

feed demand as MAgPIE object (Mt DM)

Feed demand variables

Name	Unit	Meta
Demand|Feed|++|Feed for Ruminant meat	Mt DM/yr	Feed demand for ruminant meat production
Demand|Feed|++|Feed for Dairy	Mt DM/yr	Feed demand for dairy production
Demand|Feed|++|Feed for Pig meat	Mt DM/yr	Feed demand for pig meat production
Demand|Feed|++|Feed for Poultry meat	Mt DM/yr	Feed demand for poultry meat production
Demand|Feed|++|Feed for Eggs	Mt DM/yr	Feed demand for egg production
Demand|Feed|++|Feed for Aquaculture	Mt DM/yr	Feed demand for aquaculture

Author(s)

Isabelle Weindl

Examples

```
  ## Not run:
    x <- reportFeed()

## End(Not run)
```

`reportFeedConversion` *reportFeedConversion*

Description

reportes feed demand by animal type

Usage

```
reportFeedConversion(
  gdx,
  livestockSystem = TRUE,
  balanceflow = FALSE,
  level = "regglo"
)
```

Arguments

`gdx`	GDX file
`livestockSystem`	if TRUE, ruminant products and poultry products are aggregated
`balanceflow`	If true, feed includes the calibration balanceflow
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

feed demand as MAgPIE object (Mt DM)

Feed conversion variables

Name	Unit	Meta
Productivity|Feed conversion	GE per GE	Overall feed conversion efficiency (gross energy)
Productivity|Feed conversion|Ruminant meat and dairy	GE per GE	Feed conversion for ruminant products
Productivity|Feed conversion|Poultry meat and eggs	GE per GE	Feed conversion for poultry products
Productivity|Feed conversion|Pig meat	GE per GE	Feed conversion for pig meat
Productivity|Roughage share|Ruminant meat and dairy	GE per GE	Share of roughage in ruminant feed
Productivity|Pasture share|Ruminant meat and dairy	GE per GE	Share of pasture in ruminant feed

Author(s)

Benjamin Bodirsky

Examples

```
  ## Not run:
    x <- reportFeed()

## End(Not run)
```

`reportFertilizerNitrogen` *reportFertilizerNitrogen*

Description

Reports inorganic nitrogen application on crops

Usage

```
reportFertilizerNitrogen(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	level of output

Fertilizer nitrogen variables

Name	Unit	Meta
Resources|Nitrogen|Inorganic Fertilizer Application	Mt Nr/yr	Total inorganic nitrogen fertilizer application
Resources|Nitrogen|Inorganic Fertilizer Application|+|Cereals	Mt Nr/yr	Nitrogen fertilizer application on cereals
Resources|Nitrogen|Inorganic Fertilizer Application|+|Oilcrops	Mt Nr/yr	Nitrogen fertilizer application on oilcrops

Author(s)

David M Chen

See Also

`NitrogenBudget`

Examples

```
## Not run:
x <- reportFertilizerNitrogen(gdx)

## End(Not run)
```

`reportFireEmissions` *reportFireEmissions.R*

Description

reads land cover data and deforestation data to produce future estimates of fire emissions based on extrapolating GFED data from 2003-2016

Usage

```
reportFireEmissions(gdx, level = "reg")
```

Arguments

`gdx`	GDX file
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global)

Value

emissions as MAgPIE object (unit: Mt X/yr, plus cumulative Mt X/yr)

Fire emission variables

Name	Unit	Meta
Emissions|CO2|AFOLU|Land|Fires	Mt CO2/yr	Total CO2 emissions from fires
Emissions|CO2|AFOLU|Land|Fires|+|Forest Burning	Mt CO2/yr	CO2 from forest fires
Emissions|CO2|AFOLU|Land|Fires|Forest Burning|+|Boreal Forest fires	Mt CO2/yr	CO2 from boreal
Emissions|CO2|AFOLU|Land|Fires|Forest Burning|+|Tropical Forest deforestation fires	Mt CO2/yr	CO2 from tropical
Emissions|CO2|AFOLU|Land|Fires|+|Grassland Burning fires	Mt CO2/yr	CO2 from grassland/savanna
Emissions|CO2|AFOLU|Land|Fires|+|Peat Burning	Mt CO2/yr	CO2 from peat fires

Author(s)

Michael Crawford

Examples

```
  ## Not run:
    x <- reportFireEmissions(gdx, level = "glo")

## End(Not run)
```

`reportFit` *reportFit*

Description

reports fit and error indicators compared to initial values

Usage

```
reportFit(gdx, type = "MAPE", level = "cell")
```

Arguments

`gdx`	GDX file
`type`	type of indicator. Options: R2, MAE, MPE (mean percentage error - bias), MAPE (mean absolute percentage error)
`level`	level at which the regional and global bias should be reported. Options "cell" or "grid"

Value

Selected error indicator

Cluster-level fit indicator variables

Name	Unit	Meta
Fit|Cluster|Land Cover	MAPE/MAE/MPE	Fit indicator for overall land cover
Fit|Cluster|Land Cover|Cropland	MAPE/MAE/MPE	Fit indicator for cropland
Fit|Cluster|Land Cover|Pastures and Rangelands	MAPE/MAE/MPE	Fit indicator for pasture (when no rang
Fit|Cluster|Land Cover|Managed pastures	MAPE/MAE/MPE	Fit indicator for managed pastures (wh
Fit|Cluster|Land Cover|Rangelands	MAPE/MAE/MPE	Fit indicator for rangelands (when rang
Fit|Cluster|Land Cover|Urban Area	MAPE/MAE/MPE	Fit indicator for urban area
Fit|Cluster|Land Cover|Other Land	MAPE/MAE/MPE	Fit indicator for other land
Fit|Cluster|Land Cover|Forest|Natural Forest|Primary Forest	MAPE/MAE/MPE	Fit indicator for primary forest
Fit|Cluster|Land Cover|Forest|Natural Forest|Secondary Forest	MAPE/MAE/MPE	Fit indicator for secondary forest

Fit|Cluster|Land Cover|Forest|Planted Forest	MAPE/MAE/MPE	Fit indicator for planted forest
Fit|Cluster|Land Cover|Cropland|{crop}	MAPE/MAE/MPE	Fit indicator for individual crop types

Grid-level fit indicator variables

Name	Unit	Meta
Fit|Grid|Land Cover	MAPE/MAE/MPE	Grid-level fit indicator for overall land co
Fit|Grid|Land Cover|Cropland	MAPE/MAE/MPE	Grid-level fit indicator for cropland
Fit|Grid|Land Cover|Pastures and Rangelands	MAPE/MAE/MPE	Grid-level fit indicator for pasture
Fit|Grid|Land Cover|Managed pastures	MAPE/MAE/MPE	Grid-level fit indicator for managed pastu
Fit|Grid|Land Cover|Rangelands	MAPE/MAE/MPE	Grid-level fit indicator for rangelands (wh
Fit|Grid|Land Cover|Urban Area	MAPE/MAE/MPE	Grid-level fit indicator for urban area
Fit|Grid|Land Cover|Other Land	MAPE/MAE/MPE	Grid-level fit indicator for other land
Fit|Grid|Land Cover|Forest|Natural Forest|Primary Forest	MAPE/MAE/MPE	Grid-level fit indicator for primary forest
Fit|Grid|Land Cover|Forest|Natural Forest|Secondary Forest	MAPE/MAE/MPE	Grid-level fit indicator for secondary fores
Fit|Grid|Land Cover|Forest|Planted Forest	MAPE/MAE/MPE	Grid-level fit indicator for planted forest

Author(s)

Edna Molina Bacca, Patrick v. Jeetze

Examples

```
  ## Not run:
    x <- reportFit(gdx,type)

## End(Not run)
```

`reportFoodExpenditure` *reportFoodExpenditure*

Description

reports per-capita calories food supply (including household waste)

Usage

```
reportFoodExpenditure(gdx, detail = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
level	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

per-capita calories as MAgPIE object (kcal/cap/day)

Food expenditure variables

Name	Unit	Meta
Household Expenditure|Agricultural Primary Products|Expenditure	US$2017/capita	Per-capita expenditure on a
Household Expenditure|Agricultural Primary Products|Expenditure Share	US$2017/US$2017	Share of income spent on a
Household Expenditure|Food|Expenditure	US$2017/capita	Per-capita food expenditure
Household Expenditure|Food|Expenditure Share	US$2017/capita	Share of income spent on fo

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportFoodExpenditure(gdx)

## End(Not run)
```

---

reportFootprints *reportFootprints*

Description

Reports the consumption-based (embodied) resource footprints of MAgPIE results in the standard reporting-variable hierarchy for shiny/mip. Built on `footprints` / `embodiedResourceKastner` (the column-normalised Kastner allocation), so each variable is the amount of the resource embodied in the products a region CONSUMES (production footprint of the goods consumed there, including via imports). Three metrics are reported under a common tree, `Footprints|<metric>|<Resource>|...`:

- **Total** — absolute footprint (Mha, km3, Mt CO2eq, mio people);

- **Per-Capita** — footprint divided by population;
- **Per-Tonne** — footprint per tonne of the final product consumed in each pathway.

For *Total* and *Per-Capita* the product tree (Crops -> Cereals/Oil crops/..., Livestock products, Pasture, Bioenergy crops, Forage) is reported with + summation markers and the prim/secd/feed pathway split (Primary/Secondary/Livestock end use) with ++ markers; both partitions sum to the resource total. Per-Capita stays additive because every product and pathway shares the same population denominator (and the global value is the global footprint divided by global population, not the sum of regional per-capita values). *Per-Tonne* is NOT additive — each pathway has its own tonnes denominator (primary demand / secondary (ksd) demand / livestock (kli) demand) — so it is reported only at the pathway level as flat variables with NO summation markers and no grand total.

Under `Total` only, two further sibling variables are reported alongside the (consumption-based) grand total, with the same product/pathway breakdown:

- **Total||Production** — the footprint embodied in what the region PRODUCES, regardless of who consumes it;
- **Total||Net Trade** — consumption minus production, i.e. the footprint displaced by trade (positive = net exporter of the footprint, negative = net importer). Globally this sums to ~0.

These do not carry a summation symbol relative to the (consumption) grand total, so they are reported for information without being double-counted into it.

NB requires a BILATERAL MAgPIE run (bilateral trade in the GDX). It is called from `getReport` once per resource (so each is a right-sized worker in the parallel report pool rather than one worker holding all four). On standard runs the bilateral trade matrix (`ov21_trade` with an exporter-importer dimension) is absent, so the function emits a message and returns `NULL`; the reporting wrapper then simply omits the footprint variables.

Usage

```
reportFootprints(
  gdx,
  level = "regglo",
  resources = c("land", "emissions", "water", "labor"),
  reassignLivestock = TRUE,
  secdToFeed = TRUE
)
```

Arguments

`gdx`	GDX file (bilateral MAgPIE run).
`level`	spatial aggregation: "reg", "glo" or "regglo" (default). The footprint is computed at "reg"; for the global total the absolute footprint and each denominator are summed over regions and only then divided, so the per-capita/per-tonne globals are correct intensive quantities.
`resources`	character vector of resources to report; any subset of `c("land", "emissions", "water", "labor")`.
`reassignLivestock`	logical; if TRUE (default) the Livestock pathway carries the full livestock footprint (kli products' own footprint moved into feed). See `reassignLivestockPathway`.

`secdToFeed` logical; if TRUE (default) route each crop's processed-then-fed share from secd to feed (and drop the fed-secondary share from the per-tonne secd denominator). See `embodiedResourceKastner`.

Value

consumption footprints as a MAgPIE object with reporting names, or `NULL` (with a message) if the GDX is not a bilateral trade run.

Footprint variables (Land shown; analogous for Emissions/Water/Labor)

Name	Unit	Meta
Footprints|Total|Land	million ha	Total consumption-based land footprint
Footprints|Total|Land|+|Crops	million ha	Land embodied in crops consumed
Footprints|Total|Land|++|Livestock	million ha	Footprint consumed via livestock (feed + kli own)
Footprints|Total|Land|Production	million ha	Land footprint embodied in production
Footprints|Total|Land|Production|+|Crops	million ha	Production footprint, by crop
Footprints|Total|Land|Net Trade	million ha	Consumption minus production (net footprint import)
Footprints|Total|Land|Net Trade|+|Crops	million ha	Net-trade footprint, by crop
Footprints|Per-Capita|Land	ha / capita	Land footprint per capita (additive)
Footprints|Per-Tonne|Land|Primary	ha / t	Land per tonne of primary product eaten directly
Footprints|Per-Tonne|Land|Secondary	ha / t	Land per tonne of secondary (processed) product consumed
Footprints|Per-Tonne|Land|Livestock	ha / t	Land per tonne of livestock product consumed

Author(s)

David M Chen

See Also

`footprints`, `embodiedResourceKastner`

Examples

```
## Not run:
  x <- reportFootprints(gdx)

## End(Not run)
```

reportForestYield *reportForestYield*

Description

reports MAgPIE harvested area for timber.

Usage

```
reportForestYield(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

Yield from Forests for timber production

Forest yield variables

Name	Unit	Meta
Timber Yields|Harvest|Forestry	m3 per ha	Timber yield from plantation forestry
Timber Yields|Harvest|Primary forest	m3 per ha	Timber yield from primary forest harvesting
Timber Yields|Harvest|Secondary forest	m3 per ha	Timber yield from secondary forest harvesting

Author(s)

Abhijeet Mishra, Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportForestYield(gdx)

## End(Not run)
```

reportGridCroparea *reportGridCroparea*

Description

reports Croparea from gridded (disaggregated) output

Usage

```
reportGridCroparea(gdx)
```

Arguments

gdx	GDX file

Value

area of cropland as MAgPIE object (million ha)

Grid-level croparea by irrigation

This function produces grid-level (0.5 degree) croparea data by crop type and irrigation system. Variable names follow the reportingnames mapping (e.g., Cereals.Rainfed, Oilcrops.Irrigated).

Author(s)

Jannes Breier

Examples

```
  ## Not run:
    x <- reportGridCroparea(gdx)

## End(Not run)
```

reportGridLand *reportGridLand*

Description

reports land-use from gridded (disaggregated) output

Usage

```
reportGridLand(gdx)
```

Arguments

gdx	GDX file

Value

land-use as MAgPIE object (million ha)

Grid-level land use

This function produces grid-level (0.5 degree) land use data for land cover categories (cropland, pasture, forest, urban, other land). Variable names follow the reportingnames mapping.

Author(s)

Jannes Breier

Examples

```
  ## Not run:
    x <- reportGridLand(gdx)

## End(Not run)
```

reportGridManureExcretion *reportGridManureExcretion*

Description

reports Manure with reprting names on grid level.

Usage

```
reportGridManureExcretion(gdx)
```

Arguments

gdx	GDX file

Value

MAgPIE object

Grid-level manure excretion

This function produces grid-level (0.5 degree) manure excretion and management data. Includes total manure, breakdown by AWMS and livestock type, and losses from confinement.

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
x <- reportGridManureExcretion(gdx)

## End(Not run)
```

`reportGrowingStock` *reportGrowingStock*

Description

reports Growing stocks for woody materials

Usage

```
reportGrowingStock(
  gdx,
  indicator = "relative",
  detail = FALSE,
  level = "regglo"
)
```

Arguments

`gdx`	GDX file
`indicator`	If the reported numbers are relative (mio m3/ha) or absolute (mio. m3). Default is relative.
`detail`	if detail=FALSE, the subcategories of groups are not reported.
`level`	aggregation level of returned data ("regglo" by default)

Value

production as MAgPIE object. Unit: see names

Growing stock variables

Name	Unit	Meta
Resources|Growing Stock|relative|Forest	m3/ha	Relative growing stock in forests
Resources|Growing Stock|relative|Plantations	m3/ha	Relative growing stock in plantations
Resources|Growing Stock|absolute|Forest	Mm3	Absolute growing stock in forests
Resources|Growing Stock|absolute|Plantations	Mm3	Absolute growing stock in plantations

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- reportGrowingStock(gdx)

## End(Not run)
```

`reportharvested_area_timber` *reportharvested_area_timber*

Description

reports MAgPIE harvested area for timber.

Usage

```
reportharvested_area_timber(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

Area harvested for timber production

Timber harvested area variables

Name	Unit	Meta
Resources|Timber operations|Harvested area for timber|Forestry	Mha per yr	Area harvested from managed forest
Resources|Timber operations|Harvested area for timber|Primary forest	Mha per yr	Area harvested from primary forests
Resources|Timber operations|Harvested area for timber|Secondary forest	Mha per yr	Area harvested from secondary fores

Author(s)

Abhijeet Mishra

Examples

```
## Not run:
  x <- reportharvested_area_timber(gdx)

## End(Not run)
```

`reportHourlyLaborCosts`

reportHourlyLaborCosts

Description

reports hourly labor costs in agriculture from MAgPIE results

Usage

```
reportHourlyLaborCosts(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	spatial aggregation: "reg", "glo", "regglo"

Value

hourly labor costs as MAgPIE object

Hourly labor cost variables

Name	Unit	Meta
Labor|Wages|Hourly labor costs	US$2017/h	Hourly labor costs in agriculture

Author(s)

Debbora Leip

Examples

```
## Not run:
  x <- reportHourlyLaborCosts(gdx)

## End(Not run)
```

reportHunger *reportHunger*

Description

Calculates the share of people living in hunger.

Usage

```
reportHunger(gdx)
```

Arguments

gdx	GDX file

Value

magpie object with hunger (mio people) or hunger share

Hunger variables

Name	Unit	Meta
Food Supply|Calorie Supply|Undernourished	Mio People	Number of undernourished people
Food Supply|Calorie Supply|Share of population undernourished	People/People	Share of population undernourished

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportHunger(gdx)

## End(Not run)
```

reportIncome *reportIncome*

Description

reports income

Usage

```
reportIncome(gdx, type = "ppp", level = "regglo")
```

Arguments

gdx	GDX file
type	ppp for purchase power parity, mer for market exchange rate
level	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

Annual per capita and total income as MAgPIE object (US$2017 MER/cap/yr and million US$17 PPP/yr)

Income variables

Name	Unit	Meta
Income per capita PPP	US$2017 PPP/cap/yr	GDP per capita converted to US$2017 using purchasing power parity (
Income PPP	million US$2017 PPP/yr	GDP converted to US$2017 using purchasing power parity (PPP)
Income per capita MER	US$2017 MER/cap/yr	GDP per capita converted to US$2017 at market exchange rate (MER)
Income MER	million US$2017 MER/yr	GDP converted to US$2017 at market exchange rate (MER)

Author(s)

Florian Humpenoeder, Isabelle Weindl, Felicitas Beier

Examples

```
## Not run:
x <- reportIncome(gdx)

## End(Not run)
```

reportIntakeDetailed *reportIntakeDetailed*

Description

reports detailed or aggregated per-capita kcal intake including exogenous scenarios

Usage

```
reportIntakeDetailed(gdx, detail = TRUE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
level	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

per-capita calorie intake as MAgPIE object (kcal/cap/day)

Calorie intake variables

Name	Unit	Meta
Nutrition|Calorie Intake	kcal/capita/day	Total per-capita calorie intake (excluding waste)
Nutrition|Calorie Intake|+|Crops	kcal/capita/day	Calorie intake from crops
Nutrition|Calorie Intake|+|Livestock products	kcal/capita/day	Calorie intake from livestock products
Nutrition|Calorie Intake|+|Secondary products	kcal/capita/day	Calorie intake from secondary/processed products

Author(s)

Isabelle Weindl

Examples

