Package 'LPDynR'

May 19, 2021

Title Land Productivity Dynamics Indicator

Version 1.0.2

Description

It uses 'phenological' and productivity-related variables derived from time series of vegetation indexes, such as the Normalized Difference Vegetation Index, to assess ecosystem dynamics and change, which
eventually might drive to land degradation. The final result of the Land Productivity Dynamics indicator
is a categorical map with 5 classes of land productivity dynamics, ranging from declining to increasing
productivity. See <https://eartharxiv.org/repository/view/2294/> for a description of the methods used in
the package to calculate the indicator.

Depends R (>= 3.6.0)

Imports raster, dplyr, stats, data.table, virtualspecies, magrittr,
rgdal, parallel

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

URL https://github.com/xavi-rp/LPDynR

BugReports https://github.com/xavi-rp/LPDynR/issues

NeedsCompilation no

Author Xavier Rotllan-Puig [aut, cre],
Eva Ivits [aut],
Michael Cherlet [aut]

Maintainer Xavier Rotllan-Puig <xavier.rotllan.puig@aster-projects.cat>

Repository CRAN

Date/Publication 2021-05-19 08:40:02 UTC

R topics documented:

baseline_lev 2
clust_optim 3
EFT_clust 5
LNScaling 6
LongTermChange 7
LPD_CombAssess 10
PCAs4clust 12
rm_multicol 13
sbd_cat 14
sb_cat 15
sl_cat 16
state_change 17
steadiness 18

Index **20**

`baseline_lev` *baseline_lev*

Description

baseline_lev() derives land productivity at the beginning of the time series on study, resulting in a 3-class RasterLayer object with (1) low, (2) medium and (3) high productivity

Usage

```
baseline_lev(
  obj2process = NULL,
  yearsBaseline = 3,
  drylandProp = 0.4,
  highprodProp = 0.1,
  cores2use = 1,
  filename = ""
)
```

Arguments

`obj2process` Raster* object (or its file name). If time series, each layer is one year

`yearsBaseline` Numeric. Number of years to be averaged and used as baseline. Optional. Default is 3

`drylandProp` Numeric. Proportion of drylands over total land, either expressed as a fraction of unity or percentage. Optional. Default is 0.4

`highprodProp` Numeric. Proportion of land classified as ’highly productive’ over total land, either expressed as a fraction of unity or percentage. Optional. Default is 0.1

`cores2use` Numeric. Number of cores to use for parallelization. Optional. Default is 1 (no parallelization)

`filename` Character. Output filename. Optional

Details

baseline_lev() uses the proportion of drylands over the total land ('drylandProp') to classify the level of productivity into low level. UNPD declares that 40 percent of the World's land resources are drylands (Middleton et al., 2011) and, therefore, 40 percent of pixels at the global level can be classified as low productivity land. This assumption is the default, but it should be adjusted for local and regional studies. In addition, baseline_lev() classifies by default 10 percent of pixels as high level of land productivity and the rest (100 - ('drylandProp' + 10)) as medium level. Proportion of pixels classified as 'high' can be also modified by passing the argument 'highprodProp'

Value

RasterLayer object

Author(s)

Xavier Rotllan-Puig

References

Middleton, N., L. Stringer, A. Goudie, and D. Thomas. 2011. "The Forgotten Billion. MDG Achievement in the Drylands." New York, NY, 10017, USA: United Nations Development Programme.

Examples

```
sb <- raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sb_cat.tif"))
baseline_lev(obj2process = sb,
             yearsBaseline = 3,
             drylandProp = 0.4,
             cores2use = 2)
```

`clust_optim` *clust_optim*

Description

clust_optim produces a scree plot with number of cluster at x-axis and total within-cluster sum of squares at y-axis

Usage

```
clust_optim(
  obj2clust = NULL,
  num_clstrs = seq(5, 50, 5),
  standardise_vars = TRUE,
  ...
)
```

Arguments

`obj2clust` RasterStack or RasterBrick object (or its file name). Each layer is one variable

`num_clstrs` Numeric. Optional. Vector with a sequence of number of clusters to check for optimal

