Package ‘FAMEFMR’

November 5, 2020

Title Functions used in the DELWP FAME fire analysis process

Version 0.0.0.9000

Description What the package does (one paragraph).

License `use_gpl3_license()`

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports doParallel,
tidyr,
dplyr,
fasterize,
filematrix,
foreach,
mgcv,
raster,
sf,
tabularaster,
tictoc

Depends data.table

R topics documented:

add_xystring 2
calcBBTFI_2 3
calcDeltaAbund 4
calcDraftSpList 5
calcU_All_Combs 6
calc_TFI_2 6
cellsToHectares 7
cropNAborder 8
fhProcess 9
get_Spp_No 10

inputRasters 11
Join_Names 11
LBY_f 12
makeGS_LU 12
makeGS_Summary 13
makeSppYearSum2 14
make_Spp_LU_list 15
notAllIn 15
removeEmptyDirs 16
unlistPivot_wider 16

Index **17**

add_xystring *Adds concatenated String of X and Y coordinates of centroids of polygons to Simple Features polygon object. This String acts as a key to identify spatially identical polygons for use in tidyverse pivot functions.*

Description

Adds concatenated String of X and Y coordinates of centroids of polygons to Simple Features polygon object. This String acts as a key to identify spatially identical polygons for use in tidyverse pivot functions.

Usage

```
add_xystring(myDF)
```

Arguments

`myDF`	sf polygon object

Value

character vector of XYStrings

`calcBBTFI_2` *Calculate area BBTFI and BBTFI rasters*

Description

Calculate area BBTFI and BBTFI rasters

Usage

```
calcBBTFI_2(
  FHanalysis,
  U_AllCombs_TFI = myAllCombs$U_AllCombs_TFI,
  Index_AllCombs = myAllCombs$Index_AllCombs,
  TFI_LUT,
  makeBBTFIrasters = makeBBTFIrasters
)
```

Arguments

`FHanalysis`	list containing all the fire history spatial attributes created by function fhProcess
`U_AllCombs_TFI`	data.table giving all combinations of cell values from the input rasters for the FAME analysis
`Index_AllCombs`	integer index mapping U_AllCombs_TFI to raster cells
`TFI_LUT`	data.frame lookup table from EFG for "MIN_LO_TFI","MIN_HI_TFI","MAX_TFI","EFG_NAME", read from settings
`makeBBTFIrasters`	logical whether or not to export rasters for BBTFI to disk

Details

Calculate summary area burned below TFI BBTFI for each SEASON in analysis (accommodating Hi and Lo fire intensity of first burn to determine TFI) and cumulative area BBTFI. Also optionally outputs rasters mapping areas BBTFI

Value

list containing:

- the date sequence matrix for each cell of the raster
- the EFG TFI Lookup for each cell of the raster
- the raster resolution used.
- Optionally outputs rasters of BBTFI to disk if makeBBTFIrasters==TRUE.

`calcDeltaAbund` *Summary of changes in relative abundance*

Description

Summary of changes in relative abundance

Usage

```
calcDeltaAbund(
  SpYearSumm = SpYearSummWide,
  FHanalysis,
  myBaseline,
  ResultsDir,
  HDMSpp_NO,
  TaxonList
)
```

Arguments

`SpYearSumm`	data.frame output by function makeSppYearSum2()
`FHanalysis`	list containing all the fire history spatial attributes created by function fhProcess
`myBaseline`	integer single SEASON or sequence of SEASons used to create the baseline relative species abundance for comparison of change
`ResultsDir`	path of directory where results will be written usually generated by FAME script
`HDMSpp_NO`	path of directory where results will be written usually generated by FAME script
`TaxonList`	data.frame of species attributes (read from default or user provided .csv)

Details

Calculates the change in relative abundance compared to a baseline SEASON or mean of SEASONS