```
## Not run:
  x <- reportIntakeDetailed(gdx)

## End(Not run)
```

reportKcal *reportKcal*

Description

reports per-capita calories food supply (including household waste)

Usage

```
reportKcal(gdx, detail = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
level	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

per-capita calories as MAgPIE object (kcal/cap/day)

Calorie supply variables

Name	Unit	Meta
Nutrition|Calorie Supply	kcal/capita/day	Total per-capita calorie supply (including household waste
Nutrition|Calorie Supply|+|Crops	kcal/capita/day	Calorie supply from crops
Nutrition|Calorie Supply|+|Livestock products	kcal/capita/day	Calorie supply from livestock products
Nutrition|Calorie Supply|+|Secondary products	kcal/capita/day	Calorie supply from secondary/processed products

Author(s)

Benjamin Leon Bodirsky, Kristine karstens, Abhijeet Mishra

Examples

```
  ## Not run:
    x <- reportKcal(gdx)

## End(Not run)
```

reportLaborCostsEmpl *reportLaborCostsEmpl*

Description

reports MAgPIE labor costs that go into employment calculation

Usage

```
reportLaborCostsEmpl(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

magpie object with labor costs

Labor costs for employment variables

Name	Unit	Meta
Labor|Employment|Labor costs linked to employment	million US$2017/yr	Total labor costs used for employment calcula

Author(s)

Debbora Leip

Examples

```
  ## Not run:
    x <- reportLaborCostsEmpl(gdx)

## End(Not run)
```

reportLaborProductivity *reportLaborProductivity*

Description

reports labor productivity in crop production

Usage

```
reportLaborProductivity(
  gdx,
  productAggr = TRUE,
  type = "physical",
  level = "regglo"
)
```

Arguments

gdx	GDX file
productAggr	Aggregate over products or not (boolean)
type	type of labor productivity, so far only physical (kg DM / h)
level	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

labor productivity as MAgPIE object

Labor productivity variables

Name	Unit	Meta
Labor|Productivity|Physical labor productivity|Crop products	kg DM per hour	Physical labor productivity for crops

Author(s)

Debbora Leip

Examples

```
## Not run:
  x <- reportLaborProductivity(gdx)

## End(Not run)
```

`reportLandConservation` *reportLandConservation*

Description

reports land conservation areas

Usage

```
reportLandConservation(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

land conservation area in Mha

Total conserved land variables

Name	Unit	Meta
Resources|Land Cover Conserved|Cropland	million ha	Total conserved cropland area (prote
Resources|Land Cover Conserved|Pastures and Rangelands	million ha	Total conserved pasture area (protect
Resources|Land Cover Conserved|Forest|Planted Forest	million ha	Total conserved planted forest area (p
Resources|Land Cover Conserved|Forest|Natural Forest|Primary Forest	million ha	Total conserved primary forest area (
Resources|Land Cover Conserved|Forest|Natural Forest|Secondary Forest	million ha	Total conserved secondary forest area
Resources|Land Cover Conserved|Urban Area	million ha	Total conserved urban area (protecte
Resources|Land Cover Conserved|Other Land	million ha	Total conserved other natural land ar

Protected land variables

Name	Unit	Meta
Resources|Land Cover Conserved|Cropland|+|Protected	million ha	Protected cropland area
Resources|Land Cover Conserved|Pastures and Rangelands|+|Protected	million ha	Protected pasture area
Resources|Land Cover Conserved|Forest|Planted Forest|+|Protected	million ha	Protected planted forest a
Resources|Land Cover Conserved|Forest|Natural Forest|Primary Forest|+|Protected	million ha	Protected primary forest
Resources|Land Cover Conserved|Forest|Natural Forest|Secondary Forest|+|Protected	million ha	Protected secondary fore
Resources|Land Cover Conserved|Urban Area|+|Protected	million ha	Protected urban area
Resources|Land Cover Conserved|Other Land|+|Protected	million ha	Protected other natural la

Restored land variables

Name	Unit	Meta
Resources|Land Cover Conserved|Cropland|+|Restored	million ha	Restored cropland area
Resources|Land Cover Conserved|Pastures and Rangelands|+|Restored	million ha	Restored pasture area
Resources|Land Cover Conserved|Forest|Planted Forest|+|Restored	million ha	Restored planted forest ar
Resources|Land Cover Conserved|Forest|Natural Forest|Primary Forest|+|Restored	million ha	Restored primary forest a
Resources|Land Cover Conserved|Forest|Natural Forest|Secondary Forest|+|Restored	million ha	Restored secondary fores
Resources|Land Cover Conserved|Urban Area|+|Restored	million ha	Restored urban area
Resources|Land Cover Conserved|Other Land|+|Restored	million ha	Restored other natural lan

Annual restoration variables

Name	Unit	Meta
Resources|Land Cover Conserved|Cropland|Restored annually	million ha/yr	Annual croplan
Resources|Land Cover Conserved|Pastures and Rangelands|Restored annually	million ha/yr	Annual pasture
Resources|Land Cover Conserved|Forest|Planted Forest|Restored annually	million ha/yr	Annual planted
Resources|Land Cover Conserved|Forest|Natural Forest|Primary Forest|Restored annually	million ha/yr	Annual primary
Resources|Land Cover Conserved|Forest|Natural Forest|Secondary Forest|Restored annually	million ha/yr	Annual seconda
Resources|Land Cover Conserved|Urban Area|Restored annually	million ha/yr	Annual urban a
Resources|Land Cover Conserved|Other Land|Restored annually	million ha/yr	Annual other la

Cumulative restoration variables

Name	Unit	M
Resources|Land Cover Conserved|Cropland|Restored cumulatively	million ha since 2025	Cu
Resources|Land Cover Conserved|Pastures and Rangelands|Restored cumulatively	million ha since 2025	Cu
Resources|Land Cover Conserved|Forest|Planted Forest|Restored cumulatively	million ha since 2025	Cu
Resources|Land Cover Conserved|Forest|Natural Forest|Primary Forest|Restored cumulatively	million ha since 2025	Cu
Resources|Land Cover Conserved|Forest|Natural Forest|Secondary Forest|Restored cumulatively	million ha since 2025	Cu
Resources|Land Cover Conserved|Urban Area|Restored cumulatively	million ha since 2025	Cu
Resources|Land Cover Conserved|Other Land|Restored cumulatively	million ha since 2025	Cu

Author(s)

Patrick v. Jeetze, Florian Humpenoeder

Examples

```
## Not run:
x <- reportLandConservation(gdx)

## End(Not run)
```

`reportLandTransitionMatrix`

reportLandTransitionMatrix

Description

reports land transition matrix with gross land-use change flows between all land types

Usage

```
reportLandTransitionMatrix(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	The aggregation level to be used ("regglo" by default)

Value

land transition matrix as MAgPIE object (Mha/yr)

Total land transition variables

Name	Unit	Meta
Resources|Land Transitions	Mha/yr	Total gross land-use change (sum of all off-diagonal transitions)

From Cropland variables

Name	Unit	Meta
Resources|Land Transitions|+|From Cropland	Mha/yr	Total annual gross land-use change f
Resources|Land Transitions|From Cropland|+|To Pastures and Rangelands	Mha/yr	Annual gross transition from croplar
Resources|Land Transitions|From Cropland|+|To Urban Area	Mha/yr	Annual gross transition from croplar
Resources|Land Transitions|From Cropland|+|To Other Land	Mha/yr	Annual gross transition from croplar
Resources|Land Transitions|From Cropland|+|To Forest	Mha/yr	Annual gross transition from croplar
Resources|Land Transitions|From Cropland|To Forest|+|To Planted Forest	Mha/yr	Annual gross transition from croplar

Resources|Land Transitions|From Cropland|To Forest|+|To Primary Forest	Mha/yr	Annual gross transition from croplan
Resources|Land Transitions|From Cropland|To Forest|+|To Secondary Forest	Mha/yr	Annual gross transition from croplan

From Pastures and Rangelands variables

Name	Unit	Meta
Resources|Land Transitions|+|From Pastures and Rangelands	Mha/yr	Total annual gross la
Resources|Land Transitions|From Pastures and Rangelands|+|To Cropland	Mha/yr	Annual gross transiti
Resources|Land Transitions|From Pastures and Rangelands|+|To Urban Area	Mha/yr	Annual gross transiti
Resources|Land Transitions|From Pastures and Rangelands|+|To Other Land	Mha/yr	Annual gross transiti
Resources|Land Transitions|From Pastures and Rangelands|+|To Forest	Mha/yr	Annual gross transiti
Resources|Land Transitions|From Pastures and Rangelands|To Forest|+|To Planted Forest	Mha/yr	Annual gross transiti
Resources|Land Transitions|From Pastures and Rangelands|To Forest|+|To Primary Forest	Mha/yr	Annual gross transiti
Resources|Land Transitions|From Pastures and Rangelands|To Forest|+|To Secondary Forest	Mha/yr	Annual gross transiti

From Planted Forest variables

Name	Unit	Meta
Resources|Land Transitions|+|From Planted Forest	Mha/yr	Total annual gross land-use cha
Resources|Land Transitions|From Planted Forest|+|To Cropland	Mha/yr	Annual gross transition from pl
Resources|Land Transitions|From Planted Forest|+|To Pastures and Rangelands	Mha/yr	Annual gross transition from pl
Resources|Land Transitions|From Planted Forest|+|To Urban Area	Mha/yr	Annual gross transition from pl
Resources|Land Transitions|From Planted Forest|+|To Other Land	Mha/yr	Annual gross transition from pl
Resources|Land Transitions|From Planted Forest|+|To Forest	Mha/yr	Annual gross transition from pl
Resources|Land Transitions|From Planted Forest|To Forest|+|To Primary Forest	Mha/yr	Annual gross transition from pl
Resources|Land Transitions|From Planted Forest|To Forest|+|To Secondary Forest	Mha/yr	Annual gross transition from pl

From Primary Forest variables

Name	Unit	Meta
Resources|Land Transitions|+|From Primary Forest	Mha/yr	Total annual gross land-use cha
Resources|Land Transitions|From Primary Forest|+|To Cropland	Mha/yr	Annual gross transition from pr
Resources|Land Transitions|From Primary Forest|+|To Pastures and Rangelands	Mha/yr	Annual gross transition from pr
Resources|Land Transitions|From Primary Forest|+|To Urban Area	Mha/yr	Annual gross transition from pr
Resources|Land Transitions|From Primary Forest|+|To Other Land	Mha/yr	Annual gross transition from pr
Resources|Land Transitions|From Primary Forest|+|To Forest	Mha/yr	Annual gross transition from pr
Resources|Land Transitions|From Primary Forest|To Forest|+|To Planted Forest	Mha/yr	Annual gross transition from pr

Resources|Land Transitions|From Primary Forest|To Forest|+|To Secondary Forest	Mha/yr	Annual gross transition from p

From Secondary Forest variables

Name	Unit	Meta
Resources|Land Transitions|+|From Secondary Forest	Mha/yr	Total annual gross land-use cha
Resources|Land Transitions|From Secondary Forest|+|To Cropland	Mha/yr	Annual gross transition from se
Resources|Land Transitions|From Secondary Forest|+|To Pastures and Rangelands	Mha/yr	Annual gross transition from se
Resources|Land Transitions|From Secondary Forest|+|To Urban Area	Mha/yr	Annual gross transition from se
Resources|Land Transitions|From Secondary Forest|+|To Other Land	Mha/yr	Annual gross transition from se
Resources|Land Transitions|From Secondary Forest|+|To Forest	Mha/yr	Annual gross transition from se
Resources|Land Transitions|From Secondary Forest|To Forest|+|To Planted Forest	Mha/yr	Annual gross transition from se
Resources|Land Transitions|From Secondary Forest|To Forest|+|To Primary Forest	Mha/yr	Annual gross transition from se

From Urban Area variables

Name	Unit	Meta
Resources|Land Transitions|+|From Urban Area	Mha/yr	Total annual gross land-use chang
Resources|Land Transitions|From Urban Area|+|To Cropland	Mha/yr	Annual gross transition from urba
Resources|Land Transitions|From Urban Area|+|To Pastures and Rangelands	Mha/yr	Annual gross transition from urba
Resources|Land Transitions|From Urban Area|+|To Other Land	Mha/yr	Annual gross transition from urba
Resources|Land Transitions|From Urban Area|+|To Forest	Mha/yr	Annual gross transition from urba
Resources|Land Transitions|From Urban Area|To Forest|+|To Planted Forest	Mha/yr	Annual gross transition from urba
Resources|Land Transitions|From Urban Area|To Forest|+|To Primary Forest	Mha/yr	Annual gross transition from urba
Resources|Land Transitions|From Urban Area|To Forest|+|To Secondary Forest	Mha/yr	Annual gross transition from urba

From Other Land variables

Name	Unit	Meta
Resources|Land Transitions|+|From Other Land	Mha/yr	Total annual gross land-use change
Resources|Land Transitions|From Other Land|+|To Cropland	Mha/yr	Annual gross transition from other
Resources|Land Transitions|From Other Land|+|To Pastures and Rangelands	Mha/yr	Annual gross transition from other
Resources|Land Transitions|From Other Land|+|To Urban Area	Mha/yr	Annual gross transition from other
Resources|Land Transitions|From Other Land|+|To Forest	Mha/yr	Annual gross transition from other
Resources|Land Transitions|From Other Land|To Forest|+|To Planted Forest	Mha/yr	Annual gross transition from other
Resources|Land Transitions|From Other Land|To Forest|+|To Primary Forest	Mha/yr	Annual gross transition from other
Resources|Land Transitions|From Other Land|To Forest|+|To Secondary Forest	Mha/yr	Annual gross transition from other

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportLandTransitionMatrix(gdx)

## End(Not run)
```

`reportLandUse` *reportLandUse*

Description

reports land-use

Usage

```
reportLandUse(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	The aggregation level to be used ("regglo" by default)

Value

land-use as MAgPIE object (million ha)

Total land cover variables

Name	Unit	Meta
Resources|Land Cover	million ha	Total land cover
Resources|Land Cover|Agricultural land	million ha	Agricultural land including cropland and pastures

Cropland variables

Name	Unit	Meta
Resources|Land Cover|+|Cropland	million ha	Arable land, i.e. land in bioenergy crop, food, and feed/fodder c
Resources|Land Cover|Cropland|+|Croparea	million ha	Physical cropland area used for crop production
Resources|Land Cover|Cropland|+|Fallow	million ha	Fallow cropland
Resources|Land Cover|Cropland|+|Tree Cover	million ha	Trees on cropland for agroforestry

Pasture and urban variables

Name	Unit	Meta
Resources|Land Cover|+|Pastures and Rangelands	million ha	Pasture and range land based on FAO definition of permaner
Resources|Land Cover|+|Urban Area	million ha	Built-up land associated with human settlements

Other natural land variables

Name	Unit	Meta
Resources|Land Cover|+|Other Land	million ha	Non-forest natural land including primary non-forest, restored ar
Resources|Land Cover|Other Land|Initial	million ha	Primary non-forest natural land without clearly visible indicatio
Resources|Land Cover|Other Land|Recovered	million ha	Recovered natural land due to the abandonment of agricultural o
Resources|Land Cover|Other Land|Restored	million ha	Intentionally restored natural land for the purpose of nature and/

Forest variables

Name	Unit	Meta
Resources|Land Cover|+|Forest	million ha	Sum of primary, secondary and plante
Resources|Land Cover|Forest|+|Natural Forest	million ha	Naturally regenerated forest including
Resources|Land Cover|Forest|Natural Forest|+|Primary Forest	million ha	Naturally regenerated forest of native t
Resources|Land Cover|Forest|Natural Forest|+|Secondary Forest	million ha	Forest predominantly composed of tre
Resources|Land Cover|Forest|Natural Forest|Secondary Forest|Young	million ha	Young secondary forest
Resources|Land Cover|Forest|Natural Forest|Secondary Forest|Mature	million ha	Mature secondary forest
Resources|Land Cover|Forest|+|Planted Forest	million ha	Forest predominantly composed of tre
Resources|Land Cover|Forest|Planted Forest|+|Timber	million ha	Planted forest established to meet timb
Resources|Land Cover|Forest|Planted Forest|+|CO2-price AR	million ha	Afforestation/reforestation established
Resources|Land Cover|Forest|Planted Forest|CO2-price AR|+|Plantation	million ha	CO2-price AR grown as a monocultur
Resources|Land Cover|Forest|Planted Forest|CO2-price AR|+|Natural	million ha	CO2-price AR grown with native tree
Resources|Land Cover|Forest|Planted Forest|+|NPI_NDC AR	million ha	Afforestation/reforestation established

Resources|Land Cover|Forest|Planted Forest|+|Other Planted	million ha	Planted forest present at model initiali

Author(s)

Florian Humpenoeder, Kristine Karstens, Isabelle Weindl

Examples

```
  ## Not run:
    x <- reportLandUse(gdx)

## End(Not run)
```

`reportLandUseChange` *reportLandUseChange*

Description

reports land-use change

Usage

```
reportLandUseChange(gdx, baseyear = 1995, level = "regglo")
```

Arguments

`gdx`	GDX file
`baseyear`	baseyear for calculating land-use change
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

land-use change as MAgPIE object (million ha wrt to baseyear)

Land-use change variables

Name	Unit	Meta
Resources|Land Cover Change|Cropland	million ha wrt baseyear	Change in cropland area relative to basey
Resources|Land Cover Change|Pastures and Rangelands	million ha wrt baseyear	Change in pasture area relative to baseye
Resources|Land Cover Change|Forest	million ha wrt baseyear	Change in forest area relative to baseyea
Resources|Land Cover Change|Other Land	million ha wrt baseyear	Change in other land area relative to bas

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportLandUseChange(gdx)

## End(Not run)
```

`reportLandUseForSEALS` *reportLandUseForSEALS*

Description

Writes MAgPIE land use projections to a specific NetCDF that can be read by the Spatial Economic Allocation Landscape Simulator (SEALS) model for generating high resolution land use maps.

Usage

```
reportLandUseForSEALS(
  outputdir = ".",
  magCellLand = "cell.land_0.5_share.mz",
  outFile = NULL,
  selectyears = c(2020, 2030, 2050)
)
```

Arguments

`outputdir`	output directory which contains cellular magpie output
`magCellLand`	Disaggregated land use (grid-cell land area share) as magclass object or file (.mz) from a MAgPIE run.
`outFile`	a file name the output should be written to using `ncdf4::nc_create` and `ncdf4::ncvar_put`
`selectyears`	Numeric vector of years to provide data for.

Value

Proportions of different land use classes per grid sell in a NetCDF format.

SEALS land use output

This function produces NetCDF files (not MAgPIE reporting variables) with land use proportions for crop, past, primforest, secdforest, forestry, urban, and other land types.