`standardise_vars` Logical. Optional. If TRUE (default), variables are standardised (mean = 0; sd = 1)

`...` Optional. Arguments for `kmeans`

Details

The 'scree plot method' allows the user to assess how the quality of the K-means clustering improves when increasing the number of clusters. An elbow in the curve indicates the optimal number of clusters. K-means are run with `kmeans`

Value

A scree plot

Author(s)

Xavier Rotllan-Puig

See Also

`PCAs4clust`; `EFT_clust`; `kmeans`

Examples

```
dirctry <- paste0(system.file(package='LPDynR'), "/extdata")
variables_noCor <- rm_multicol(dir2process = dirctry,
                               multicol_cutoff = 0.7)
clust_optim(obj2clust = variables_noCor,
            num_clstrs = seq(5, 50, 5))
```

EFT_clust *EFT_clust*

Description

EFT_clust derives the Ecosystem Functional Types using K-means to perform a clustering on the pixels of the Raster* object

Usage

```
EFT_clust(
  obj2clust = NULL,
  n_clust = 20,
  standardise_vars = TRUE,
  filename = "",
  ...
)
```

Arguments

`obj2clust`	RasterStack or RasterBrick object (or its file name). Each layer is one variable
`n_clust`	Numeric. Number of total clusters. Optional. Default = 20
`standardise_vars`	Logical. Optional. If TRUE (default), variables are standardised (mean = 0; sd = 1)
`filename`	Character. Output filename. Optional
`...`	Arguments for `kmeans`. Optional

Details

`kmeans` does not optimize the final number of clusters. It needs to be set by means of 'n_clust' (default = 20). There are several methods and statistics to determine the optimal number. `clust_optim` produces a scree plot to help the user to decide the optimal number of clusters.

EFT_clust passes as default to `kmeans` iter.max = 500 and algorithm = "MacQueen", but these can be modified passing these arguments through '...'

Please note that the variables are standardised (mean = 0; sd = 1) before running the clustering

An evaluation of the clustering is provided together with the RasterLayer object. It is calculated as model$betweenss / model$totss * 100; where 'betweenss' and 'totss' are generated by `kmeans`

Value

A list with two components: (1) a RasterLayer object with the clusters and (2) a vector with the clustering evaluation in percentage

Author(s)

Xavier Rotllan-Puig

See Also

`PCAs4clust`; `clust_optim`; `kmeans`

Examples

```
dirctry <- paste0(system.file(package='LPDynR'), "/extdata")
variables_noCor <- rm_multicol(dir2process = dirctry,
                               multicol_cutoff = 0.7)
EFT_clust(obj2clust = variables_noCor,
         n_clust = 10)
```

`LNScaling` *LNScaling*

Description

LNScaling (Local Net Productivity Scaling) uses a productivity variable (Raster*), e.g. season growth, to calculate the actual status of land productivity relative to its potential in homogeneous land areas or Ecosystem Functional Types (RasterLayer). If the productivity variable 'ProdVar' is a RasterStack or RasterBrick object with time series, it is calculated the average of the last 5 years

Usage

```
LNScaling(EFTs = NULL, ProdVar = NULL, cores2use = 1, filename = "")
```

Arguments

`EFTs` RasterLayer object (or its file name). Ecosystem Functional Types. Its first variable has the number of EFT (cluster) each pixel belongs to

`ProdVar` Raster* object (or its file name). Productivity variable (e.g. Cyclic fraction -season growth-)

`cores2use` Numeric. Number of cores to use for parallelization. Optional. Default is 1 (no parallelization)

`filename` Character. Output filename. Optional

Details

The Local Net Primary Production Scaling (LNS) method (Prince, 2009) calculates the difference between the potential and actual Net Primary Production for each pixel in homogeneous land areas. The current land production related to the local potential reflects the current level of productivity efficiency and, therefore, it is useful for the delineation of a land productivity status map

Value

RasterLayer object

Author(s)

Xavier Rotllan-Puig

References

Prince, S.D., Becker-Reshef, I. and Rishmawi, K. 2009. "Detection and Mapping of Long-Term Land Degradation Using Local Net Production Scaling: Application to Zimbabwe." REMOTE SENSING OF ENVIRONMENT 113 (5): 1046–57