Value

data frame wide format summary chance in relative abundance of species SEASON

calcDraftSpList *Calculate a draft species list for defined polygon*

Description

Calculate a draft species list for defined polygon

Usage

```
calcDraftSpList(
  REG_NO,
  RasterRes = 225,
  PUBLIC_LAND_ONLY,
  myPoly = myPoly,
  generalRasterDir = "./InputGeneralRasters",
  splist = "./ReferenceTables/DraftTaxonListStatewidev2.csv",
  HDMVals = HDMVals225
)
```

Arguments

`REG_NO` integer DELWP fire region number 1:6 ,99 for Statewide analysis, or 7 for ad hoc boundary polygon default =7 (see look up table REG_LUT for values)

`RasterRes` integer 225 - raster resolution is always 225 for this fucntion for speed

`PUBLIC_LAND_ONLY`
logical whether to restrict analysis to public land only or the whole polygon

`myPoly` default clipPoly sf polygon data frame of LF_REGIONs (default) or ad hoc polygon - used in conjunction with REG_NO

`generalRasterDir`
relative path to directory containing rasters of FIRE_REG, and PUBLIC LAND (PLM_GEN)

`splist` path to default species attribute table default is

`HDMVals`

Details

Calculate the proportion of cells for the HDM in the region for each species is intended only as a starting point and requires manual quality control to produce a useful species list for the area by editing the resulting .csv file

Value

data.frame created from splist with columns appended for:

- cellsInState count of the number of cells in the state within the Binary HDM for the species

- cellsInArea count of the number of cells within myPoly and within the Binary HDM for the species
- areaProp proportion of binary HDM for the state within myPoly

`calcU_All_Combs` *Calculate all combinations of input raster values*

Description

Calculate all combinations of input raster values

Usage

```
calcU_All_Combs(FHAnalysis, cropRasters)
```

Arguments

`FHAnalysis`	list containing all the fire history spatial attributes created by function fhProcess()
`cropRasters`	list of rasters and indices and cell values created by function cropNAborder()

Value

list of: \

`calc_TFI_2` *Main Tolerable fire interval (TFI) status calculation*

Description

Main Tolerable fire interval (TFI) status calculation

Usage

```
calc_TFI_2(
  FHanalysis,
  U_AllCombs_TFI = myAllCombs$U_AllCombs_TFI,
  Index_AllCombs = myAllCombs$Index_AllCombs,
  TFI_LUT,
  OutputRasters = makeTFIRasters
)
```

Arguments

`FHanalysis` list containing all the fire history spatial attributes created by function fhProcess()

`U_AllCombs_TFI` data.table giving all combinations of cell values from the input rasters for the FAME analysis made in function calcU_All_Combs()

`Index_AllCombs` integer index mapping U_AllCombs_TFI to raster cells made in function calcU_All_Combs()

`TFI_LUT` data.frame Lookup table from EFG for "MIN_LO_TFI","MIN_HI_TFI","MAX_TFI","EFG_NAME", read form settings

`OutputRasters` logical whether to uptut rasters of TFI status for each year

Details

Calculates where each cell is currently at below MinTFI or above MAX_TFI returns the per cell and long table summarised by multiple admin units and evc

---

`cellsToHectares` *Calculates multiplier to convert from raster cell count to area in hectares*

Description

Calculates multiplier to convert from raster cell count to area in hectares

Usage

```
cellsToHectares(RasterMetres = RasterRes)
```

Arguments

`RasterMetres` numeric Value cell resolution in Metres (usually from RasterRes in settings file).

Value

numeric Multiplier to convert cell count to area in hectares cropNAborder *crop border of NA cells from rasters and get cell indices for remaning cells function to get the minimum bounding box of the cells with non NA values in a raster and save them to crop other rasters to same extent. also creates some rasters cropped to correct extent for instance for region and EFG also gets indices of cells in raster of same extent as crop to the shape provided*

Description

crop border of NA cells from rasters and get cell indices for remaning cells function to get the minimum bounding box of the cells with non NA values in a raster and save them to crop other rasters to same extent. also creates some rasters cropped to correct extent for instance for region and EFG also gets indices of cells in raster of same extent as crop to the shape provided