Author(s)

Patrick v. Jeetze

Examples

```
## Not run:
x <- reportLandUseForSEALS(
  magCellLand = "cell.land_0.5_share.mz",
  outFile = "cell.land_0.5_SEALS.nc",
  selectyears = c(2020, 2030, 2050)
)

## End(Not run)
```

---

reportLivestockDemStructure

reportLivestockDemStructure

Description

reports the share of different livestock products (excluding fish) in total livestock calorie food supply

Usage

```
reportLivestockDemStructure(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

livestock demand structure as MAgPIE object (kcal/kcal)

Livestock demand structure variables

Name	Unit	Meta
Nutrition|Dietary Composition|Livestock Demand Structure|+|Ruminant meat	kcal/kcal	Share of ruminant meat in
Nutrition|Dietary Composition|Livestock Demand Structure|+|Poultry meat and eggs	kcal/kcal	Share of poultry and eggs i
Nutrition|Dietary Composition|Livestock Demand Structure|+|Dairy	kcal/kcal	Share of dairy in livestock
Nutrition|Dietary Composition|Livestock Demand Structure|+|Monogastric meat	kcal/kcal	Share of monogastric meat

Author(s)

Isabelle Weindl

Examples

```
  ## Not run:
    x <- reportLivestockDemStructure(gdx)

## End(Not run)
```

`reportLivestockShare` *reportLivestockShare*

Description

reports the share of livestock products (including fish) in total calorie food supply

Usage

```
reportLivestockShare(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

per-capita calories as MAgPIE object (kcal/cap/day)

Livestock share variables

Name	Unit	Meta
Nutrition|Dietary Composition|Livestock Share	kcal/kcal	Share of livestock products (incl. fish) in total calorie supply

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportLivestockShare(gdx)

## End(Not run)
```

`reportManure` *reportManure*

Description

Reports the Nitrogen in Manure of all animals for future MAgPIE projections

Usage

```
reportManure(gdx, nutrient = "nr", level = "regglo")
```

Arguments

`gdx`	GDX file
`nutrient`	nr, p, c...
`level`	aggregation level of returned data ("regglo" by default)

Manure variables

Name	Unit	Meta
Resources|Nitrogen|Manure	Mt Nr/yr	Total manure nitrogen production
Resources|Nitrogen|Manure|++|Confinement	Mt Nr/yr	Manure from confined livestock
Resources|Nitrogen|Manure|++|Grazing	Mt Nr/yr	Manure deposited during grazing
Resources|Nitrogen|Manure|+|Ruminants	Mt Nr/yr	Manure from ruminant animals
Resources|Nitrogen|Manure|+|Monogastric	Mt Nr/yr	Manure from monogastric animals

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportManure(gdx)

## End(Not run)
```

`reportManureFuel` *reportManureFuel*

Description

Reports manure used as direct combustion fuel from a MAgPIE GDX file at ISO country level. Converts from nitrogen mass (Mt N) to energy (PJ) using livestock-specific heating values and nitrogen contents from Hoyos-Sebá et al. (2024).

Usage

```
reportManureFuel(gdx, file = NULL)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie

Value

MAgPIE object with manure fuel at ISO level in PJ.

Author(s)

Kristine Karstens

Examples

```
## Not run:
  x <- reportManureFuel(gdx)

## End(Not run)
```

`reportNetForestChange` *reportNetForestChange*

Description

reports net and gross forest area change

Usage

```
reportNetForestChange(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

NetForestChange as magclass object (Mha per year)

Net forest change variables

Name	Unit	Meta
Resources|NetForestChange	Mha/yr	Annual net change in total forest area

Gross forest loss variables

Name	Unit	Meta
Resources|GrossForestLoss	Mha/yr	Annual gross deforested area
Resources|GrossForestLoss|+|Primary	Mha/yr	Annual gross loss of primary forest area
Resources|GrossForestLoss|+|Secondary	Mha/yr	Annual gross loss of secondary forest area
Resources|GrossForestLoss|+|Planted	Mha/yr	Annual gross loss of planted forest area
Resources|GrossForestLoss|Planted|Plantation|+|Timber	Mha/yr	Annual gross loss of timber plantation area
Resources|GrossForestLoss|Planted|Plantation|+|CO2-price AR	Mha/yr	Annual gross loss of CO2-price afforestation/refo
Resources|GrossForestLoss|Planted|Natural|+|CO2-price AR	Mha/yr	Annual gross loss of CO2-price afforestation/refo
Resources|GrossForestLoss|Planted|Natural|+|NPI_NDC AR	Mha/yr	Annual gross loss of NPI/NDC afforestation/refor

Gross forest gain variables

Name	Unit	Meta
Resources|GrossForestGain	Mha/yr	Annual gross expansion of forest area
Resources|GrossForestGain|+|Primary	Mha/yr	Annual gross expansion of primary forest area
Resources|GrossForestGain|+|Secondary	Mha/yr	Annual gross expansion of secondary forest area
Resources|GrossForestGain|+|Planted	Mha/yr	Annual gross expansion of planted forest area
Resources|GrossForestGain|Planted|Plantation|+|Timber	Mha/yr	Annual gross expansion of timber plantation area
Resources|GrossForestGain|Planted|Plantation|+|CO2-price AR	Mha/yr	Annual gross expansion of carbon plantation area
Resources|GrossForestGain|Planted|Natural|+|CO2-price AR	Mha/yr	Annual gross reforestation and/or afforestation ar
Resources|GrossForestGain|Planted|Natural|+|NPI_NDC AR	Mha/yr	Annual gross expansion of NPI/NDC afforestatio

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- reportNetForestChange(gdx)

## End(Not run)
```

`reportNitrogenBudgetCropland`
reportNitrogenBudgetCropland

Description

Reports the Nitrogen Budgets of Croplands for future MAgPIE projections

Usage

```
reportNitrogenBudgetCropland(gdx, include_emissions = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`include_emissions`	TRUE also divides the N surplus into different emissions
`level`	aggregation level of returned data ("regglo" by default); use "grid" for 0.5 degree grid level

Nitrogen inputs variables

Name	Unit	Meta
Resources|Nitrogen|Cropland Budget|Inputs	Mt Nr/yr	Total nitrogen inputs to cropland
Resources|Nitrogen|Cropland Budget|Inputs|+|Fertilizer	Mt Nr/yr	Synthetic fertilizer nitrogen applied to cr
Resources|Nitrogen|Cropland Budget|Inputs|+|Manure	Mt Nr/yr	Manure nitrogen applied to cropland
Resources|Nitrogen|Cropland Budget|Inputs|+|BNF	Mt Nr/yr	Biological nitrogen fixation on cropland
Resources|Nitrogen|Cropland Budget|Inputs|+|Atmospheric deposition	Mt Nr/yr	Atmospheric nitrogen deposition on crop
Resources|Nitrogen|Cropland Budget|Inputs|+|Seeds	Mt Nr/yr	Nitrogen from seeds

Nitrogen withdrawals variables

Name	Unit	Meta
Resources|Nitrogen|Cropland Budget|Withdrawals	Mt Nr/yr	Total nitrogen withdrawals from cr
Resources|Nitrogen|Cropland Budget|Withdrawals|+|Harvest	Mt Nr/yr	Nitrogen removed in harvested crop
Resources|Nitrogen|Cropland Budget|Withdrawals|+|Above-ground residues	Mt Nr/yr	Nitrogen in above-ground crop resi
Resources|Nitrogen|Cropland Budget|Withdrawals|+|Below-ground residues	Mt Nr/yr	Nitrogen in below-ground crop resi

Nitrogen balance variables

Name	Unit	Meta
Resources|Nitrogen|Cropland Budget|Balance	Mt Nr/yr	Total nitrogen balance on cropland
Resources|Nitrogen|Cropland Budget|Balance|+|Surplus	Mt Nr/yr	Nitrogen surplus on cropland (inputs minus
Resources|Nitrogen|Cropland Budget|Balance|+|Soil organic matter	Mt Nr/yr	Net nitrogen flow into soil organic matter (n
Resources|Nitrogen|Cropland Budget|Balance|+|Balance flow	Mt Nr/yr	Nitrogen balance flow (calibration term)

Author(s)

Benjamin Leon Bodirsky

See Also

NitrogenBudget

Examples

```
## Not run:
x <- reportNitrogenBudgetCropland(gdx)

## End(Not run)
```

`reportNitrogenBudgetNonagland` *reportNitrogenBudgetNonagland*

Description

Reports the Nitrogen Budgets of non-agricultural lands for future MAgPIE projections

Usage

```
reportNitrogenBudgetNonagland(gdx, level = "reg")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("reg" by default); use "grid" for grid level

Non-agricultural nitrogen budget variables

Name	Unit	Meta
Resources|Nitrogen|Non-Agricultural Land Budget|Balance	Mt Nr/yr	Nitrogen surplus on non-agricultural lan
Resources|Nitrogen|Non-Agricultural Land Budget|Inputs	Mt Nr/yr	Total nitrogen inputs to non-agricultural
Resources|Nitrogen|Non-Agricultural Land Budget|Inputs|+|Fixation	Mt Nr/yr	Biological N fixation on non-ag land
Resources|Nitrogen|Non-Agricultural Land Budget|Inputs|+|Deposition	Mt Nr/yr	Atmospheric N deposition on non-ag lar

Author(s)

Benjamin Leon Bodirsky

See Also

`NitrogenBudget`

Examples

```
  ## Not run:
    x <- reportNitrogenBudgetNonagland(gdx)

## End(Not run)
```

`reportNitrogenBudgetPasture` *reportNitrogenBudgetCropland*

Description

Reports the Nitrogen Budgets of Croplands for future MAgPIE projections

Usage

```
reportNitrogenBudgetPasture(gdx, include_emissions = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`include_emissions`	TRUE also divides the N surplus into different emissions
`level`	aggregation level of returned data ("regglo" by default); use "grid" for grid level

Nitrogen pasture budget variables

Name	Unit	Meta
Resources|Nitrogen|Pasture Budget|Inputs	Mt Nr/yr	Total nitrogen inputs to pastures
Resources|Nitrogen|Pasture Budget|Inputs|+|Manure	Mt Nr/yr	Manure nitrogen deposited on pastures
Resources|Nitrogen|Pasture Budget|Inputs|+|Atmospheric deposition	Mt Nr/yr	Atmospheric nitrogen deposition on pastur
Resources|Nitrogen|Pasture Budget|Withdrawals	Mt Nr/yr	Total nitrogen withdrawals from pastures
Resources|Nitrogen|Pasture Budget|Balance	Mt Nr/yr	Nitrogen balance on pastures
Resources|Nitrogen|Pasture Budget|Balance|+|Surplus	Mt Nr/yr	Nitrogen surplus on pastures

Author(s)

Benjamin Leon Bodirsky

See Also

`NitrogenBudget`

Examples

```
## Not run:
  x <- reportNitrogenBudgetCropland(gdx)

## End(Not run)
```

`reportNitrogenEfficiencies` *reportNitrogenEfficiencies*

Description

Reports different nitrogen use efficiency indicators

Usage

```
reportNitrogenEfficiencies(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Nitrogen efficiency variables

Name	Unit	Meta
Resources|Nitrogen|Cropland Budget|Nitrogen Use Efficiency complete	Mt Nr/Mt Nr	Complete nitrogen use efficiency or
Resources|Nitrogen|Cropland Budget|Nitrogen Use Efficiency basic	Mt Nr/Mt Nr	Basic nitrogen use efficiency (harve
Resources|Nitrogen|Cropland Budget|Soil Nitrogen Uptake Efficiency	Mt Nr/Mt Nr	Soil nitrogen uptake efficiency
Resources|Nitrogen|Pasture Budget|Nitrogen Use Efficiency complete	Mt Nr/Mt Nr	Complete nitrogen use efficiency or

Author(s)

Benjamin Leon Bodirsky

See Also

`reportNitrogenEfficiencies`

Examples

```
## Not run:
x <- reportNitrogenEfficiencies(gdx)

## End(Not run)
```

`reportNitrogenPollution` *reportNitrogenPollution*

Description

Reports total Nitrogen Pollution as the sum of surplus from cropland, pasture, awms, consumption and non-agricutlural land

Usage

```
reportNitrogenPollution(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Nitrogen pollution variables

Name	Unit	Meta
Resources|Nitrogen|Pollution|Surplus	Mt Nr/yr	Total nitrogen pollution surplus from all sou
Resources|Nitrogen|Pollution|Surplus|+|Cropland	Mt Nr/yr	Nitrogen surplus from cropland
Resources|Nitrogen|Pollution|Surplus|+|Pasture	Mt Nr/yr	Nitrogen surplus from pasture
Resources|Nitrogen|Pollution|Surplus|+|Animal waste management	Mt Nr/yr	Nitrogen losses from animal waste managen
Resources|Nitrogen|Pollution|Surplus|+|Non-agricultural land	Mt Nr/yr	Nitrogen surplus from non-agricultural land
Resources|Nitrogen|Pollution|Surplus|+|End-of-life losses	Mt Nr/yr	Nitrogen losses from food consumption and

Nitrogen aggregate variables

Name	Unit	Meta
Resources|Nitrogen|Nutrient surplus from agricultural land	Mt Nr/yr	Nitrogen surplus from crop
Resources|Nitrogen|Nutrient surplus from agricultural land and manure management	Mt Nr/yr	Nitrogen surplus from crop
Resources|Nitrogen|Nutrient surplus from all land and manure management	Mt Nr/yr	Nitrogen surplus from all l

Author(s)

Benjamin Leon Bodirsky, Michael Crawford

See Also

NitrogenBudget

Examples

```
## Not run:
x <- reportNitrogenPollution(gdx)

## End(Not run)
```

`reportOutputPerWorker` *reportOutputPerWorker*

Description

reports output per worker in crop+livestock production from MAgPIE results

Usage

```
reportOutputPerWorker(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

output per worker as MAgPIE object

Output per worker variables

Name	Unit	Meta
Labor|Productivity|Monetary output per worker	US$2017/worker	Monetary output per agricultural worker

Author(s)

Debbora Leip

Examples

```
## Not run:
  x <- reportOutputPerWorker(gdx)

## End(Not run)
```

`reportPastSoilCarbon` *reportPastSoilCarbon*

Description

reports pasture soil carbon

Usage

```
reportPastSoilCarbon(gdx)
```

Arguments

`gdx`	GDX file

Value

Soil carbon values as magpie objetc

Pasture soil carbon variables

Name	Unit	Meta
Resources|Soil Carbon Target|Pasture|Continuous grazing|Density	tC per ha	Target soil carbon density for pasture
Resources|Soil Carbon Target Change|Pasture|Continuous grazing|Density	tC per ha	Change in target soil carbon density

Author(s)

Marcos Alves

Examples

```
## Not run:
x <- reportPastSoilCarbon(gdx)

## End(Not run)
```

reportPBbiosphere *reportPBbiosphere*

Description

reports biosphere planetary boundary: Share of intact land relative to total land area (unitless) Share of intact land covered by areas within Global Safety Net (unitless) Share of land area that satisfies landscape target (unitless)

Usage

```
reportPBbiosphere(gdx, level = "regglo", intactnessRule = "carbon:0.95")
```

Arguments

gdx	GDX file
level	level of aggregation (regglo: regions and global)
intactnessRule	rule for intact land can be based on percentage of potential carbon density reached or on age classes for secondary forests, planted forest and other natural land. The argument is split into two components: rule: carbon or ageclass threshold: share of carbon density reached to be classified as intact or threshold in years can be set via this argument Example: "carbon:0.95" or "ageclass:70"

Value

MAgPIE object

Biosphere planetary boundary variables

Name	Unit	Meta
Planetary Boundary|Biosphere|Share of intact land relative to total land area	unitless	Fraction of intact la
Planetary Boundary|Biosphere|Share of intact land covered by areas within Global Safety Net	unitless	Share of intact land
Planetary Boundary|Biosphere|Share of land area that satisfies landscape target	unitless	Share of land wher

Author(s)

Patrick von Jeetze, Felicitas Beier

Examples

```
## Not run:
x <- reportPBbiosphere(gdx)

## End(Not run)
```

reportPBland *reportPBland*

Description

reports land planetary boundary: forest area

Usage

```
reportPBland(gdx, level = "regglo", foresttype = "all")
```

Arguments

gdx	GDX file
level	level of aggregation (regglo: regions and global)
foresttype	managed forest types that are included in the calculation of the forest area (all: all managed forests, noTimber: timber plantations are not counted)

Value

MAgPIE object

Land planetary boundary variables

Name	Unit	Meta
Planetary Boundary|Land|Forest cover	Mha	Total forest area (natural and managed forests)

Author(s)

Felicitas Beier, Patrick von Jeetze

Examples

```
  ## Not run:
    x <- reportPBland(gdx)

## End(Not run)
```

reportPBnitrogen *reportPBnitrogen*

Description

reports nitrogen planetary boundary

Usage

```
reportPBnitrogen(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	level of aggregation (regglo: regions and global)

Value

MAgPIE object

Nitrogen planetary boundary variables

Name	Unit	Meta
Planetary Boundary|Nitrogen|Agricultural Nitrogen surplus	Mt N/yr	Total nitrogen surplus from cropland and pasture

Author(s)

Felicitas Beier, Mike Crawford

Examples

```
  ## Not run:
    x <- reportPBnitrogen(gdx)

## End(Not run)
```

reportPBwater *reportPBwater*

Description

reports water planetary boundaries

Usage

```
reportPBwater(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	level of aggregation (regglo: regions and global)

Value

MAgPIE object

Water planetary boundary variables

Name	Unit	Meta
Planetary Boundary|Freshwater|Water consumption	km3/yr	Total blue water consumption (agricultural and non-agricultur

Author(s)

Felicitas Beier, Jens Heinke

Examples

```
  ## Not run:
    x <- reportPBwater(gdx)

## End(Not run)
```

reportPeatland *reportPeatland*

Description

reports peatland area

Usage

```
reportPeatland(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

peatland area as magclass object (million ha)

Peatland variables

Name	Unit	Meta
Resources|Peatland	million ha	Area of peatlands
Resources|Peatland|+|Intact	million ha	Area of intact peatlands
Resources|Peatland|+|Degraded	million ha	Area of drained peatlands
Resources|Peatland|+|Rewetted	million ha	Area of rewetted peatlands

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
  x <- reportPeatland(gdx)

## End(Not run)
```

`reportPlantationEstablishment` *reportPlantationEstablishment*

Description

reports MAgPIE harvested area for timber.

Usage

```
reportPlantationEstablishment(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

Area harvested for timber production

Plantation establishment variables

Name	Unit	Meta
Resources|Timber operations|Area Newly Established|Forestry	Mha per yr	Annual area of new timber plantations establis

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- reportPlantationEstablishment(gdx)

## End(Not run)
```

reportPopulation *reportPopulation*

Description

reports Population

Usage

```
reportPopulation(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

population as MAgPIE object

Population variables

Name	Unit	Meta
Population	million people	Total population

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportPopulation(gdx)

## End(Not run)
```

reportPriceAgriculture *reportPriceAgriculture*

Description

reports food commodity prices

Usage

```
reportPriceAgriculture(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

agricultural commodity prices as MAgPIE object (USD)

Agricultural price variables

Name	Unit	Meta
Prices|Agriculture|Crops	US$2017/tDM	Prices for crop products
Prices|Agriculture|Livestock products	US$2017/tDM	Prices for livestock products

Author(s)

Mishko Stevanovic

Examples

```
## Not run:
  x <- reportPriceAgriculture(gdx)

## End(Not run)
```

`reportPriceBioenergy` *reportPriceBioenergy*

Description

reports bioenergy prices

Usage

```
reportPriceBioenergy(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

bioenergy price as MAgPIE object Unit: see names

Bioenergy price variables

Name	Unit	Meta
Prices|Bioenergy	US$2017/GJ	Bioenergy price

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportPriceBioenergy(gdx)

## End(Not run)
```

reportPriceElasticities *reportHunger*

Description

Calculates the share of people living in hunger.