See Also

EFT_clust

Examples

```
dirctry <- paste0(system.file(package='LPDynR'), "/extdata")
variables_noCor <- rm_multicol(dir2process = dirctry,
                               multicol_cutoff = 0.7)
EFTs_raster <- EFT_clust(obj2clust = variables_noCor,
                         n_clust = 10)
sb <- raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sb_cat.tif"))

LNScaling(EFTs = EFTs_raster[[1]],
          ProdVar = sb)
```

LongTermChange *LongTermChange*

Description

LongTermChange combines the Steadiness Index with the baseline levels of land productivity and with the change of state along the time series, resulting in a 22-class object (see details)

Usage

```
LongTermChange(
  SteadinessIndex = NULL,
  BaselineLevels = NULL,
  StateChange = NULL,
  filename = ""
)
```

Arguments

`SteadinessIndex` RasterLayer object (or its file name). Steadiness Index (4-class)

`BaselineLevels` RasterLayer object (or its file name). Baseline levels of land productivity (beginning of time series; 3-class)

`StateChange` RasterLayer object (or its file name). Change of state of land productivity (beginning minus end of time series; 3-class)

`filename` Character. Output filename. Optional

Details

St1-low-No Change <- 1

St1-low-Change 1 categ <- 2

St1-low-Change 2 or more categs <- 3

St1-medium-No Change <- 4

St1-medium-Change 1 categ <- 5

St1-medium-Change 2 or more categs <- 6

St1-high-No Change <- 7

St1-high-Change 1 categ <- 8

St1-high-Change 2 or more categs <- 9

St2-low-No Change <- 10

St2-low-Change 1 categ <- 10

St2-low-Change 2 or more categs <- 10

St2-medium-No Change <- 11

St2-medium-Change 1 categ <- 11

St2-medium-Change 2 or more categs <- 11

St2-high-No Change <- 12

St2-high-Change 1 categ <- 12

St2-high-Change 2 or more categs <- 12

St3-low-No Change <- 13

St3-low-Change 1 categ <- 13

St3-low-Change 2 or more categs <- 13

St3-medium-No Change <- 14

St3-medium-Change 1 categ <- 14

St3-medium-Change 2 or more categs <- 14

St3-high-No Change <- 15

St3-high-Change 1 categ <- 15

St3-high-Change 2 or more categs <- 15

St4-low-No Change <- 16

St4-low-Change 1 categ <- 17

St4-low-Change 2 or more categs <- 18

St4-medium-No Change <- 19

St4-medium-Change 1 categ <- 20

St4-medium-Change 2 or more categs <- 21

St4-high-No Change <- 22

St4-high-Change 1 categ <- 22

St4-high-Change 2 or more categs <- 22

Values = 0 in the final map indicates that there is a scarcity of data in the productivity variable (i.e. only 1 year with data), so that the indicator cannot be calculated

Value

RasterLayer object

Author(s)

Xavier Rotllan-Puig

See Also

steadiness, baseline_lev, state_change

Examples

```
sb <- raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sb_cat.tif"))
SteadinessIndex_raster <- steadiness(obj2process = sb)
BaselineLevels_raster <- baseline_lev(obj2process = sb,
                                      yearsBaseline = 3,
                                      drylandProp = 0.4)
StateChange_raster <- state_change(obj2process = sb,
                                   yearsBaseline = 3)

LongTermChange(SteadinessIndex = SteadinessIndex_raster,
               BaselineLevels = BaselineLevels_raster,
               StateChange = StateChange_raster)
```

LPD_CombAssess *LPD_CombAssess*

Description

LPD_CombAssess combines a 'LandProd_change' map (RasterLayer) with a 'LandProd_current' map (RasterLayer), giving a 5-classes map ranging from declining to increasing land productivity. 'LandProd_current' is reclassified into two classes: pixels with less than 'local_prod_threshold' (in percentage; 50 by default) of potential local productivity (within the EFT) and pixels with more or equal to 'local_prod_threshold'.