Usage

```
cropNAborder(
  REG_NO = 7,
  RasterRes = RasterRes,
  PUBLIC_LAND_ONLY,
  myPoly = clipPoly,
  generalRasterDir = "./InputGeneralRasters"
)
```

Arguments

`REG_NO` integer DELWP fire region number 1:6 ,99 for Statewide analysis, or 7 for ad hoc boundary polygon default =7 (see look up table REG_LUT for values)

`RasterRes` numeric raster resolution of the analysis in metres (usually set in settings file or shiny app)

`PUBLIC_LAND_ONLY`
Logical TRUE/FALSE

`myPoly` default clipPoly sf polygon data frame of LF_REGIONs (default) or ad hoc polygon - used in conjunction with REG_NO

`generalRasterDir`
relative path to directory containing rasters of DELWP FIRE_REG, DELWP REGION, EFG, PUBLIC LAND (PLM_GEN)

Value

A list containing:

- Raster raster cropped of all border rows and columns that are all NA,
- Extent extent of the raster
- IDX integer vector cell numbers of cells in the cropped raster

- clipIDX integer vector cell numbers only for cells with the input polygon
- EFG integer vector EFG values for cells within clipped area
- RGN integer vector Fire Region numbers for cells within clipped area
- DELWP integer vector DELWP Region numbers for cells within clipped area
- PLM logical for cells within clipped area

`fhProcess` *Main Fire History Fire Sequence analysis function*

Description

Main Fire History Fire Sequence analysis function

Usage

```
fhProcess(
  rawFH = "path of the rawFH file to use - a shapefile",
  start.SEASON = NULL,
  end.SEASON = NULL,
  OtherAndUnknown,
  validFIRETYPE
)
```

Arguments

`rawFH` path to the input fire history shapefile usually provided in settings

`start.SEASON` integer First SEASON for which output is wanted (four digit year as integer), if NUll then second season in in history is used (cannot use first season because it has no interval, this may still fail if there is no overlap)

`end.SEASON` integer Last SEASON required, if NULL then largest value in fire history scenario used

`OtherAndUnknown`
integer Value to use for cases where fire type is: "OTHER" or "UNKNOWN" = NA, "BURN" = 1, "BUSHFIRE" = 2. NA = Fire excluded from analysis. usually set in settings file

`validFIRETYPE` character vector of valid FIRETYPE values for checking the input file , provided in settings file.

Details

The function takes a shapefile of Fire history contain polygons with two fields FIRETYPE and SEASON Where polygons of different FIRETYPE or SEASON overlap the function constructs unique non-overlapping polygon of their intersections (and non intersecting areas) and attributes each polygon with sequential fire SEASON (SEAS01, SEAS02 ...) and corresponding FIRETYPE (TYPE01,TYPE02 ...)

It then calculates all the intervals between sequential fires, and Time Since fire (TSF) and Last Fire Type (LFT) and Last burnt year (LBY) for each SEASON as defined in the input arguments, these values are append to the output sf polygon dataframe.

Value

A list containing:

- OutDF sf polygons dataframe containing all the fire history attributes
- TimeSpan integer vector sequence of SEASONS to in the analysis output
- YSFNames names of TSF years in output, needed by downstream functions
- LBYNames names of LBY years in output, needed by downstream functions
- LFTNames names of LBY years in output, needed by downstream functions

get_Spp_No *Extract VBA (Victorian Biodiversity Atlas) species ID numbers from file paths extracts four or five digit species numbers (Victorian Biodiversity Atlas TAXON_IDs) from vector of paths or file names containing files of e.g. species HDMS containing the 5 digit TAXON_ID in their name*

Description

Extract VBA (Victorian Biodiversity Atlas) species ID numbers from file paths extracts four or five digit species numbers (Victorian Biodiversity Atlas TAXON_IDs) from vector of paths or file names containing files of e.g. species HDMS containing the 5 digit TAXON_ID in their name

Usage

```
get_Spp_No(x = "Vector of Sp file Pathnames")
```

Arguments

x	Vector of species file Pathnames containing VBA numbers

Value

numeric vector of 4or 5 digits (ususally TAXON_ID)