Usage

```
reportPriceElasticities(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

magpie object with hunger (mio people) or hunger share

Price elasticity variables

Name	Unit	Meta
Food Supply|PriceElasticities|Total Calories	%/%	Price elasticity of total calorie demand
Food Supply|PriceElasticities|Staples	%/%	Price elasticity of staple food demand
Food Supply|PriceElasticities|Livestock Products	%/%	Price elasticity of livestock product demand
Food Supply|PriceElasticities|Vegetables, Fruits and Nuts	%/%	Price elasticity of fruits and vegetables demand

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportHunger(gdx)

## End(Not run)
```

`reportPriceFoodIndex` *reportPriceFoodIndex*

Description

reports food price index

Usage

```
reportPriceFoodIndex(gdx, baseyear = "y2020", level = "regglo")
```

Arguments

`gdx`	GDX file
`baseyear`	baseyear of the price index
`level`	aggregation level of returned data ("regglo" by default)

Value

Food price index as MAgPIE object Unit: see names

Food price index variables

Name	Unit	Meta
Prices|Index|Agriculture|Food products	1	Food price index relative to baseyear
Prices|Index|Agriculture|Food products|Plant-based	1	Plant-based food price index
Prices|Index|Agriculture|Food products|Plant-based|Maize	1	Maize price index
Prices|Index|Agriculture|Food products|Plant-based|Rice	1	Rice price index
Prices|Index|Agriculture|Food products|Plant-based|Soybean	1	Soybean price index
Prices|Index|Agriculture|Food products|Plant-based|Temperate cereals	1	Wheat/temperate cereals price index
Prices|Index|Agriculture|Food products|Livestock	1	Livestock food price index

Author(s)

Florian Humpenoeder, Felicitas Beier

Examples

```
## Not run:
  x <- reportPriceFoodIndex(gdx)

## End(Not run)
```

reportPriceGHG *reportPriceGHG*

Description

reports GHG emission prices

Usage

```
reportPriceGHG(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

GHG emission prices as MAgPIE object

GHG price variables

Name	Unit	Meta
Prices|GHG Emission|CO2	US$2017/tCO2	Carbon dioxide emission price
Prices|GHG Emission|N2O	US$2017/tN2O	Nitrous oxide emission price
Prices|GHG Emission|CH4	US$2017/tCH4	Methane emission price

Author(s)

Florian Humpenoeder, Amsalu W. Yalew

Examples

```
  ## Not run:
    x <- reportPriceGHG(gdx)

## End(Not run)
```

reportPriceLand *reportPriceLand*

Description

reports land prices (land rent)

Usage

```
reportPriceLand(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

land prices as MAgPIE object Unit: see names

Land price variables

Name	Unit	Meta
Prices|Land|Cropland	US$2017/ha	Land rent (shadow price of cropland constraint)

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportPriceLand(gdx)

## End(Not run)
```

reportPriceShock *reportPriceShock*

Description

Reports the change in consumption and expenditure due to higher or lower food prices

Usage

```
reportPriceShock(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

magpie object

Price shock variables

Name	Unit	Meta
Food Supply|Calorie Supply|Price Induced Change|Absolute|Total Calories	kcal/cap/day	Absolute total
Food Supply|Calorie Supply|Price Induced Change|Absolute|Livestock Calories	kcal/cap/day	Livestock
Food Supply|Calorie Supply|Price Induced Change|Relative|Total Calories	kcal/kcal	Relative total
Household Expenditure|Food|Price Induced Change|Absolute|Food Expenditure	USD/cap	Absolute expense
Household Expenditure|Food|Price Induced Change|Relative|Food Expenditure	USD/USD	Relative expense

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
  x <- reportPriceShock(gdx)

## End(Not run)
```

reportPriceWater *reportPriceWater*

Description

reports water prices

Usage

```
reportPriceWater(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

water usage as MAgPIE object Unit: see names

Water price variables

Name	Unit	Meta
Prices|Water|Agriculture	Index 2005=100	Agricultural water price index

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
x <- reportPriceWater(gdx)

## End(Not run)
```

reportPriceWoodyBiomass

reportPriceWoodyBiomass

Description

reports woody biomass prices (land rent)

Usage

```
reportPriceWoodyBiomass(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

land prices as MAgPIE object Unit: see names

Woody biomass price variables

Name	Unit	Meta
Prices|Wood	US$2017/tDM	Wood price
Prices|Woodfuel	US$2017/tDM	Woodfuel price

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- reportPriceWoodyBiomass(gdx)

## End(Not run)
```

reportProcessing *reportProcessing*

Description

reportes processing input and output quantities primary-to-process or primary-to-secondary

Usage

```
reportProcessing(
  gdx,
  detail = TRUE,
  indicator = "primary_to_process",
  level = "regglo"
)
```

Arguments

gdx	GDX file
detail	if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
indicator	"primary_to_process" for process or "secondary_from_primary" for secondary product output
level	aggregation level of returned data ("regglo" by default)

Value

processing demand as MAgPIE object (Mt DM)

Processing demand variables

Name	Unit	Meta
Demand|Processing|++|Crops	Mt DM/yr	Primary crop products processed into secondary p
Demand|Processing|++|Livestock products	Mt DM/yr	Primary livestock products processed into seconda
Processing|Raw material|Processed into Secondary products	Mt DM/yr	Total raw materials processed into secondary prod

Author(s)

David Chen, Benjamin Leon Bodirsky

Examples

```
## Not run:
x <- reportProcessing(gdx = gdx, detail = TRUE, indicator = "primary_to_process")

## End(Not run)
```

`reportProcessingResiduesForestry`
reportProcessingResiduesForestry

Description

Reports processing residues from timber production available for energy use.

Processing residues are sawmill byproducts (sawdust, bark, slabs, offcuts) generated during conversion of industrial roundwood to finished products. They are NOT currently tracked as a variable in MAgPIE's GAMS code and represent additional biomass supply for REMIND's bioenergy sector.

Note: Logging residues (branches, tops) tracked in `v73_prod_residues` are NOT included here because they already feed into MAgPIE's woodfuel supply equation (`q73_prod_woodfuel`) and would be double-counted if reported separately for REMIND.

Usage

```
reportProcessingResiduesForestry(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	Level of regional aggregation ("regglo" by default)

Details

Full mass balance of timber harvest:

When 1 hectare of forest is harvested, the total biomass removed is:

- **Stem biomass** = `im_growing_stock` (tDM/ha)
- **Logging residues** (branches, tops) = 15% of stem, tracked in `v73_prod_residues`
- **Total removed** = stem x 1.15

The model splits stem production between industrial roundwood (`vm_prod("wood")`) and woodfuel (`vm_prod("woodfuel")`). Only the industrial roundwood fraction enters processing facilities and generates processing residues. Woodfuel is burned directly. Logging residues already feed into MAgPIE's woodfuel supply (NOT reported here).

```
1 hectare harvested -> total biomass = stem x 1.15
|
+-- Stem (100% = vm_prod, split between "wood" and "woodfuel" by the model)
|  If allocated to industrial roundwood (vm_prod("wood")) -> processing facilities:
|   |
|   +-- Sawnwood pathway (~57% of IR):
|   |   +-- 50% finished sawnwood
|  |  +-- 43% processing residues (sawdust, slabs, bark, offcuts) -> THIS FUNCTION
|   |   +--  7% losses
|   |
|   +-- Pulpwood pathway (~35% of IR):
|   |   +-- 95-100% utilized (black liquor burned internally at pulp mill)
|   |   +-- 0-5% solid residues available externally
|   |
|   +-- Other industrial roundwood (~8% of IR):
|       +-- Assumed similar to sawnwood (43% residue rate)
|
+-- Logging residues (15% of stem = branches, tops)
    -> Already in MAgPIE woodfuel via v73_prod_residues (NOT reported here)
```

Processing residue rate:

The rate is derived from FAO product composition of industrial roundwood (from `f73_prod_specific_timber`, stable at these shares across 1965-2015) and product-specific recovery rates:

Product	Share of IR	Recovery	Residue rate	Source
Sawnwood (sawlogs + veneer)	~57%	50%	**43%**	FAO/UNECE; Mantau 2012 UNECE DP-51
Pulpwood	~35%	95-100%	**0-5%**	Black liquor burned internally
Other industrial roundwood	~8%	~50%	**43%**	Assumed same as sawnwood

Production-weighted rate: 0.57 x 0.43 + 0.35 x 0.025 + 0.08 x 0.43 = **0.29** Rounded to **0.30** (stable at 30-35% across 1965-2015 FAO data).

Why only industrial roundwood, not woodfuel:

Processing residues are computed from industrial roundwood supply (`ov_supply("wood")`) only. Woodfuel (`ov_supply("woodfuel")`) is excluded because it is burned directly for energy — it does not pass through sawmills or pulp mills and therefore generates no processing residues.

Why all processing residues go to energy:

In reality, some processing residues are used for pulp or particleboard. However, MAgPIE models all wood products as aggregate "industrial roundwood" (`kforestry` = `wood`). The downstream split into sawnwood, pulp, and panels is not modeled — there is no competing use for processing residues within MAgPIE. Therefore, all processing residues are assumed available for energy use.

Construction wood:

`calcConstructionWoodDemand` in mrcommons applies a factor 2 to construction wood demand, reflecting a 50% sawmill recovery rate (Churkina et al. 2020). This means half of the harvested roundwood for construction becomes processing waste. These residues are clean, uniform offcuts with high energy recovery potential. Reported separately from general processing residues.

What this function does NOT include:

- **Logging residues** (branches, tops): already in MAgPIE woodfuel (`v73_prod_residues`)
- **Post-consumer waste wood** (demolition, furniture): excluded for consistency with `carbonLTS`, which tracks CO2 from HWP decay using IPCC first-order decay (half-lives: sawnwood 30yr, construction 60yr, pulpwood 2yr). Adding end-of-life energy recovery would require splitting HWP outflow into burned vs decayed fractions — assumptions not currently in the model. The CO2 from HWP decay is already reported; the energy from burning waste wood is not tracked to avoid parallel lifetime assumptions.
- **Black liquor**: burned internally at pulp mills, not available for external energy

References:

- FAO/UNECE conversion factors for sawnwood recovery rates
- Mantau, U. (2012). Wood flows in Europe (EU27). UNECE/FAO Discussion Paper 51 (DP-51).
- Oswalt et al. (2019). Forest Resources of the United States. USDA.
- Churkina et al. (2020). Buildings as a global carbon sink. Nature Sustainability.

Value

MAgPIE object with processing residues in PJ/yr

Variables

Name	Unit	Meta
Residues for energy|Forestry|+|Processing residues from construction wood	PJ/yr	Sawmill waste from construction wood
Residues for energy|Forestry|+|Processing residues from other roundwood	PJ/yr	30% of non-construction roundwood
Residues for energy|+|Forestry	PJ/yr	Total forestry processing residues for er

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
  x <- reportProcessingResiduesForestry(gdx)

## End(Not run)
```

reportProcessingWoodResidues *reportProcessingWoodResidues*

Description

Reports the potential of wood processing residues (sawmill byproducts) from a MAgPIE GDX file at ISO country level and converts to energy (PJ). Processing residues are computed from industrial roundwood demand using a 30

Usage

```
reportProcessingWoodResidues(gdx, file = NULL)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie

Value

MAgPIE object with potential wood processing residues at ISO level in PJ.

Author(s)

Kristine Karstens

Examples

```
## Not run:
  x <- reportProcessingWoodResidues(gdx)

## End(Not run)
```

reportProducerPriceIndex *reportProducerPriceIndex*

Description

reports producer price index

Usage

```
reportProducerPriceIndex(gdx, prod_groups = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
prod_groups	whether to return only product groups
level	aggregation level of returned data ("regglo" by default)

Value

Producer price index as MAgPIE object Unit: see names

Producer price index variables

Name	Unit	Meta
Prices|Index2020|Agriculture|Producer|Primary food products	Index 2020=100	Producer price index for primary food p
Prices|Index2020|Agriculture|Producer|Crops	Index 2020=100	Producer price index for crops
Prices|Index2020|Agriculture|Producer|Livestock products	Index 2020=100	Producer price index for livestock produ
Prices|Index2020|Agriculture|Producer|Bioenergy	Index 2020=100	Producer price index for bioenergy crop
Prices|Index2020|Agriculture|Producer|All agricultural products	Index 2020=100	Producer price index for all agricultural

Author(s)

Isabelle Weindl, David M CHen

Examples

```
  ## Not run:
    x <- reportProducerPriceIndex(gdx)

## End(Not run)
```

---

reportProduction *reportProduction*

Description

reports production

Usage

```
reportProduction(gdx, detail = FALSE, agmip = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
agmip	if agmip = TRUE, additional sector aggregates required for AgMIP are reported (e.g. "AGR")
level	aggregation level of returned data ("regglo" by default)

Value

production as MAgPIE object. Unit: see names

Production variables

Name	Unit	Meta
Production	Mt DM/yr	Total agricultural production
Production|+|Crops	Mt DM/yr	Production of crops
Production|+|Livestock products	Mt DM/yr	Production of livestock products (excluding fish)
Production|+|Secondary products	Mt DM/yr	Production of secondary products (processed agricultural goods)
Production|+|Pasture	Mt DM/yr	Production of pasture biomass
Production|+|Bioenergy crops	Mt DM/yr	Production of second-generation bioenergy crops (short rotation grasses, sho

Author(s)

Benjamin Leon Bodirsky, Isabelle Weindl

Examples

```
  ## Not run:
    x <- reportProduction(gdx)

## End(Not run)
```

---

reportProductionBioenergy

reportProductionBioenergy

Description

reports 2nd gen bioenergy production

Usage

```
reportProductionBioenergy(gdx, detail = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

production as MAgPIE object. Unit: see names

Bioenergy production variables

Name	Unit	Meta
Production|Bioenergy|2nd generation	EJ/yr	Second generation bioenergy production (grassy a
Production|Bioenergy|2nd generation|++|Grassy bioenergy crops	EJ/yr	Production from short rotation grasses
Production|Bioenergy|2nd generation|++|Woody bioenergy crops	EJ/yr	Production from short rotation trees
Production|Bioenergy|2nd generation|Cumulative	EJ	Cumulative second generation bioenergy producti

Author(s)

Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportProductionBioenergy(gdx)

## End(Not run)
```

reportProductionGrowth *reportProductionGrowth*

Description

reports production growth rate

Usage

```
reportProductionGrowth(gdx, detail = FALSE)
```

Arguments

gdx	GDX file
detail	if true, provides results for all commodities, otherwhise aggregates some groups

Value

Production growth rates (index)

Production growth variables

Name	Unit	Meta
Production|Production Growth Rate	Index	Production volume index relative to base year
Production|Production Growth Rate|+|Crop products	Index	Crop production growth index
Production|Production Growth Rate|+|Livestock products	Index	Livestock production growth index

Author(s)

Xiaoxi Wang

Examples

```
## Not run:
  x <- reportProductionGrowth(gdx="fulldata.gdx",detail=TRUE)

## End(Not run)
```

reportProductionNr *reportProductionNr*

Description

reports production in Nr analogous to reportProduction

Usage

```
reportProductionNr(gdx, detail = FALSE)
```

Arguments

gdx	GDX file
detail	if detail = FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")

Value

production as MAgPIE object. Unit: see names

Nitrogen production variables

Name	Unit	Meta
Production Nr	Mt Nr/yr	Total nitrogen in agricultural production
Production Nr|+|Crop products	Mt Nr/yr	Nitrogen in crop production
Production Nr|+|Livestock products	Mt Nr/yr	Nitrogen in livestock production

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
  x <- reportProductionNr(gdx)

## End(Not run)
```

reportProtein *reportProtein*

Description

reports per-capita protein food supply (including household waste)

Usage

```
reportProtein(gdx, detail = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

per-capita protein as MAgPIE object (protein/cap/day)

Protein supply variables

Name	Unit	Meta
Nutrition|Protein Supply	protein/capita/day	Total per-capita protein supply (including household wa
Nutrition|Protein Supply|+|Crops	protein/capita/day	Protein supply from crops
Nutrition|Protein Supply|+|Livestock products	protein/capita/day	Protein supply from livestock products
Nutrition|Protein Supply|+|Secondary products	protein/capita/day	Protein supply from secondary/processed products

Author(s)

Benjamin Leon Bodirsky, Kristine Karstens, Abhijeet Mishra, Florian Humpenoeder

Examples

```
  ## Not run:
    x <- reportKcal(gdx)

## End(Not run)
```

`reportRelativeHourlyLaborCosts` *reportRelativeHourlyLaborCosts*

Description

reports labor costs per ag. worker in relation to GDP pc from MAgPIE results

Usage

```
reportRelativeHourlyLaborCosts(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

labor costs per ag. worker in relation to GDP pc as MAgPIE object

Relative hourly labor cost variables

Name	Unit	Meta
Labor|Wages|Labor costs per worker relative to GDP pc	%	Agricultural labor costs relative to GDP per capita

Author(s)

Debbora Leip

Examples

```
  ## Not run:
    x <- reportRelativeHourlyLaborCosts(gdx)

## End(Not run)
```

reportResidues *reportSOM*

Description

Report soil organic carbon stock size for future MAgPIE projections

Usage

```
reportResidues(gdx)
```

Arguments

gdx	GDX file

Residue variables

Name	Unit	Meta
Resources|Carbon|Cropland|Residues	Mt C/yr	Total carbon in crop residues
Resources|Carbon|Cropland|Residues|+|Cereals	Mt C/yr	Carbon in cereal residues
Resources|Carbon|Cropland|Residues|+|Oilcrops	Mt C/yr	Carbon in oilcrop residues

Author(s)

Kristine Karstens

Examples

```
  ## Not run:
    x <- reportSOM(gdx)

## End(Not run)
```

`reportRotationLength` *reportRotationLength*

Description

reports Forest rotation length.

Usage

```
reportRotationLength(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

Forest rotation length

Rotation length variables

Name	Unit	Meta
Rotation lengths|Forestry	years	Optimal rotation length for timber plantations

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- reportRotationLength(gdx)

## End(Not run)
```

`reportRuralDemandShares` *reportRuralDemandShares*

Description

reports rural demand and production shares based on local consumption

Usage

```
reportRuralDemandShares(gdx, type = "tradOnly", level = "regglo")
```

Arguments

`gdx`	GDX file
`type`	Type of ratio that should be calculated • `all`: How much rural & trad demand as a share of all demand is satisfied locally • `tradOnly`: How much rural & trad demand as a share of rural & trad demand is satisfied locally • `potential`: How much total gridded demand is potentially satisfied by gridded production
`level`	spatial aggregation: "reg", "glo", "regglo"

Value

share of food demand at disaggregated level coming from local production as MAgPIE object

Rural demand share variables

Name	Unit	Meta
Share of Rural Demand Satisfied by Rural Production|Primary Crop and Livestock Products	0 - 1	Share of rural demand r
Share of Total Demand Satisfied by Rural Production|Primary Crop and Livestock Products	0 - 1	Share of total demand n
Share of Total Demand Potentially Satisfied by Local Production	0 - 1	Potential local productio

Author(s)

David M Chen

Examples

```
## Not run:
x <- reportruralDemandShares(gdx)

## End(Not run)
```

`reportSDG1` *reportSDG1*

Description

reports all SDG indicators relevant for SDG1 - Poverty

Usage