If 'LandProd_current' = NULL, 'LandProd_change' is directly reclassified into the same 5-classes map without using 'LandProd_current'. See the ATBD for the way pixels are reclassified.

Usage

```
LPD_CombAssess(
  LandProd_change = NULL,
  LandProd_current = NULL,
  local_prod_threshold = 50,
  filename = ""
)
```

Arguments

`LandProd_change`
: RasterLayer object (or its file name). Land Productivity Long Term Change Map

`LandProd_current`
: RasterLayer object (or its file name). Land Productivity Current Status Map

`local_prod_threshold`
: Numeric. Potential local productivity threshold (within the Ecosystem Functional Type) in percentage. Optional. Default = 50

`filename`
: Character. Output filename. Optional

Details

LandProd_change c(1:6, 8:9) & LandProd_current < 'local_prod_threshold' <- 1 Declining land productivity

LandProd_change c(3, 6) & LandProd_current >= 'local_prod_threshold' <- 1 Declining land productivity

LandProd_change c(7) & LandProd_current < 'local_prod_threshold' <- 2 Early signs of decline of land productivity

LandProd_change c(1:2, 4:5, 8:9) & LandProd_current >= 'local_prod_threshold' <- 2 Early signs of decline of land productivity

LandProd_change c(7) & LandProd_current >= 'local_prod_threshold' <- 3 Negative fluctuation (stable, but stressed land prod.)

LandProd_change c(10:12) <- 3 Negative fluctuation (stable, but stressed land prod.) LandProd_change c(13:15) <- 4 Positive fluctuation (stable, not stressed land prod.)

LandProd_change c(16:17, 19) & LandProd_current < 'local_prod_threshold' <- 4 Positive fluctuation (stable, not stressed land prod.)

LandProd_change c(18, 20:22) & LandProd_current < 'local_prod_threshold' <- 5 Increasing land productivity

LandProd_change c(16:22) & LandProd_current >= 'local_prod_threshold' <- 5 Increasing land productivity

Values = 0 in the final map indicates that there is a scarcity of data in the productivity variable (i.e. only 1 year with data), so that the indicator cannot be calculated

'local_prod_threshold' is the threshold used to classify 'LandProd_current' into pixels with low or high potential productivity within its Ecosystem Functional Type

Value

RasterLayer

Author(s)

Xavier Rotllan-Puig

See Also

LongTermChange; LNScaling

Examples

```
sb <- raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sb_cat.tif"))
SteadinessIndex_raster <- steadiness(obj2process = sb)
BaselineLevels_raster <- baseline_lev(obj2process = sb,
                                      yearsBaseline = 3,
                                      drylandProp = 0.4)
StateChange_raster <- state_change(obj2process = sb,
                                   yearsBaseline = 3)
LandProd_change_raster <- LongTermChange(SteadinessIndex = SteadinessIndex_raster,
                                         BaselineLevels = BaselineLevels_raster,
                                         StateChange = StateChange_raster)

dirctry <- paste0(system.file(package='LPDynR'), "/extdata")
variables_noCor <- rm_multicol(dir2process = dirctry,
                               multicol_cutoff = 0.7)
EFTs_raster <- EFT_clust(obj2clust = variables_noCor,
                         n_clust = 10)

LandProd_current_raster <- LNScaling(EFTs = EFTs_raster[[1]],
                                     ProdVar = sb)
```

```
LPD_CombAssess(LandProd_change = LandProd_change_raster,
               LandProd_current = LandProd_current_raster)
```

PCAs4clust *PCAs4clust*

Description

PCAs4clust runs a two-steps process to prepare the data to be clustered

Usage

```
PCAs4clust(obj2process = NULL, cumul_var_threshold = 0.9, filename = "", ...)
```

Arguments

obj2process	Raster* object (or its file name). Each layer is one variable
cumul_var_threshold	Numeric. Optional (default = 0.9). Threshold of cumulative variance to select the number of useful PCs
filename	Character. Output filename. Optional
...	Optional. Arguments for prcomp