```
reportSDG1(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

MAgPIE object

SDG1 Poverty variables

Name	Unit	Meta
SDG|SDG01|Per-capita income	US$2017 PPP/cap/yr	GDP per capita (after shock)

Author(s)

Benjamin Bodirsky

Examples

```
## Not run:
  x <- reportSDG3(gdx)

## End(Not run)
```

reportSDG12 *reportSDG12*

Description

reports all SDG indicators relevant for SD12 - Sustainable Production and Consumption

Usage

```
reportSDG12(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

MAgPIE object

SDG12 Sustainable consumption variables

Name	Unit	Meta
SDG|SDG12|Material footprint	tDM/capita/yr	Per-capita crop demand (material footprint proxy)
SDG|SDG12|Food waste	kcal/cap/day	Per-capita daily food waste (caloric availability minus intake)
SDG|SDG12|Food waste total	Mt DM/yr	Total food waste in dry matter
SDG|SDG12|Food loss	Mt DM/yr	Food losses in supply chain (pre-consumer waste)

Author(s)

Benjamin Bodirsky

Examples

```
## Not run:
  x <- reportSDG12(gdx)

## End(Not run)
```

reportSDG15 *reportSDG15*

Description

reports all SDG indicators relevant for SD15 - Life on Land

Usage

```
reportSDG15(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

MAgPIE object

SDG15 Life on land variables

Name	Unit	Meta
SDG|SDG15|Forest share	share of total land	Share of land covered by forest (primar
SDG|SDG15|Primary forest share	share of total land	Share of land covered by primary forest
SDG|SDG15|Afforestation	million ha	Area of afforestation (NDC and additio
SDG|SDG15|Other natural land share	share of total land	Share of land covered by other natural l
SDG|SDG15|Terrestrial biodiversity	index	Biodiversity Intactness Index (BII)
SDG|SDG15|Non-agricultural land share	share of total land	Share of land not used for agriculture
SDG|SDG15|Biological nitrogen fixation on cropland	Mt N/yr	Total biological nitrogen fixation on cro
SDG|SDG15|Industrial and intentional biological fixation of N	Mt N/yr	Sum of fertilizer and crop nitrogen fixat

Author(s)

Benjamin Bodirsky, Isabelle Weindl

Examples

```
## Not run:
  x <- reportSDG15(gdx)

## End(Not run)
```

reportSDG2 *reportSDG2*

Description

reports all SDG indicators relevant for SD2 - Hunger

Usage

```
reportSDG2(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

MAgPIE object

SDG2 Hunger variables

Name	Unit	Meta
SDG|SDG02|Prevalence of underweight	million	Population with underweight BMI
SDG|SDG02|Prevalence of underweight|Children	million	Children under 5 with underweight BM
SDG|SDG02|Food availability	kcal/cap/day	Daily per-capita caloric availability
SDG|SDG02|Food expenditure share	income	Share of income spent on food (value a
SDG|SDG02|Agricultural primary product expenditure share	income	Share of income spent on agricultural p
SDG|SDG02|Agricultural commodity price index wrt 2020	1	Food price index relative to 2020 baseli
SDG|SDG02|Prevalence of obesity|Children	million	Children under 5 with obesity
SDG|SDG02|Investment in AgR&D	million US$2017/yr	Investment in agricultural research and

Author(s)

Benjamin Bodirsky

Examples

```
  ## Not run:
    x <- reportSDG2(gdx)

## End(Not run)
```

reportSDG3 *reportSDG3*

Description

reports all SDG indicators relevant for SDG3 - Health

Usage

```
reportSDG3(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

MAgPIE object

SDG3 Health variables

Name	Unit	Meta
SDG|SDG03|Prevalence of overweight	million	Population with overweight BMI
SDG|SDG03|Prevalence of obesity	million	Population with obesity
SDG|SDG03|Prevalence of overweight|Children	million	Children under 5 with overweight BMI
SDG|SDG03|Prevalence of obesity|Children	million	Children under 5 with obesity
SDG|SDG03|Consumption of alcohol	kcal/cap/day	Daily per-capita caloric intake from alcohol

Author(s)

Benjamin Bodirsky

Examples

```
## Not run:
  x <- reportSDG3(gdx)

## End(Not run)
```

reportSDG6 *reportSDG6*

Description

reports all SDG indicators relevant for SDG6 - Access to Water

Usage

```
reportSDG6(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

MAgPIE object

SDG6 Water variables

Name	Unit	Meta
SDG|SDG06|Fertilizer use	Mt N/yr	Nitrogen fertilizer application on cropland
SDG|SDG06|Nitrogen surplus on cropland	Mt N/yr	Nitrogen surplus from cropland budget
SDG|SDG06|Agricultural water use	km3/yr	Agricultural water withdrawal during growing period

Author(s)

Felicitas Beier, Isabelle Weindl

Examples

```
  ## Not run:
    x <- reportSDG6(gdx)

## End(Not run)
```

reportSDG9 *reportSDG9*

Description

reports all SDG indicators relevant for SD9 - Industrial innovation and infrastructure

Usage

```
reportSDG9(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

MAgPIE object

Author(s)

Felicitas Beier

Examples

```
  ## Not run:
    x <- reportSDG9(gdx)

## End(Not run)
```

reportSOM *reportSOM*

Description

Report soil organic carbon stock size for future MAgPIE projections

Usage

```
reportSOM(gdx, baseyear = 1995, level = "regglo")
```

Arguments

gdx	GDX file
baseyear	baseyear for calculating carbon stock change
level	aggregation level of returned data ("regglo" by default)

Actual soil carbon stock variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Stock|SOC in top 30 cm	Mt C	Total soil organic carbon stock in top 3
Resources|Soil Carbon|Actual|Stock|SOC in top 30 cm|+|Cropland Soils	Mt C	SOC stock in cropland soils
Resources|Soil Carbon|Actual|Stock|SOC in top 30 cm|+|Noncropland Soils	Mt C	SOC stock in non-cropland soils

Soil carbon stock change variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Stock Change|SOC in top 30 cm	Mt C wrt baseyear	Change in total S
Resources|Soil Carbon|Actual|Stock Change|SOC in top 30 cm|+|Cropland Soils	Mt C wrt baseyear	Change in croplar
Resources|Soil Carbon|Actual|Stock Change|SOC in top 30 cm|+|Noncropland Soils	Mt C wrt baseyear	Change in non-cr

Actual soil carbon density variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Density|SOC in top 30 cm	tC/ha	Average SOC density in top 30 cm
Resources|Soil Carbon|Actual|Density|SOC in top 30 cm|Cropland Soils	tC/ha	SOC density in cropland soils
Resources|Soil Carbon|Actual|Density|SOC in top 30 cm|Noncropland Soils	tC/ha	SOC density in non-cropland soils

Target soil carbon stock variables

Name	Unit	Meta
Resources|Soil Carbon|Target|Stock|SOC in top 30 cm	Mt C	Target total SOC stock at steady state
Resources|Soil Carbon|Target|Stock|SOC in top 30 cm|+|Cropland Soils	Mt C	Target SOC stock in cropland soils
Resources|Soil Carbon|Target|Stock|SOC in top 30 cm|+|Noncropland Soils	Mt C	Target SOC stock in non-cropland soils

Target soil carbon density variables

Name	Unit	Meta
Resources|Soil Carbon|Target|Density|SOC in top 30 cm	tC/ha	Target average SOC density at steady st
Resources|Soil Carbon|Target|Density|SOC in top 30 cm|Cropland Soils	tC/ha	Target SOC density in cropland soils
Resources|Soil Carbon|Target|Density|SOC in top 30 cm|Noncropland Soils	tC/ha	Target SOC density in non-cropland soi

Actual carbon share variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Carbon Share|SOC in top 30 cm	tC/tC	SOC relative to natural state
Resources|Soil Carbon|Actual|Carbon Share|SOC in top 30 cm|Cropland Soils	tC/tC	Cropland SOC relative to natural
Resources|Soil Carbon|Actual|Carbon Share|SOC in top 30 cm|Noncropland Soils	tC/tC	Non-cropland SOC relative to na

Target carbon share variables

Name	Unit	Meta
Resources|Soil Carbon|Target|Carbon Share|SOC in top 30 cm	tC/tC	Target SOC share at steady state
Resources|Soil Carbon|Target|Carbon Share|SOC in top 30 cm|Cropland Soils	tC/tC	Target cropland SOC share at stea
Resources|Soil Carbon|Target|Carbon Share|SOC in top 30 cm|Noncropland Soils	tC/tC	Target non-cropland SOC share a

Author(s)

Kristine Karstens

Examples

```
  ## Not run:
    x <- reportSOM(gdx)

## End(Not run)
```

reportSOM2 *reportSOM2*

Description

Report soil organic carbon stock size for future MAgPIE projections (new som realization)

Usage

```
reportSOM2(gdx, baseyear = 1995)
```

Arguments

gdx	GDX file
baseyear	baseyear for calculating carbon stock change

Actual soil carbon stock variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Stock|SOC in top 30 cm	Mt C	Actual soil organic carbon stock
Resources|Soil Carbon|Actual|Stock|SOC in top 30 cm|+|Cropland Soils	Mt C	SOC in cropland soils
Resources|Soil Carbon|Actual|Stock|SOC in top 30 cm|+|Noncropland Soils	Mt C	SOC in non-cropland soils

Target soil carbon stock variables

Name	Unit	Meta
Resources|Soil Carbon|Target|Stock|SOC in top 30 cm	Mt C	Target SOC stock at steady state
Resources|Soil Carbon|Target|Stock|SOC in top 30 cm|+|Cropland Soils	Mt C	Target SOC stock in cropland soils
Resources|Soil Carbon|Target|Stock|SOC in top 30 cm|+|Noncropland Soils	Mt C	Target SOC stock in non-cropland soils

Soil carbon stock change variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Stock Change|SOC in top 30 cm	Mt C wrt baseyear	Change in total SO
Resources|Soil Carbon|Actual|Stock Change|SOC in top 30 cm|+|Cropland Soils	Mt C wrt baseyear	Change in croplan
Resources|Soil Carbon|Actual|Stock Change|SOC in top 30 cm|+|Noncropland Soils	Mt C wrt baseyear	Change in non-cro

Soil carbon debt variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Debt|SOC in top 30 cm	Mt C	SOC debt relative to natural state
Resources|Soil Carbon|Actual|Debt|SOC in top 30 cm|+|Cropland Soils	Mt C	Cropland SOC debt relative to natural st
Resources|Soil Carbon|Actual|Debt|SOC in top 30 cm|+|Noncropland Soils	Mt C	Non-cropland SOC debt relative to natu

Actual soil carbon density variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Density|SOC in top 30 cm	tC/ha	SOC density
Resources|Soil Carbon|Actual|Density|SOC in top 30 cm|+|Cropland Soils	tC/ha	SOC density in cropland soils
Resources|Soil Carbon|Actual|Density|SOC in top 30 cm|+|Noncropland Soils	tC/ha	SOC density in non-cropland soils

Target soil carbon density variables

Name	Unit	Meta
Resources|Soil Carbon|Target|Density|SOC in top 30 cm	tC/ha	Target SOC density at steady state
Resources|Soil Carbon|Target|Density|SOC in top 30 cm|+|Cropland Soils	tC/ha	Target SOC density in cropland soils
Resources|Soil Carbon|Target|Density|SOC in top 30 cm|+|Noncropland Soils	tC/ha	Target SOC density in non-cropland s

Actual carbon share variables

Name	Unit	Meta
Resources|Soil Carbon|Actual|Carbon Share|SOC in top 30 cm	tC/tC	SOC relative to natural state
Resources|Soil Carbon|Actual|Carbon Share|SOC in top 30 cm|+|Cropland Soils	tC/tC	Cropland SOC relative to natur
Resources|Soil Carbon|Actual|Carbon Share|SOC in top 30 cm|+|Noncropland Soils	tC/tC	Non-cropland SOC relative to r

Target carbon share variables

Name	Unit	Meta
Resources|Soil Carbon|Target|Carbon Share|SOC in top 30 cm	tC/tC	Target SOC share at steady stat
Resources|Soil Carbon|Target|Carbon Share|SOC in top 30 cm|+|Cropland Soils	tC/tC	Target cropland SOC share at st

Resources|Soil Carbon|Target|Carbon Share|SOC in top 30 cm|+|Noncropland Soils	tC/tC	Target non-cropland SOC share	

Author(s)

Kristine Karstens

Examples

```
  ## Not run:
    x <- reportSOM2(gdx)

## End(Not run)
```

`reportTau` *reportTau*

Description

reports Tau

Usage

```
reportTau(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

tau values as MAgPIE object (Index)

Tau variables

Name	Unit	Meta
Productivity|Landuse Intensity Indicator Tau	Index	Agricultural land-use intensity indicator for crops

Author(s)

Florian Humpenoeder, Patrick v. Jeetze

Examples

```
## Not run:
x <- reportTau(gdx)

## End(Not run)
```

`reportTc` *reportTc*

Description

reports Tc

Usage

```
reportTc(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	Aggregation level of the returned Tc report

Value

tc values as MAgPIE object (%/yr)

Technological change variables

Name	Unit	Meta
Productivity|Yield-increasing technological change crops	%/yr	Annual rate of yield-increasing technological change for c

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
  x <- reportTc(gdx)

## End(Not run)
```

`reportTimber` *reportTimber*

Description

reports MAgPIE demand for timber.

Usage

```
reportTimber(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

Timber demand

Timber demand variables

Name	Unit	Meta
Timber|Volumetric|Demand|+|Roundwood	Mm3/yr	Total roundwood demand
Timber|Volumetric|Demand|Roundwood|+|Industrial roundwood	Mm3/yr	Demand for industrial roundwood
Timber|Volumetric|Demand|Roundwood|+|Wood fuel	Mm3/yr	Demand for wood fuel
Timber|Volumetric|Demand|Roundwood|+|Timber for construction	Mm3/yr	Demand for timber used in construction

Timber production variables

Name	Unit	Meta
Timber|Volumetric|Production|+|Roundwood	Mm3/yr	Total roundwood production
Timber|Volumetric|Production|Roundwood|+|Industrial roundwood	Mm3/yr	Production of industrial roundwood
Timber|Volumetric|Production|Roundwood|+|Wood fuel	Mm3/yr	Production of wood fuel
Timber|Volumetric|Production|Roundwood|+|Timber for construction	Mm3/yr	Production of timber for construction

Timber trade variables

Name	Unit	Meta
Timber|Volumetric|Net-Trade|+|Roundwood	Mm3/yr	Net export of roundwood
Timber|Volumetric|Net-Trade|Roundwood|+|Industrial roundwood	Mm3/yr	Net export of industrial roundwood
Timber|Volumetric|Net-Trade|Roundwood|+|Wood fuel	Mm3/yr	Net export of wood fuel
Timber|Volumetric|Net-Trade|Roundwood|+|Timber for construction	Mm3/yr	Net export of timber for construction
Timber|Volumetric|Exports|+|Roundwood	Mm3/yr	Gross exports of roundwood
Timber|Volumetric|Exports|Roundwood|+|Industrial roundwood	Mm3/yr	Gross exports of industrial roundwood
Timber|Volumetric|Exports|Roundwood|+|Wood fuel	Mm3/yr	Gross exports of wood fuel
Timber|Volumetric|Exports|Roundwood|+|Timber for construction	Mm3/yr	Gross exports of timber for construction
Timber|Volumetric|Imports|+|Roundwood	Mm3/yr	Gross imports of roundwood
Timber|Volumetric|Imports|Roundwood|+|Industrial roundwood	Mm3/yr	Gross imports of industrial roundwood
Timber|Volumetric|Imports|Roundwood|+|Wood fuel	Mm3/yr	Gross imports of wood fuel
Timber|Volumetric|Imports|Roundwood|+|Timber for construction	Mm3/yr	Gross imports of timber for construction

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- reportTimber(gdx)

## End(Not run)
```

---

reportTimberDemand *reportTimberDemand*

Description

reports MAgPIE demand for timber in tDM (complementing `reportTimber` which reports in Mm3).

Usage

```
reportTimberDemand(gdx)
```

Arguments

gdx	GDX file

Value

Timber demand

Timber demand variables

Name	Unit	Meta
Timber demand|Roundwood	mio tDM	Total roundwood demand
Timber demand|Industrial roundwood	mio tDM	Industrial wood demand
Timber demand|Wood fuel	mio tDM	Wood fuel demand

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- reportTimberDemand(gdx)

## End(Not run)
```

---

reportTotalHoursWorked

reportTotalHoursWorked

Description

reports total hours worked in crop+livestock production (and maccs) from MAgPIE results

Usage

```
reportTotalHoursWorked(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	spatial aggregation: "reg", "glo", "regglo"

Value

total hours worked as MAgPIE object

Total hours worked variables

Name	Unit	Meta
Labor|Total Hours Worked	mio h	Total hours worked in crop and livestock production

Author(s)

Debbora Leip

Examples

```
  ## Not run:
    x <- reportTotalHoursWorked(gdx)

## End(Not run)
```

`reportTrade` *reportTrade*

Description

reports trade

Usage

```
reportTrade(gdx, detail = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`detail`	if true, provides estimates for all commodities, otherwise aggregates some groups
`level`	The aggregation level of the trade reporting

Value

Net-Exports and self sufficiency (exports/domestic supply) as MAgPIE object. Unit: see names

Net trade variables

Name	Unit	Meta
Trade|Net-Trade|+|Crops	Mt DM/yr	Net export of crops (positive = net exporter)
Trade|Net-Trade|+|Livestock products	Mt DM/yr	Net export of livestock products (excluding fish)
Trade|Net-Trade|+|Secondary products	Mt DM/yr	Net export of secondary products
Trade|Net-Trade|+|Bioenergy crops	Mt DM/yr	Net export of bioenergy crops

Gross trade variables

Name	Unit	Meta
Trade|Exports|+|Crops	Mt DM/yr	Gross exports of crops
Trade|Exports|+|Livestock products	Mt DM/yr	Gross exports of livestock products
Trade|Imports|+|Crops	Mt DM/yr	Gross imports of crops
Trade|Imports|+|Livestock products	Mt DM/yr	Gross imports of livestock products

Self-sufficiency variables

Name	Unit	Meta
Trade|Self-sufficiency|+|Crops	1	Self-sufficiency ratio for crops (production/domestic supply)
Trade|Self-sufficiency|+|Livestock products	1	Self-sufficiency ratio for livestock products

Author(s)

Benjamin Leon Bodirsky, Mishko Stevanovic

Examples

```
## Not run:
  x <- reportTrade(gdx="fulldata.gdx",detail=TRUE)

## End(Not run)
```

reportTradeGrowth *reportTradeGrowth*

Description

reports trade growth rate

Usage

```
reportTradeGrowth(gdx, detail = FALSE)
```

Arguments

gdx	GDX file
detail	if true, provides results for all commodities, otherwhise aggregates some groups

Value

Trade growth rates (index)

Trade growth variables

Name	Unit	Meta
Trade|Trade Growth Rate	Index	Trade volume index relative to base year
Trade|Trade Growth Rate|+|Crop products	Index	Crop trade growth index
Trade|Trade Growth Rate|+|Livestock products	Index	Livestock trade growth index

Author(s)

Xiaoxi Wang

Examples

```
  ## Not run:
    x <- reportTradeGrowth(gdx="fulldata.gdx",detail=TRUE)

## End(Not run)
```

reportValueMaterialDemand *reportValueMaterialDemand*

Description

reports value of material demand

Usage

```
reportValueMaterialDemand(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

magpie object

Material demand value variables

Name	Unit	Meta
Value|Bioeconomy Demand	million US$2017/yr	Total value of bioeconomy material demand
Value|Bioeconomy Demand|+|Crops	million US$2017/yr	Value of crop products for material demand
Value|Bioeconomy Demand|+|Residues	million US$2017/yr	Value of residues for material demand

Author(s)

David Chen

Examples

```
## Not run:
x <- reportValueMaterialDemand(gdx)

## End(Not run)
```

reportValueTrade *reportValueTrade*

Description

reports trade value

Usage

```
reportValueTrade(gdx, detail = FALSE, level = "regglo")
```

Arguments

gdx	GDX file
detail	if true, provides estimates for all commodities, otherwhise aggregates some groups
level	aggregation level of returned data ("regglo" by default)

Value

trade value as MAgPIE object Unit: see names

Trade value variables

Name	Unit	Meta
Trade Value|Net-Exports	million US$2017/yr	Net trade value (exports minus imports)
Trade Value|Exports	million US$2017/yr	Gross export value
Trade Value|Imports	million US$2017/yr	Gross import value

Author(s)

Florian Humpenoeder

Examples

```
## Not run:
  x <- reportValueTrade(gdx)

## End(Not run)
```

reportVegfruitShare *reportVegfruitShare*

Description

reports the share of livestock products (including fish) in total calorie food supply

Usage

```
reportVegfruitShare(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	An aggregation level for the spatial dimension. Can be any level available via superAggregateX.

Value

per-capita calories as MAgPIE object (kcal/cap/day)

Vegfruit share variables

Name	Unit	Meta
Nutrition|Dietary Composition|Vegetables Fruits Nuts Share	kcal/kcal	Share of vegetables, fruits, and nuts in total calorie

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- reportLivestockShare(gdx)

## End(Not run)
```

`reportWageDevelopment` *reportWageDevelopment*

Description

reports indicator on wage development: hourly labor costs in each time step relative to hourly labor costs in 2000

Usage

```
reportWageDevelopment(gdx, baseYear = 2000, level = "regglo")
```

Arguments

`gdx`	GDX file
`baseYear`	year relative to which the wage development should be calculated
`level`	spatial aggregation: "reg", "glo", "regglo"

Value

indicator on wage development as MAgPIE object

Wage development variables

Name	Unit	Meta
Labor|Wages|Hourly labor costs relative to 2000	index	Hourly labor cost development relative to base year

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- reportWageDevelopment(gdx)

## End(Not run)
```

reportWaterAvailability
reportWaterAvailability

Description

reports water availability

Usage

```
reportWaterAvailability(gdx, level = "regglo")
```

Arguments

gdx	GDX file
level	aggregation level of returned data ("regglo" by default)

Value

water availability as MAgPIE object Unit: see names

Water availability variables

Name	Unit	Meta
Resources|Water|Availability|Agriculture	km3/yr	Water available for agricultural use

Author(s)

Felicitas Beier

Examples

```
## Not run:
  x <- reportWaterAvailability(gdx)

## End(Not run)
```

`reportWaterIndicators` *reportWaterIndicators*

Description

reports a set of water indicators

Usage

```
reportWaterIndicators(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	aggregation level of returned data ("regglo" by default)

Value

MAgPIE object

Water indicator variables

Name	Unit	Meta
Water|Environmental flow violation volume	km3/yr	Volume of environmental flow
Water|Environmental flow violation share of total water withdrawals	share	EFV share of human water wit
Water|Environmental flow violation share of water availability	share	EFV share of water availability
Water|Irrigated Area suffering under Environmental Flow Violation	Mha	Area in clusters with EFV
Water|Share of total Irrigated Area suffering from Environmental Flow Violations	share	Share of irrigated area in EFV
Water|Withdrawal to Availability ratio	fraction	Water stress indicator

Author(s)

Felicitas Beier

Examples

```
## Not run:
x <- reportWaterIndicators(gdx)

## End(Not run)
```

reportWaterUsage *reportWaterUsage*

Description

reports water usage for agricultural sector, crops and livestock and non-agricultural sector

Usage

```
reportWaterUsage(gdx, detail = TRUE, level = "regglo")
```

Arguments

gdx	GDX file
detail	logical. Setting to FALSE reports for agricultural sector, TRUE reports for combined, crops and livestock separately
level	aggregation level of returned data ("regglo" by default)

Value

water usage as MAgPIE object Unit: see names

Agricultural water withdrawal variables

Name	Unit	Meta
Resources|Water|Withdrawal|Agriculture	km3/yr	Total agricultural water withdrawal
Resources|Water|Withdrawal|Agriculture|Crops	km3/yr	Water withdrawal for crop irrigation
Resources|Water|Withdrawal|Agriculture|Crops|+|Crops	km3/yr	Water withdrawal for food and feed crop
Resources|Water|Withdrawal|Agriculture|Crops|+|Bioenergy crops	km3/yr	Water withdrawal for bioenergy crops
Resources|Water|Withdrawal|Agriculture|Livestock|+|Livestock products	km3/yr	Water withdrawal for livestock

Agricultural water consumption variables

Name	Unit	Meta
Resources|Water|Consumption|Agriculture	km3/yr	Total agricultural water consumption
Resources|Water|Consumption|Agriculture|Crops	km3/yr	Water consumption for crop irrigation
Resources|Water|Consumption|Agriculture|Livestock	km3/yr	Water consumption for livestock

Non-agricultural water variables

Name	Unit	Meta
Resources|Water|Withdrawal|Non-agriculture	km3/yr	Total non-agricultural water withdrawal
Resources|Water|Withdrawal|Non-agriculture|+|domestic	km3/yr	Water withdrawal for domestic use
Resources|Water|Withdrawal|Non-agriculture|+|manufacturing	km3/yr	Water withdrawal for manufacturing sector
Resources|Water|Withdrawal|Non-agriculture|+|electricity	km3/yr	Water withdrawal for electricity generation
Resources|Water|Consumption|Non-agriculture	km3/yr	Total non-agricultural water consumption

Author(s)

Florian Humpenoeder, Vartika Singh, Miodrag Stevanovic, Felicitas Beier

Examples

```
## Not run:
x <- reportWaterUsage(gdx)

## End(Not run)
```

`reportWoodFuel` *reportWoodFuel*

Description

Reports wood fuel demand from a MAgPIE GDX file at ISO country level and converts from volumetric units (Mm³) to energy (PJ).

Usage

```
reportWoodFuel(gdx, file = NULL)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie

Value

MAgPIE object with wood fuel demand at ISO level in PJ. Dimension 1: ISO country, Dimension 2: year, Dimension 3: "woodfuel"

Author(s)

Kristine Karstens

Examples

```
## Not run:
  x <- reportWoodFuel(gdx)

## End(Not run)
```

`reportWorkingAgePopulation`
reportWorkingAgePopulation

Description

reports working age population

Usage

```
reportWorkingAgePopulation(gdx, level = "regglo")
```

Arguments

`gdx`	GDX file
`level`	spatial aggregation: "reg", "glo", "regglo", "iso"

Value

working age population as MAgPIE object

Working age population variables

Name	Unit	Meta
Working age population	million people	Population aged 15-64 years

Author(s)

Debbora Leip

Examples

```
  ## Not run:
    x <- reportWorkingAgePopulation(gdx)

## End(Not run)
```

`reportYields` *reportYields*

Description

reports yields

Usage

```
reportYields(gdx, detail = FALSE, physical = TRUE, level = "regglo")
```

Arguments

`gdx`	GDX file
`detail`	if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
`physical`	if true (default) physical area (croparea) used for yield calculation; if false harvested area used for yield calculation
`level`	aggregation level of returned data ("regglo" by default)

Details

Realized yields, weighted by the endogenous cropping pattern. The `Productivity|Yields|Input data|*` variables hold the 1995 cropping pattern fixed instead, so the ratio between the two families is not a clean yield gap.

Value

yield as MAgPIE object (t DM/ha)

Yield by physical area variables

Reported when `physical = TRUE` (the default). 22 variables at `detail = FALSE`, 76 at `detail = TRUE`. The stem itself is not reported, and no name carries a + summation marker - unlike `reportYieldsCropRaw` and `reportYieldsCropCalib`, which do emit them.

Name	Unit	Meta
Productivity|Yields|Yield by physical area|Crops	t DM/ha	Yield of all crops
Productivity|Yields|Yield by physical area|Crops|Cereals	t DM/ha	Cereals (maize, rice, temperate cereals, tropical cer

Productivity|Yields|Yield by physical area|Crops|Oil crops	t DM/ha	Oil crops (cotton seed, groundnuts, oilpalms, other
Productivity|Yields|Yield by physical area|Crops|Sugar crops	t DM/ha	Sugar crops (sugar beet, sugar cane)
Productivity|Yields|Yield by physical area|Crops|Other crops	t DM/ha	Other crops (fruits, vegetables, nuts, potatoes, pulse
Productivity|Yields|Yield by physical area|Bioenergy crops	t DM/ha	Second-generation bioenergy crops
Productivity|Yields|Yield by physical area|Forage	t DM/ha	Forage crops
Productivity|Yields|Yield by physical area|Pasture	t DM/ha	Pasture biomass; no water split

Every entry except Pasture is also reported split by water supply, as a lowercase leaf (e.g. \\Crops\\irrigated) rather than a top-level branch. With `detail = TRUE` each group additionally gains its individual crops on the same pattern.

Yield by harvested area variables

Reported when `physical = FALSE`. Same tree and counts, under the stem Productivity\|Yields\|Yield by harvested area. Harvested area is physical area scaled by the regional multicropping index `f18_multicropping`, so these yields differ from those above by cropping intensity.

Author(s)

Florian Humpenoeder, Xiaoxi Wang, Kristine Karstens, Abhijeet Mishra, Felicitas Beier

Examples

```
## Not run:
x <- reportYields(gdx)

## End(Not run)
```

`reportYieldsCropCalib` *reportYieldsCropCalib*

Description

reports potential yields after calibration

Usage

```
reportYieldsCropCalib(gdx, detail = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`detail`	if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
`level`	aggregation level of returned data ("regglo" by default)

Details

`Calibrated` is the crop-model input after management calibration but before technological change; `Calibrated|Including technological change` additionally applies tau and equals `vm_yld`. Both are weighted with the 1995 cropping pattern held fixed.

Value

yield as MAgPIE object (t DM/ha)

Calibrated input-data yield variables

Name	Unit	Meta
Productivity|Yields|Input data|Calibrated	t DM/ha	Potential crop yields after calibration, b
Productivity|Yields|Input data|Calibrated|+|Cereals	t DM/ha	Calibrated cereal yields
Productivity|Yields|Input data|Calibrated|Including technological change	t DM/ha	Calibrated potential yields with the tau

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
x <- reportYieldsCropCalib(gdx)

## End(Not run)
```

`reportYieldsCropRaw` *reportYieldsCropRaw*

Description

reports potential yields before calibration

Usage

```
reportYieldsCropRaw(gdx, detail = FALSE, level = "regglo")
```

Arguments

`gdx`	GDX file
`detail`	if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops")
`level`	aggregation level of returned data ("regglo" by default)

Details

Uncalibrated potential yields as they come from the crop model, before management calibration and technological change. Weighted with the 1995 cropping pattern held fixed.

Value

yield as MAgPIE object (t DM/ha)

Uncalibrated input-data yield variables

Name	Unit	Meta
Productivity|Yields|Input data|Uncalibrated	t DM/ha	Potential crop yields before calibration
Productivity|Yields|Input data|Uncalibrated|+|Cereals	t DM/ha	Uncalibrated cereal yields

Author(s)

Edna J. Molina Bacca

Examples

```
## Not run:
x <- reportYieldsCropRaw(gdx)

## End(Not run)
```

---

reportYieldShifter *reportYieldShifter*

Description

Reports the Crop model input yield shifter

Usage

```
reportYieldShifter(
  gdx,
  file = NULL,
  level = "reg",
  baseyear = "y2000",
  relative = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
baseyear	baseyear for the yield shifter. Also fixes land patterns for aggregation to baseyear.
relative	relative or absolute changes to baseyear

Value

crop yield as MAgPIE object (unit depends on attributes)

Yield shifter variables

Name	Unit	Meta
Productivity|Climate Change Yield Shifter	Index baseyear=1	Climate-induced yield changes relative to b
Productivity|Climate Change Yield Shifter|Cereals|Rainfed	Index baseyear=1	Yield shifter for rainfed cereals
Productivity|Climate Change Yield Shifter|Cereals|Irrigated	Index baseyear=1	Yield shifter for irrigated cereals

Author(s)

Benjamin Leon Bodirsky

See Also

reportYieldShifter

Examples

```
  ## Not run:
    x <- reportYieldShifter(gdx)

## End(Not run)
```

ResidueBiomass *ResidueBiomass*

Description

reads Crop Residue Biomass out of a MAgPIE gdx file

Usage

```
ResidueBiomass(
  gdx,
  level = "reg",
  products = "kcr",
  product_aggr = FALSE,
  attributes = "dm",
  water_aggr = TRUE,
  plantpart = "both"
)
```

Arguments

gdx	GDX file
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not. Usually boolean, but here also the value "kres" is allowed, which provides kcr aggregated to kres
attributes	dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector.
water_aggr	aggregate irrigated and non-irriagted production or not (boolean).
plantpart	both ag or bg

Value

production as MAgPIE object (unit depends on attributes)

Author(s)

Benjamin Leon Bodirsky

See Also

reportProduction, demand

Examples

```
## Not run:
x <- production(gdx)

## End(Not run)
```

`Residues` *Residues*

Description

reads various crop residue (carbon) outputs out of a MAgPIE gdx file

Usage

```
Residues(
  gdx,
  level = "regglo",
  products = "kres",
  waterAggr = TRUE,
  output = "all"
)
```

Arguments

`gdx`	GDX file
`level`	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global)
`products`	Selection of products (either "kcr" or "kres")
`waterAggr`	Aggregate irrigated and non-irriagted production or not (boolean).
`output`	Switch between different outputs: "biomass", "fieldBalance", "resDemand", all

Value

production as MAgPIE object (unit depends on attributes)

Author(s)

Kristine Karstens, Michael Crawford

See Also

`ResidueBiomass`

Examples

```
  ## Not run:
    x <- Residues(gdx)

## End(Not run)
```

ResidueUsage *ResidueUsage*

Description

reads Crop Residue Usage out of a MAgPIE gdx file

Usage

```
ResidueUsage(
  gdx,
  level = "reg",
  products = "kcr",
  product_aggr = FALSE,
  attributes = "dm",
  water_aggr = TRUE
)
```

Arguments

gdx	GDX file
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not. Usually boolean, but here also the value "kres" is allowed, which provides kcr aggregated to kres
attributes	dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector.
water_aggr	aggregate irrigated and non-irriagted production or not (boolean).

Value

production as MAgPIE object (unit depends on attributes)

Author(s)

Kristine Karstens, Michael Crawford

See Also

ResidueBiomass

Examples

```
## Not run:
x <- ResidueUsage(gdx)

## End(Not run)
```

RotationLength *RotationLength*

Description

reads rotation length out of a MAgPIE gdx file

Usage

```
RotationLength(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate

Details

Forest rotation length

Value

Forest rotation length

Author(s)

Abhijeet Mishra

Examples

```
## Not run:
  x <- RotationLength(gdx)

## End(Not run)
```

ruralDemandShares *ruralDemandShares*

Description

reports rural demand shares based on local consumption

Usage

```
ruralDemandShares(
  gdx,
  type = "tradOnly",
  level = "reg",
  product_aggr = TRUE,
  file = NULL
)
```

Arguments

gdx	GDX file
type	Type of ratio that should be calculated • all: How much rural & trad demand as a share of all demand is satisfied locally • tradOnly: How much rural & trad demand as a share of rural & trad demand is satisfied locally • potential: How much total gridded demand is potentially satisfied by gridded production
level	spatial aggregation to report employment ("reg", "glo" or "regglo")
product_aggr	sum over products if TRUE
file	a file name the output should be written to using write.magpie

Value

share of food consumed locally

Author(s)

David M Chen

Examples

```
## Not run:
x <- localDemandShares(gdx)

## End(Not run)
```

Seed *Seed*

Description

Calculates MAgPIE demand for Seed out of a gdx file

Usage

```
Seed(gdx, level = "reg", attributes = "dm")
```

Arguments

gdx	GDX file
level	Level of regional aggregation ("reg", "glo", "regglo")
attributes	dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector.

Details

Demand definitions are equivalent to FAO CBS categories

Value

demand as MAgPIE object (Unit depends on attributes)

Author(s)

Benjamin Leon Bodirsky

Examples

```
  ## Not run:
    x <- demand(level="regglo", products="kcr")

## End(Not run)
```

SOM *SOM*

Description

Calculates soil organic carbon stock size based on a MAgPIE gdx file

Usage

```
SOM(
  gdx,
  file = NULL,
  type = "stock",
  reference = "actual",
  level = "reg",
  noncrop_aggr = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
type	"stock" (default) for absoulte values, "density" for per hectar values
reference	default is "actual" (cshare in actual carbon stocks). Other option is "target" (cshare in target carbon stocks).
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
noncrop_aggr	aggregate non cropland types to 'noncropland' (if FALSE all land types of pools59 will be reported)

Value

A MAgPIE object containing som values

Author(s)

Kristine Karstens

Examples

```
## Not run:
x <- SOM(gdx)

## End(Not run)
```

SOM2 *SOM2*

Description

Calculates soil organic carbon stock size based on a MAgPIE gdx file (for threepool realization)

Usage

```
SOM2(gdx, type = "stock", level = "regglo", noncropAggr = TRUE)
```

Arguments

gdx	GDX file
type	"stock" (default) for absoulte values, "density" for per hectar values
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global)
noncropAggr	aggregate non cropland types to 'noncropland' (if FALSE all land types of pools59 will be reported)

Value

A MAgPIE object containing som values

Author(s)

Kristine Karstens

Examples

```
## Not run:
  x <- SOM2(gdx)

## End(Not run)
```

submitCalibration *submitCalibration*

Description

Submits Calibration Factors of current run to calibration archive. Currently covers calibration factors for yields and land conversion costs. This is useful to make runs more comparable to each other. The function can be also used as part of a script running a collection of runs.

Usage

```
submitCalibration(
  name,
  file = c("modules/14_yields/input/f14_yld_calib.csv",
    "modules/39_landconversion/input/f39_calib.cs3",
    "modules/39_landconversion/input/f39_calib_past.csv"),
  archive = "/p/projects/landuse/data/input/calibration"
)
```

Arguments

name	name under which the calibration should be stored. Should be as self-explaining as possible. The total file name has the format calibration_<name>_<date>.tgz.
file	path to a f14_yld_calib.csv, f39_calib.cs3 (older version f39_calib.csv) and f39_calib_past.csv file (in this order; the third is optional). Alternatively a fulldata.gdx file containing the calibration factors can be used. Supported file types are "csv", "cs3" and "gdx".
archive	path to the archive the calibration factors should be stored

Value

file name of the stored calibration factors (useful for scripts in which you might want to re-use a calibration setting at a later stage again)

Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Patrick v. Jeetze

Examples

```
## Not run:
fname <- submitCalibration("TestCalibration", file = "fulldata.gdx")

## End(Not run)
```

---

superAggregateX *superAggregateX*

Description

drop-in replacement for superAggregate based on toolAggregate

Usage

```
superAggregateX(
  data,
  aggr_type,
  level = "reg",
  weight = NULL,
  crop_aggr = FALSE
)
```

Arguments

data	A MAgPIE object
aggr_type	Aggregation Type. Can be any function for one or two objects (data and weight) of the same size. Currently pre-supported functions: "sum","mean","weighted_mean".
level	Either a level or the name of a mapping file. Allowed level types are global "glo", regional "reg" and "regglo". The mapping file can only map from regions to other regions.
weight	Currently only used for weighted_mean
crop_aggr	determines whether output should be crop-specific (FALSE) or aggregated over all crops (TRUE). The method used for aggregation is set by aggr_type

Value

returns a MAgPIE object.

Author(s)

Jan Philipp Dietrich

See Also

Other Spatial: addGeometry(), clusterOutputToTerraVector(), gdxAggregate(), mappingToLongFormat()

---

surplusChange *surplusChange*

Description

calculates aggregate change in economic surplus in mio.US$ based on a MAgPIE gdx files from two different scenarios.

Usage

```
surplusChange(gdx1, gdx2, file = NULL, level = "reg", type = "consumer")
```

Arguments

gdx1	GDX file from benchmark scenario
gdx2	GDX file from the analyzed scenario
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
type	Economic surplus type: "consumer" (default), "producer" or "welfare"

Value

A MAgPIE object containing aggregate changes in producer surplus, consumer surplus and aggregate economic welfare between an analyzed scenario and a benchmark scenario, in million $US.

Author(s)

Miodrag Stevanovic

Examples

```
  ## Not run:
    x <- surplusChange(gdx1, gdx2)

## End(Not run)
```

---

tau *tau*

Description

Calculates Landuse intensity indicator tau based on a MAgPIE gdx file

Usage

```
tau(
  gdx,
  file = NULL,
  level = "reg",
  start_value = FALSE,
  digits = 4,
  prev_year = "y1985",
  type = "crop"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregateX
start_value	If TRUE, the initial values are added under the year prev_year
digits	The result will be rounded to this number of digits
prev_year	Year to store the initialization tau information in
type	currently only "crop"

Value

A MAgPIE object containing tau values (index)

Author(s)

Jan Philipp Dietrich, Patrick v. Jeetze

Examples

```
## Not run:
x <- tau(gdx)

## End(Not run)
```

---

taxRevenueRotations *taxRevenueRotations*

Description

calculates taxes of crop rotations as difference between the selected scenario and the baseline scenario that shall capture the internalized incentives for crop rotations.