Details

Firstly, a Principal Component Analysis ('screening PCA') with all the variables in 'obj2process' is run in order to know the optimal number of variables to be used in a subsequent PCA, as well as the most associated variable to those Principal Components (PCs). A threshold of cumulative variance (cumul_var_threshold; default = 0.9) is needed. Secondly, a 'final PCA' is run with the results of the 'screening PCA' (i.e. number of PC axes and their most associated variables). PCAs4clust uses prcomp to run PCAs

Value

RasterBrick object

Author(s)

Xavier Rotllan-Puig

See Also

rm_multicol; prcomp

Examples

```
dirctry <- paste0(system.file(package='LPDynR'), "/extdata")
variables_noCor <- rm_multicol(dir2process = dirctry,
                               multicol_cutoff = 0.7)
PCAs4clust(obj2process = variables_noCor,
           cumul_var_threshold = 0.9)
```

rm_multicol *rm_multicol*

Description

rm_multicol calculates the average of each variable and removes those variables highly correlated

Usage

```
rm_multicol(
  dir2process = NULL,
  multicol_cutoff = 0.7,
  cores2use = 1,
  filename = "",
  ...
)
```

Arguments

`dir2process`	Character. Directory where the Raster* objects are stored. All the .tif files in the directory will be read in to be used
`multicol_cutoff`	Numeric. Cutoff value of (Pearson's) correlation. Optional. Default is 0.70
`cores2use`	Numeric. Number of cores to use for parallelization. Optional. Default is 1 (no parallelization)
`filename`	Character. Output filename. Optional
`...`	Optional. Arguments for removeCollinearity()

Details

Firstly, after reading in all .tif files in 'dir2process', if they are multiband (time series), averages are calculated. Secondly, it creates a RasterBrick object with those (averaged) variables which have a Pearson's correlation coefficient below 'multicol_cutoff'. See `removeCollinearity` for further arguments and functionalities

Value

RasterBrick object

Author(s)

Xavier Rotllan-Puig

References

Leroy B, Meynard CN, Bellard C, Courchamp F (2015). “virtualspecies, an R package to generate virtual species distributions”. Ecography. doi: 10.1111/ecog.01388

See Also

removeCollinearity

Examples

```
dirctry <- paste0(system.file(package='LPDynR'), "/extdata")  # directory with variables to process
rm_multicol(dir2process = dirctry,
            multicol_cutoff = 0.7,
            plot = TRUE)
```

---

sbd_cat *Season Beginning Day*

Description

RasterBrick object containing time series of phenological data (400 cells; 10 layers). Years 2000-2009. The variable has been derived from MODIS using Timesat

Usage

```
sbd_cat
```

Format

GeoTIFF. RasterBrick object with dimensions: 20, 20, 400, 10 (nrow, ncol, ncell, nlayers)

sbd_cat Season beginning day

Details

Downloaded from www.eea.europa.eu (20/08/2020)

Coord. ref. : +proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs

Resolution : 500, 500 (x, y)

Extent : 3640000, 3650000, 2140000, 2150000 (xmin, xmax, ymin, ymax)

Source

https://www.eea.europa.eu/data-and-maps

References

https://www.eea.europa.eu/data-and-maps/data/annual-start-of-vegetation-growing

Examples

```
raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sbd_cat.tif"))
```

---

sb_cat *Standing Biomass*

Description

RasterBrick object containing time series of land productivity data (400 cells; 10 layers). Years 2000-2009. The variable has been derived from MODIS using Timesat

Usage

```
sb_cat
```

Format

GeoTIFF. RasterBrick object with dimensions: 20, 20, 400, 10 (nrow, ncol, ncell, nlayers)

sb_cat Standing biomass

Details

Downloaded from www.eea.europa.eu (20/08/2020)

Coord. ref. : +proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs

Resolution : 500, 500 (x, y)

Extent : 3640000, 3650000, 2140000, 2150000 (xmin, xmax, ymin, ymax)

Source

https://www.eea.europa.eu/data-and-maps

References

https://www.eea.europa.eu/data-and-maps/data/annual-above-ground-vegetation-productivity

Examples

```
raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sb_cat.tif"))
```