Usage

```
taxRevenueRotations(
  gdx,
  file = NULL,
  level = "regglo",
  penalty = "onlyTaxRevenue"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	aggregation level, reg, glo or regglo
penalty	"OnlyTaxRevenue" provides the tax Revenues from a rotation tax/subsidy. "OnlyInternalizedServices" provides the penalty by foregone Ecosystem Services, the part of the externality which is internalized by the farmer independent of the tax. "FullPenalty" provides the sum of both, which is what the model sees.

Author(s)

Benjamin Leon Bodirsky

Examples

```
## Not run:
x <- wageRent(gdx)

## End(Not run)
```

---

tc *tc*

Description

Calculates TC rates based on a MAgPIE gdx file

Usage

```
tc(
  gdx,
  file = NULL,
  level = "reg",
  annual = TRUE,
  avrg = FALSE,
  baseyear = 1995,
  type = "crop"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

annual	If TRUE, annual values are reported. If FALSE, the values for the whole timestep are reported. If FALSE, avrg has no effect
avrg	If FALSE the annual tc rates of the current period are returned, otherwise the average annual tc rate for the period tbase to tn is returned. tbase defaults to the first timestep (see baseyear)
baseyear	Determines the base year timestep for annual tc calculation. Average tc rates for later timesteps are calculated with respect to baseyear. No tc rates for timesteps before baseyear are returned)
type	currently only 'crop'

Value

A MAgPIE object containing tc rates. Annual ones if annual=TRUE, for the whole timestep if annual=FALSE.

Author(s)

Jan Philipp Dietrich

Examples

```
## Not run:
x <- tc(gdx)

## End(Not run)
```

Timber *Timber*

Description

reads timber demand out of a MAgPIE gdx file

Usage

```
Timber(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "regglo" (regional and global), custom region aggregation or any secdforest aggregation level defined in superAggregateX

Details

Forest demandfor timber production

Value

Forest demandfor timber production

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- Timber(gdx)

## End(Not run)
```

TimberDemand *TimberDemand*

Description

reads timber demand out of a MAgPIE gdx file

Usage

```
TimberDemand(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate

Details

Forest demandfor timber production

Value

Forest demandfor timber production

Author(s)

Abhijeet Mishra

Examples

```
  ## Not run:
    x <- TimberDemand(gdx)

## End(Not run)
```

TimberProductionVolumetric
TimberProductionVolumetric

Description

reads timber production out of a MAgPIE gdx file

Usage

```
TimberProductionVolumetric(
  gdx,
  file = NULL,
  level = "regglo",
  sumProduct = FALSE,
  sumSource = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate
sumProduct	sum over wood and woodfuel (TRUE/FALSE)
sumSource	sum over timber sources: timber plantations, primary forest, secondary forest and non-forest land (woodfuel only) (TRUE/FALSE)

Details

Annual timber production from timber plantations, primary forest, secondary forest and non-forest land (woodfuel only). Converted from mio. ton DM per year to mio. m3 per year using volumetric conversion factors.

Value

Timber production in mio. m3 per year

Author(s)

Abhijeet Mishra, Florian Humpenoeder

Examples

```
## Not run:
x <- TimberProductionVolumetric(gdx)

## End(Not run)
```

`timePeriods` *timePeriods*

Description

Calculates MAgPIE time period lengths between each two timesteps

Usage

```
timePeriods(gdx)
```

Arguments

gdx	GDX file

Value

magpie time periods as MAgPIE object as a number of years

Author(s)

Mishko Stevanovic

Examples

```
  ## Not run:
    x <- timePeriods(gdx=gdx)

## End(Not run)
```

totalHoursWorked *totalHoursWorked*

Description

returns total hours worked per year in crop+livestock production from MAgPIE results

Usage

```
totalHoursWorked(gdx, level = "reg", file = NULL)
```

Arguments

gdx	GDX file
level	spatial aggregation to report employment ("reg", "glo", or "regglo")
file	a file name the output should be written to using write.magpie

Value

total hours worked in agriculture per year

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- totalHoursWorked(gdx)

## End(Not run)
```

trade *trade*

Description

Calculates MAgPIE trade or self-sufficiencies out of a gdx file

Usage

```
trade(
  gdx,
  file = NULL,
  level = "reg",
  products = "k_trade",
  productAggr = FALSE,
  attributes = "dm",
  weight = FALSE,
  relative = FALSE,
  type = "net-exports"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation ("reg", "glo", "regglo", or name of custom mapping)
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
productAggr	aggregate over products or not (boolean)
attributes	dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector.
weight	in case relative=T also the weighting for the self sufficiencies is provided as it is an intensive parameter
relative	if relative=TRUE, self sufficiencies are reported, so the amount of production divided by domestic demand
type	exports-imports ("net-exports"), gross imports ("imports") or gross exports ("exports"), in the bilateral case we report a few others given balanceflow: ; only valid if relative=FALSE

Details

Trade definitions are equivalent to FAO CBS categories

Value

trade (production-demand) as MAgPIE object; unit depends on attributes

Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder, Mishko Stevanovic

Examples

```
  ## Not run:
    x <- trade(gdx="fulldata.gdx", level="regglo", products="kcr")

## End(Not run)
```

`tradedPrimaries` *tradedPrimaries*

Description

Converts trade flows into primary product equivalents by tracing secondary and livestock products back to their primary inputs. Supports both net trade flows and bilateral trade matrices. Trade flows are decomposed into three pathways: (1) direct primary trade, (2) primaries embodied in secondary products, and (3) primaries needed as feed for livestock.

Usage

```
tradedPrimaries(
  gdx,
  file = NULL,
  bilateral = NULL,
  convFactor = "exporter",
  kastner = TRUE,
  level = "reg",
  disaggLivestock = FALSE,
  selfselfTrade = TRUE
)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`bilateral`	Logical or NULL. If NULL (default), auto-detects: uses bilateral trade matrix (ov21_trade) if present, otherwise falls back to net trade. If TRUE, requires bilateral data and warns if not found before falling back. If FALSE, always uses net trade flows.
`convFactor`	Character. When using bilateral trade, determines whether processing shares and feed baskets are taken from "exporter" or "importer" region. Default is "exporter" (footprint allocation to producing region).
`kastner`	Logical. If TRUE and bilateral=TRUE, applies Kastner et al. 2011 adjustment to bilateral trade matrix. Default is TRUE.
`level`	Level of regional aggregation ("reg", "glo", "regglo", or custom mapping)

disaggLivestock
: Logical. If TRUE, the feed pathway retains the livestock product dimension so that feed crop demands can be traced back to specific animal products. Output dim 3 will be kli_product.kve_product for the feed pathway instead of the aggregated feed.kve_product. The prim and secd pathways are unchanged (pathway.kve_product). Default is FALSE.

selfselfTrade
: if TRUE makes the self.self trade diagonal the amount from trade matrix, as opposed to the domestic consumption diagonal of Kastner matrix

Details

When convFactor="exporter", the exporting region's processing pathways and feed baskets are used, which is appropriate for production-based footprint accounting. When convFactor="importer", the importing region's factors are used, which maintains backward compatibility with the original tradedPrimaries function.

For bilateral trade, the output includes full origin-destination detail. For net trade, the output is aggregated by region.

When disaggLivestock=TRUE, callers can assign embodied resources either to the feed crop (collapse dim 3.1, i.e. the livestock dimension) or to the animal product (collapse dim 3.2, i.e. the crop dimension).

Value

MAgPIE object with primary product trade equivalents in dry matter (tDM). For bilateral trade: dimensions are (exporter.importer, year, pathway.product) For net trade: dimensions are (region, year, pathway.product) When disaggLivestock=TRUE: feed items have dimensions kli_product.kve_product instead of feed.kve_product; prim/secd items are unchanged.

Author(s)

David M Chen, Kristine Karstens

Examples

```
## Not run:
  # Bilateral trade with exporter's factors (production-based accounting)
  x <- tradedPrimaries(gdx, bilateral = TRUE, convFactor = "exporter")

  # Bilateral trade with importer's factors (consumption-based accounting)
  x <- tradedPrimaries(gdx, bilateral = TRUE, convFactor = "importer")

  # Net trade (backward compatible with tradedPrimaries)
  x <- tradedPrimaries(gdx, bilateral = FALSE)

## End(Not run)
```

tradeKastner *tradeKastner*

Description

Implements Kastner et al. 2011 (DOI: 10.1016/j.ecolecon.2011.01.012) on the MAgPIE bilateral trade matrix. Adjusts the trade matrix based on a assumption of proportionality such that intermediate trade partners (importing to re-export) are removed from the matrix.

Usage

```
tradeKastner(
  gdx,
  trade,
  level = "reg",
  products = "kall",
  attributes = "dm",
  selfselfTrade = TRUE
)
```

Arguments

gdx	GDX file to read from (for production data)
trade	Bilateral trade data (ov21_trade or other)
level	Level of regional aggregation ("reg", "glo", "regglo", or custom mapping)
products	Selection of products (e.g. "kall", "kcr", "kli")
attributes	dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm")
selfselfTrade	if TRUE makes the self.self trade diagonal the amount from trade matrix, as opposed to the domestic consumption diagonal of Kastner matrix

Value

MAgPIE object with bilateral trade adjusted using Kastner method. Dimensions: (importer_exporter, year, product) Values represent apparent consumption of bilateral trade (production + imports - exports)

Author(s)

David M Chen

tradeValue *tradeValue*

Description

Calculates the value of traded goods based on a gdx file

Usage

```
tradeValue(
  gdx,
  file = NULL,
  level = "reg",
  products = "k_trade",
  product_aggr = FALSE,
  type = "net-exports",
  glo_weight = "export",
  relative = FALSE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation ("reg", "glo", "regglo")
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr")
product_aggr	aggregate over products or not (boolean)
type	exports-imports ("net-exports"), gross imports ("imports") or gross exports ("exports"); only valid if relative=FALSE
glo_weight	Decides the calculation of global prices. Weighting schemes are applied for estimation of global producer price. If "export" prices are calculated as average of regional exporters' prices, weighted by the export volumes. If "production" (default), prices are calculated as average of regional prices weighted by regional production. Alternatively, if "free_trade", the global prices are directly taken from the shadow prices of the global trade constraint, and no averaging is performed. Alternatively, if "constant_prices_initial" constant 1995 global prices for each commodity are used as weight.
relative	if relative=TRUE, self sufficiencies are reported (the amount of production divided by domestic demand)

Value

A MAgPIE object containing the value of trade flows in Million of US dollars

Author(s)

Misko Stevanovic, Florian Humpenoeder, Edna J. Molina Bacca

Examples

```
## Not run:
x <- tradeValue(gdx)

## End(Not run)
```

`tryList` *tryList*

Description

Internal support function to run a list of reportings in a `tryReport` environment.

Usage

```
tryList(..., gdx, level = "regglo")
```

Arguments

`...`	report function to be run
`gdx`	gdx file to report from
`level`	spatial level (either "regglo" for region+global or "iso" for ISO countries)

Value

A list of magpie objects (successful reports) or NULL (failed reports)

Author(s)

Jan Philipp Dietrich

See Also

`tryReport`, `reportResult`

Other Infrastructure: `expectVariablesPresent()`, `metadata_comments()`, `out()`, `tryReport()`

`tryReport` *tryReport*

Description

Internal support function to run a reporting in a try environment and properly report problems if something goes wrong without stopping the further processing in case of an error.

Usage

```
tryReport(report, gdx, level = "regglo", env = parent.frame())
```

Arguments

`report`	report function to be run
`gdx`	gdx file to report from
`level`	spatial level (either "regglo" for region+global, "iso" for country-level, or the file of a mapping file).
`env`	environment to evaluate the report in

Value

A named list with information on the outcome of the report (success, error, validationError, warning)

Author(s)

Jan Philipp Dietrich

See Also

`reportResult`

Other Infrastructure: `expectVariablesPresent()`, `metadata_comments()`, `out()`, `tryList()`

`validation` *validation*

Description

Create Validation pdf from MAgPIE output and corresponding validation.mif

Usage

```
validation(
  gdx,
  hist,
  file = "validation.pdf",
  runinfo = NULL,
  clusterinfo = NULL,
  debug = FALSE,
  reportfile = NULL,
  scenario = NULL,
  getReport = NULL,
  ...
)
```

Arguments

gdx	GDX file
hist	Validation data. All formats allowed which can be converted to quitte (including characters containing the path to a mif file)
file	a file name the output pdf
runinfo	(optional) Rdata object with run information
clusterinfo	(optional) RDS file or vector containing mapping information on 0.5degree between regions and cluster
debug	Switch to activate or deactivate debug mode
reportfile	file name to which a backup of the magpie reporting should be written (file ending should be ".mif"). No report written if set to NULL or if report is already provided via getReport!
scenario	scenario name used inside reportfile. Not used if reportfile is NULL.
getReport	the return value of the getReport function. Can be provided if available to reduce overall runtime.
...	additional arguments supplied to the validationpdf function

Author(s)

Jan Philipp Dietrich

Examples

```
  ## Not run:
    validation("fulldata.gdx","validation.mif",filter="Yield")

## End(Not run)
```

ValueMaterialDemand *ValueMaterialDemand*

Description

calculates agricultural costs without taxes and incentives (i.e. GHG taxes and BII incentives)

Usage

```
ValueMaterialDemand(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	aggregation level, reg, glo or regglo

Author(s)

David M Chen

Examples

```
## Not run:
x <- ValueMaterialDemand(gdx)

## End(Not run)
```

VegfruitShare *VegfruitShare*

Description

Calculates the share of fruits, vegetables and nuts in total food supply from the food demand model

Usage

```
VegfruitShare(
  gdx,
  file = NULL,
  level = "reg",
  after_shock = TRUE,
  calibrated = TRUE,
  attributes = "kcal"
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global)
after_shock	FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income
calibrated	if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used
attributes	unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr").

Value

magpie object with the livestock share in a region or country. Unit is dimensionsless, but value depends on chosen attribute

Author(s)

Benjamin Bodirsky

Examples

```
  ## Not run:
    x <- VegfruitShare(gdx)

## End(Not run)
```

---

wageDevelopment *wageDevelopment*

Description

calculates indicator to describe wage development based on agricultural wages in MAgPIE (hourly labor costs relative to a base year)

Usage

```
wageDevelopment(gdx, baseYear = 2000, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
baseYear	year relative to which the wage development should be calculated
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation ("iso", "reg", "glo", "regglo")

Value

MAgPIE object containing indicator on wage development

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- wageDevelopment(gdx)

## End(Not run)
```

---

wageRent *wageRent*

Description

calculates wage rent for exogenous wage scenarios

Usage

```
wageRent(gdx, file = NULL, level = "regglo")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	aggregation level, reg, glo or regglo

Author(s)

Debbora Leip

Examples

```
## Not run:
x <- wageRent(gdx)

## End(Not run)
```

`water_AAI` *water_AAI*

Description

reads area actually irrigated from a MAgPIE gdx file

Usage

```
water_AAI(gdx, file = NULL, level = "reg")
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using `write.magpie`. See `write.magpie` for supported file types
`level`	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate

Value

A MAgPIE object containing the area actually irrigated (Mha)

Author(s)

Stephen Wirth, Anne Biewald, Felicitas Beier

Examples

```
## Not run:
x <- water_AEI(gdx)

## End(Not run)
```

water_AEI *water_AEI*

Description

reads area equipped for irrigation from a MAgPIE gdx file

Usage

```
water_AEI(gdx, file = NULL, level = "reg")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing the area equipped for irrigation (Mha)

Author(s)

Markus Bonsch

Examples

```
  ## Not run:
    x <- water_AEI(gdx)

## End(Not run)
```

water_avail *water_avail*

Description

reads available water from a MAgPIE gdx file

Usage

```
water_avail(
  gdx,
  file = NULL,
  level = "reg",
  sources = NULL,
  sum = TRUE,
  digits = 4
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
sources	Vector of water sources that shall be obtained. NULL for all sources
sum	Sum the contribution of different sources (TRUE) or display them individually (FALSE)
digits	integer. For rounding of the return values

Value

A MAgPIE object containing the available water (km^3)

Author(s)

Markus Bonsch, Felicitas Beier

Examples

```
## Not run:
x <- water_avail(gdx)

## End(Not run)
```

---

water_efficiency *water_efficiency*

Description

reads Irrigation efficiency from a MAgPIE gdx file

Usage

```
water_efficiency(gdx, file = NULL, level = "reg")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate

Value

A MAgPIE object containing the irrigation efficiency on the requested aggregation level

Author(s)

Markus Bonsch

Examples

```
  ## Not run:
    x <- water_efficiency(gdx)

## End(Not run)
```

---

water_price *water_price*

Description

reads water prices from a MAgPIE gdx file

Usage

```
water_price(
  gdx,
  file = NULL,
  level = "reg",
  weight = "value",
  index = FALSE,
  index_baseyear = 2005,
  digits = 4
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate
weight	For determining weights to use for generating water prices at levels beyond 'cellular'. Takes "value" and "quantity". "value" sums regional weights by value of water per cluster, "quantity" sums regional weight by qty of water per cluster
index	FALSE (default) or TRUE
index_baseyear	baseyear to use for index calculation (only used if index=TRUE)
digits	integer. For rounding of the return values

Value

A MAgPIE object containing the water shadow prices (US Dollar/cubic metre).

Author(s)

Markus Bonsch, Vartika Singh, Miodrag Stevanovic

Examples

```
  ## Not run:
    x <- water_price(gdx)

## End(Not run)
```

---

water_usage *water_usage*

Description

reads area usage from a MAgPIE gdx file

Usage

```
water_usage(
  gdx,
  file = NULL,
  level = "reg",
  users = NULL,
  sum = FALSE,
  seasonality = "total",
  abstractiontype = "withdrawal",
  digits = 4
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "grid" (grid-cell) "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate
users	NULL or "sectors" or "kcr" or "kli". If NULL, all sectors including crop-wise water use and livestock will be obtained. If sectors, will only report for high-level sectors - agriculture, industry, electricity, domestic, ecosystem. Sum is applicable only in the case of sectors
sum	determines whether output should be sector specific (FALSE) or aggregated over all sectors (TRUE)
seasonality	water usage time of the year. options: "grper" (growing period) or "total" (entire year). Note: currently only implemented for non-agricultural water usage.
abstractiontype	water usage abstraction type: "withdrawal" or "consumption"
digits	integer. For rounding of the return values

Value

A MAgPIE object containing the water usage (km^3/yr)

Author(s)

Markus Bonsch, Vartika Singh, Felicitas Beier

Examples

```
## Not run:
x <- water_usage(gdx)

## End(Not run)
```

---

waterEFR *waterEFR*

Description

reads environmental flow requirements (as they enter MAgPIE) from a MAgPIE gdx file

Usage

```
waterEFR(gdx, file = NULL, level = "cell", digits = 4)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global)
digits	integer. For rounding of the return values

Value

A MAgPIE object containing environmental flow requirements (km^3)

Author(s)

Felicitas Beier

Examples

```
## Not run:
x <- waterEFR(gdx)

## End(Not run)
```

---

waterEFVarea *waterEFVarea*

Description

calculates area that falls into cluster experiencing environmental flow violations from MAgPIE outputs

Usage

```
waterEFVarea(gdx, file = NULL, level = "reg", digits = 4)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global), or "grid" (for disaggregated output using cropland as weight)
digits	integer. For rounding of the return values

Value

A MAgPIE object containing the area under environmental flow violations (Mha)

Author(s)

Felicitas Beier

Examples

```
## Not run:
x <- waterEFVarea(gdx)

## End(Not run)
```

`waterEFViolation` *waterEFViolation*

Description

calculates environmental flow violation volume from MAgPIE outputs

Usage

```
waterEFViolation(gdx, file = NULL, level = "reg", digits = 4)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global), or "grid" (for disaggregated output using cropland as weight)
`digits`	integer. For rounding of the return values

Value

A MAgPIE object containing the volume of environmental flow violations (km^3)

Author(s)

Felicitas Beier

Examples

```
## Not run:
x <- waterEFViolation(gdx)

## End(Not run)
```

waterEFVratio *waterEFVratio*

Description

calculates ratio of environmental flow violation volume (EFV) to water environmental flow requirements (EFR) in MAgPIE.

Usage

```
waterEFVratio(gdx, file = NULL, level = "cell")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or "grid" (grid cell)

Value

MAgPIE object

Author(s)

Felicitas Beier

Examples

```
  ## Not run:
    x <- waterEFVratio(gdx)

## End(Not run)
```

waterStress *waterStress*

Description

calculates which areas are affected by water stress from water availability and water demand in MAgPIE. Water stress is calculated based on the proportion of water withdrawals to water availability. Thresholds based on World Resources Institute definition (https://www.wri.org/data/water-stress-country): Low stress: <10 Low-to-medium stress: 10-20 Medium to high stress: 20-40 High stress: 40-80 Extremely high stress: >80

Usage

```
waterStress(gdx, stressRatio = 0.4, file = NULL, level = "cell")
```

Arguments

gdx	GDX file
stressRatio	threshold defining level of water stress (e.g. 0.2 for medium water stress, 0.4 for high water stress)
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global)

Value

MAgPIE object indicating whether location is water stressed (1) or not (0)

Author(s)

Felicitas Beier

Examples

```
## Not run:
x <- waterStress(gdx)

## End(Not run)
```

---

waterStressedPopulation

waterStressedPopulation

Description

People living in water stressed region

Usage

```
waterStressedPopulation(gdx, file = NULL, level = "cell", absolute = TRUE)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or "grid" (grid cell)
absolute	TRUE: reports people living in water stressed region in million, FALSE: returns share of population

Value

MAgPIE object

Author(s)

Felicitas Beier

Examples

```
  ## Not run:
    x <- waterStressRatio(gdx)

## End(Not run)
```

`waterStressRatio` *waterStressRatio*

Description

calculates water stress ratio from water availability and water demand in MAgPIE. Water stress ratio is the ratio of water withdrawals (in the growing period) to water availability (in the growing period)

Usage

```
waterStressRatio(gdx, file = NULL, level = "cell")
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or "grid" (grid cell)

Value

MAgPIE object

Author(s)

Felicitas Beier

Examples

```
  ## Not run:
    x <- waterStressRatio(gdx)

## End(Not run)
```

woodHarvestArea *woodHarvestArea*

Description

Reads wood harvest area separated by source (primforest, secdforest, forestry, other) and age classes from a gdx. The data is on cluster level and the unit is Mha per year.

Usage

```
woodHarvestArea(gdx)
```

Arguments

gdx	A fulldata.gdx of a magpie run, usually with endogenous forestry enabled

Value

A magpie object with the following dimensions: region, id, year, source, ageClass

Author(s)

Pascal Sauer

woodProduction *woodProduction*

Description

Reads roundwood and fuelwood production/harvest data separated by source (primforest, secdforest, forestry, other) from a gdx. The data is on cluster level and the unit is Petagram (= mio. t) dry matter per year (Pg DM yr-1).

Usage

```
woodProduction(gdx)
```

Arguments

gdx	A fulldata.gdx of a magpie run, usually with endogenous forestry enabled

Value

A magpie object with the following dimensions: region, id, year, source, woodType

Author(s)

Pascal Sauer

yields *yields*

Description

Calculates crop yields based on a MAgPIE gdx file

Usage

```
yields(
  gdx,
  file = NULL,
  level = "reg",
  products = "kcr",
  product_aggr = FALSE,
  attributes = "dm",
  water_aggr = TRUE
)
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregateX
products	Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr"), also including "pasture"
product_aggr	aggregate over products or not (boolean)
attributes	dry matter: Mt/ha ("dm"), gross energy: PJ/ha ("ge"), reactive nitrogen: Mt/ha ("nr"), phosphor: Mt/ha ("p"), potash: Mt/ha ("k"), wet matter: Mt/ha ("wm"). Can also be a vector.
water_aggr	aggregate irrigated and non-irriagted production or not (boolean).

Value

crop yield as MAgPIE object (unit depends on attributes)

Author(s)

Florian Humpenoeder

See Also

reportYields

Examples

```
  ## Not run:
    x <- yields(gdx)

## End(Not run)
```

`YieldsCropCalib` *YieldsCropCalib*

Description

Reads potential yields after calibration

Usage

```
YieldsCropCalib(gdx, file = NULL, level = "cell", tau = FALSE)
```

Arguments

`gdx`	GDX file
`file`	a file name the output should be written to using write.magpie
`level`	Level of regional aggregation
`tau`	if TRUE add effect of TAU to the yield

Value

A MAgPIE object containing values of potential yields after the calibration routines. With `tau = TRUE` the result equals `vm_yld`.

Author(s)

Edna Molina Bacca

Examples

```
## Not run:
x <- YieldsCropCalib(gdx)

## End(Not run)
```

YieldsCropRaw *YieldsCropRaw*

Description

Reads uncalibrated potential yields, i.e. before the calibration routines

Usage

```
YieldsCropRaw(gdx, file = NULL, level = "cell")
```

Arguments

gdx	GDX file
file	a file name the output should be written to using write.magpie
level	Level of regional aggregation

Value

A MAgPIE object containing values of potential yields as they come from the crop model, before the calibration routines are applied.

Author(s)

Edna Molina Bacca

Examples

```
## Not run:
x <- YieldsCropRaw(gdx)

## End(Not run)
```

Index

∗ **Data Changelog**
addToDataChangelog, 10
changelogVariables, 24
changelogYears, 24
∗ **Infrastructure**
expectVariablesPresent, 59
metadata_comments, 120
out, 126
tryList, 320
tryReport, 321
∗ **Spatial**
addGeometry, 9
clusterOutputToTerraVector, 25
gdxAggregate, 72
mappingToLongFormat, 119
superAggregateX, 305
∗ **ValidationPDF**
validation, 321

addGeometry, 9
addGeometry(), *26*, *72*, *120*, *306*
addToDataChangelog, 10, *24*
addToDataChangelog(), *24*
agEmployment, 11
AgGDP, 12
AgriResearchIntensity, 13
anthropometrics, 13

BII, 15, *48*
biochar, 16
bioplasticDemand, 17
bodyweight, 18

carbonHWP, 19
carbonLTS, 20, *113, 114*
carbonstock, 22
cellularFit, 23
changelogVariables, 24
changelogVariables(), *11*, *24*
changelogYears, 24
changelogYears(), *11*, *24*
clearCacheMagpie4, 25
clusterOutputToTerraVector, 25
clusterOutputToTerraVector(), *10*, *72*, *120*, *306*
consumptionValue, 26
CostCapital, 27
CostOverall, 28
costs, 29
CostsAEI, 30
CostsFertilizer, 30
costsMACCS, 31
costsPresolve, 32
costsWholesale, 33
CostsWithoutIncentives, 33
CostTC, 34
CostTransport, 35
croparea, 36, *48*
CropareaDiversityIndex, 37, *37*
cropland, 38
croplandSCM, 39
croplandTreeCover, 39
cshare, 40

deco, 41
demand, 42, *143*, *297*
demandBioenergy, 43
disaggregateLandConservation, 44
discountRates, 45
domesticFootprintTrade, 45

embodiedBiodiversity, 47
embodiedEmissions, 48, *49*
embodiedLabor, 50, *50*
embodiedLand, 51, *51, 52*
embodiedResourceKastner, *48–51*, 52, *52–54*, *68–70*, *145, 146*, *195*, *197*
embodiedWater, *52*, 53, *54*
emisCO2, 55, *141*
Emissions, 56, *141*

EmissionsBeforeTechnicalMitigation, 57
emisSOC, 58
expectVariablesPresent, 59
expectVariablesPresent(), *121*, *127*, *320*, *321*
expenditureIndexFood, 60

factorCosts, 61
factorCostShares, 62
fallow, 63
feed, 63
FoodDemandModuleConsumerPrices, 64
FoodExpenditure, 65
FoodExpenditureShare, 66
foodmodelstat, 67
footprintDemand, 68, *70*
footprints, *68*, 69, *146*, *195*, *197*
ForestYield, 71

gdxAggregate, 72
gdxAggregate(), *10*, *26*, *120*, *306*
getReport, 73, *196*
getReportAgMIP, 74
getReportDemandStandalone, 76
getReportDietaryIndicators, 77
getReportFableScenathon, 78
getReportFSECAlessandroPassaro, 78
getReportFSECCosts, 79
getReportFSECCropDiversityGrid, 80
getReportFSECPollution, 81
getReportFSECSimonDietz, 82
getReportFSECStevenLord, 83
getReportGridINMS, 83
getReportGridNitrogenPollution, 84
getReportINMS, 85
getReportIso, 86
getReportMAgPIE2GAINS, 88
getReportMAgPIE2LPJmL, 89
getReportMAgPIE2REMIND, 90
getReportPBindicators, 91
GrowingStock, 93

harvested_area_timber, 94
hourlyLaborCosts, 95
Hunger, 95

income, 96
Intake, 97
IntakeDetailed, 98
IntakeDetailedProtein, 99

kayaFractions, 100, *102*
kayaLaspeyres, 101, *101*
Kcal, 102

laborCostsEndo, 104
laborProductivity, 105
land, *48*, 105
land_price, 107
landCarbonSink, 108
landConservation, 109
landForestry, 110
landTransitionMatrix, 111
lastIter, 112
legacyEmissions, 113, *156*, *181*
LivestockDemStructure, 115
LivestockShare, 116

m_yeardiff, 117
magpie4 *(magpie4-package)*, 9
magpie4-package, 9
malmquist, 118
ManureExcretion, 118
mappingToLongFormat, 119
mappingToLongFormat(), *10*, *26*, *72*, *306*
metadata_comments, 120
metadata_comments(), *60*, *127*, *320*, *321*
modelstat, 121

NetForestChange, 122
NitrogenBudget, 122, *191*, *226–228*, *231*
NitrogenBudgetNonagland, 123
NitrogenBudgetPasture, 124
NitrogenBudgetWithdrawals, 125

OtherLand, 125
out, 126
out(), *60*, *121*, *320*, *321*
outputCheck, 127
outputPerWorker, 127

PeatlandArea, 128
PeatlandEmissions, 129
PlantationEstablishment, 130
population, 131
PriceElasticities, 132
PriceGHG, 133
priceIndex, 134
priceIndexFood, 135

prices, 136
primaryPerSecondary, 138
PrimSecdOtherLand, 139
processing, 140
productEmissions, 141
production, *138*, 142
productionProfit, 143
productionRevenue, 144

readGDXBilateral, 145
reassignLivestockPathway, *53*, *70*, 145, *196*
relativeHourlyLaborCosts, 146
reportAAI, 147
reportAEI, 148
reportAgEmployment, 149
reportAgGDP, 150
reportAgriResearchIntensity, 151
reportAnthropometrics, 152
reportBII, 153
reportBiochar, 154
reportBiogasFeedstock, 154
reportBioplasticDemand, 155
reportCarbonstock, *73*, 156
reportConsumVal, 157
reportCostCapitalInvestment, 158
reportCostCapitalStocks, 159
reportCostOverall, 160
reportCosts, 161
reportCostsAccounting, 162
reportCostsAEI, 163
reportCostsFertilizer, 164
reportCostsInputFactors, 165
reportCostsMACCS, 166
reportCostsPresolve, 167
reportCostsTrade, 168
reportCostsWholesale, 168
reportCostsWithoutIncentives, 169
reportCostTC, 170
reportCostTransport, 171
reportCroparea, *36*, 172
reportCropareaGrid, 173
reportCropDiversity, 174
reportCroplandSCM, 175
reportCropResidues2ndBE, 176
reportDemand, 177
reportDemandBioenergy, 178
reportDemandNr, 179
reportEmissions, *73*, *156*, 180
reportEmissionsBeforeTechnicalMitigation, 184
reportExpenditureFoodIndex, 185
reportExtraResidueEmissions, 187
reportFactorCostShares, 188
reportFeed, 189
reportFeedConversion, 190
reportFertilizerNitrogen, 191
reportFireEmissions, 192
reportFit, 193
reportFoodExpenditure, 194
reportFootprints, *68*, *70*, 195
reportForestYield, 198
reportGridCroparea, 199
reportGridLand, 200
reportGridManureExcretion, 201
reportGrowingStock, 202
reportharvested_area_timber, 203
reportHourlyLaborCosts, 204
reportHunger, 205
reportIncome, 206
reportIntakeDetailed, 207
reportKcal, 208
reportLaborCostsEmpl, 209
reportLaborProductivity, 210
reportLandConservation, 211
reportLandTransitionMatrix, *111*, 213
reportLandUse, *38*, *106*, *110*, 216
reportLandUseChange, 218
reportLandUseForSEALS, 219
reportLivestockDemStructure, 220
reportLivestockShare, 221
reportManure, 222
reportManureFuel, 223
reportNetForestChange, 224
reportNitrogenBudgetCropland, 225
reportNitrogenBudgetNonagland, 227
reportNitrogenBudgetPasture, 228
reportNitrogenEfficiencies, 229, *229*
reportNitrogenPollution, 230
reportOutputPerWorker, 231
reportPastSoilCarbon, 232
reportPBbiosphere, 233
reportPBland, 234
reportPBnitrogen, 235
reportPBwater, 236
reportPeatland, 237
reportPlantationEstablishment, 238

reportPopulation, *132*, 239
reportPriceAgriculture, 240
reportPriceBioenergy, 241
reportPriceElasticities, 242
reportPriceFoodIndex, 243
reportPriceGHG, *133*, 244
reportPriceLand, 245
reportPriceShock, 246
reportPriceWater, 247
reportPriceWoodyBiomass, 248
reportProcessing, 249
reportProcessingResiduesForestry, 250
reportProcessingWoodResidues, 253
reportProducerPriceIndex, 253
reportProduction, *143*, 254, *297*
reportProductionBioenergy, 255
reportProductionGrowth, 257
reportProductionNr, 258
reportProtein, 259
reportRelativeHourlyLaborCosts, 260
reportResidues, 261
reportResult, *320, 321*
reportRotationLength, 262
reportRuralDemandShares, 263
reportSDG1, 264
reportSDG12, 265
reportSDG15, 266
reportSDG2, 267
reportSDG3, 268
reportSDG6, 269
reportSDG9, 270
reportSOM, 270
reportSOM2, 273
reportTau, 275
reportTc, 276
reportTimber, 277, *278*
reportTimberDemand, 278
reportTotalHoursWorked, 279
reportTrade, 280
reportTradeGrowth, 282
reportValueMaterialDemand, 283
reportValueTrade, 284
reportVegfruitShare, 285
reportWageDevelopment, 286
reportWaterAvailability, 287
reportWaterIndicators, 288
reportWaterUsage, 289
reportWoodFuel, 290
reportWorkingAgePopulation, 291
reportYields, 292, *338*
reportYieldsCropCalib, 293
reportYieldsCropRaw, 294
reportYieldShifter, 295, *296*
ResidueBiomass, *176*, 297, *298*, *300*
Residues, 298
ResidueUsage, *176*, 299
RotationLength, 300
ruralDemandShares, 301

Seed, 302
SOM, 303
SOM2, 304
submitCalibration, 304
superAggregateX, 305
superAggregateX(), *10*, *26*, *72*, *120*
surplusChange, 306

tau, 307
taxRevenueRotations, 308
tc, 309
Timber, 310
TimberDemand, 311
TimberProductionVolumetric, 312
timePeriods, 313
totalHoursWorked, 314
trade, *48*, *138*, 314
tradedPrimaries, *47*, 316
tradeKastner, 318
tradeValue, 319
tryList, 320
tryList(), *60*, *121*, *127*, *321*
tryReport, *320*, 321
tryReport(), *60*, *121*, *127*, *320*

validation, 321
ValueMaterialDemand, 323
VegfruitShare, 323

wageDevelopment, 324
wageRent, 325
water_AAI, 326
water_AEI, 327
water_avail, 327
water_efficiency, 328
water_price, 329
water_usage, 330
waterEFR, 331

waterEFVarea, 332
waterEFViolation, 333
waterEFVratio, 334
waterStress, 334
waterStressedPopulation, 335
waterStressRatio, 336
woodHarvestArea, 337
woodProduction, 337
write.magpie, *326*

yields, 338
YieldsCropCalib, 339
YieldsCropRaw, 340