`sl_cat` *Season Length*

Description

RasterBrick object containing time series of phenological data (400 cells; 10 layers). Years 2000-2009. The variable has been derived from MODIS using Timesat

Usage

```
sl_cat
```

Format

GeoTIFF. RasterBrick object with dimensions: 20, 20, 400, 10 (nrow, ncol, ncell, nlayers)

sl_cat Season Length

Details

Downloaded from www.eea.europa.eu (20/08/2020)

Coord. ref. : +proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs

Resolution : 500, 500 (x, y)

Extent : 3640000, 3650000, 2140000, 2150000 (xmin, xmax, ymin, ymax)

Source

https://www.eea.europa.eu/data-and-maps

References

https://www.eea.europa.eu/data-and-maps/data/annual-above-ground-vegetation-season

Examples

```
raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sl_cat.tif"))
```

state_change *state_change*

Description

state_change derives land productivity state change between the beginning and the end of the time series on study, resulting in a 3-class RasterLayer object with (1) no change, (2) changed between 1 and x classes or (3) changed more than x classes, where x can be defined by the user (default is 1)

Usage

```
state_change(
  obj2process = NULL,
  yearsBaseline = 3,
  changeNclass = 1,
  cores2use = 1,
  filename = ""
)
```

Arguments

`obj2process`	Raster* object (or its file name). If time series, each layer is one year
`yearsBaseline`	Numeric. Number of years to be averaged at the beginning and end of the time series. Optional. Default is 3
`changeNclass`	Numeric. Number of classes changed for classification. Optional. Default is 1
`cores2use`	Numeric. Number of cores to use for parallelization. Optional. Default is 1 (no parallelization)
`filename`	Character. Output filename. Optional

Details

state_change uses the average of ’yearsBaseline’ number of years at the beginning and the end of the time series

Value

RasterLayer object

Author(s)

Xavier Rotllan-Puig

Examples

```
sb <- raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sb_cat.tif"))
state_change(obj2process = sb,
             yearsBaseline = 3,
             changeNclass = 1,
             cores2use = 2)
```

---

steadiness *steadiness*

Description

steadiness derives the Steadiness Index from a land productivity variable

Usage

```
steadiness(obj2process = NULL, cores2use = 1, filename = "")
```

Arguments

`obj2process`	Raster* object (or its file name). If time series, each layer is one year
`cores2use`	Numeric. Number of cores to use for parallelization. Optional. Default is 1 (no parallelization)
`filename`	Character. Output filename. Optional

Details

The Steadiness Index is based on the combination of two metrics calculated per pixel: (1) the slope derived from a linear regression of the different years of the time series and (2) the net change on the same period. It results in a 4-class RasterLayer object ranging from (1) strong negative to (4) strong positive ecosystem dynamics. See Ivits et al. (2013) for further explanations.

Values = 0 in the final map indicates that there is a scarcity of data in the productivity variable (i.e. only 1 year with data), so that the indicator cannot be calculated

Value

RasterLayer object

Author(s)

Xavier Rotllan-Puig

References

Ivits, E., M. Cherlet, W. Mehl, and S. Sommer. 2013. “Ecosystem Functional Units Characterized by Satellite Observed Phenology and Productivity Gradients: A Case Study for Europe.” Ecological Indicators 27: 17–28. doi: 10.1016/j.ecolind.2012.11.010

Examples

```
sb <- raster::brick(paste0(system.file(package='LPDynR'), "/extdata/sb_cat.tif"))
steadiness(obj2process = sb)
```

Index

∗ **datasets**
sb_cat, 15
sbd_cat, 14
sl_cat, 16

baseline_lev, 2, *9*

clust_optim, 3, *5, 6*

EFT_clust, *4*, 5, *7*

kmeans, *4–6*

LNScaling, 6, *11*
LongTermChange, 7, *11*
LPD_CombAssess, 10

PCAs4clust, *4*, *6*, 12
prcomp, *12*

removeCollinearity, *13, 14*
rm_multicol, *12*, 13

sb_cat, 15
sbd_cat, 14
sl_cat, 16
state_change, *9*, 17
steadiness, *9*, 18