`inputRasters` *Set correct input general rasters*

Description

Set correct input general rasters

Usage

```
inputRasters(RasterRes)
```

Arguments

`RasterRes` numeric raster resolution of the analysis in metres (usually set in settings file or shiny app)

Value

list of input raster names correct for RasterRes or error if RasterRes is not 75 or 225

`Join_Names` *Joins one or more lookup tables to table containing ID values Function joins Lookup tables (LUTS) to dataframe containing ID_NO: Name combinations*

Description

Joins one or more lookup tables to table containing ID values Function joins Lookup tables (LUTS) to dataframe containing ID_NO: Name combinations

Usage

```
Join_Names(myDF, LUTS = c("TFI_LUT", "FIREFMZ_LUT", "REG_LUT", "DELWP_LUT"))
```

Arguments

`myDF` dataframe or similar containing indices for the LUTS listed, to which the LUTS will be left_joined

`LUTS` vector of names of LUTS in memory defaults =c("TFI_LUT","FIREFMZ_LUT","REG_LUT","DELWP_

Value

a data.frame with the LUTS joined to it

LBY_f *Calculate last burned year matrix (LBY)*

Description

Calculate last burned year matrix (LBY)

Usage

```
LBY_f(M, y)
```

Arguments

M	numeric matrix of fire sequences sequence in rows, values are SEASON
y	numeric SEASON

Details

Function to calculate last burnt year (LBY) from matrix of rows of fire season iterating by year (y) used in calc_TFI_2

Value

matrix of last burned year row for each unique fire history column for each SEASON

makeGS_LU *Make long format Growth Stage Lookup matrix*

Description

Make long format Growth Stage Lookup matrix

Usage

```
makeGS_LU(EFG_TSF_4GS = myEFG_TSF_4GS)
```

Arguments

EFG_TSF_4GS	data.fame of growth stages for each EFG with start and end years

Details

expands a growth stage lookup table (provided in settings file) from four growth stages (1:4) per EFG with their years since fire spans as min(YSF) and max(YSF) to an array with YSF as row, EFG_NO as column and growth stage (1:4) as value. NOTE: YSF has 1 added to both the Lookup and the input to deal with YSF==0 which cannot be used in the array indexing

Value

matrix rows YSF, columns EFG_NO, values GS number (1:4)

`makeGS_Summary` *Summarise area by growth stage.*

Description

Summarise area by growth stage.

Usage

```
makeGS_Summary(
  FHanalysis,
  U_AllCombs_TFI = myAllCombs$U_AllCombs_TFI,
  Index_AllCombs = myAllCombs$Index_AllCombs
)
```

Arguments

`FHanalysis`	list containing all the fire history spatial attributes created by function fhProcess
`U_AllCombs_TFI`	data.table giving all combinations of cell values from the input rasters for the FAME anaysis
`Index_AllCombs`	integer index mapping U_AllCombs_TFI to raster cells #'

Details

Generates wide and long format summary of area for each EFG and season grouped by EFG, EFG_NAME, PLM ,FIRE_FMZ_NAME, FIRE_REGION_NAME, DELWP_REGION.

Value

list of two data.frames grouped by EFG, EFG_NAME, PLM ,FIRE_FMZ_NAME, FIRE_REGION_NAME, DELWP_REGION

- GS_Summary_wide Wide format table summarises area by Growth Stage and SEASON
- GS_Summary_long Long format table summarises area by Growth Stage and SEASON

makeSppYearSum2 *Species' relative abundance calculation and summary*

Description

Species' relative abundance calculation and summary

Usage

```
makeSppYearSum2(
  FHanalysis,
  myHDMSpp_NO = HDMSpp_NO,
  writeSpRasters = writeSpRasters,
  myLU_List = LU_List,
  ResultsDir = ResultsDir,
  HDMVals = HDMVals,
  TaxonList = TaxonList,
  writeYears = NULL,
  writeSp = writeSp
)
```

Arguments

`FHanalysis`	list containing all the fire history spatial attributes created by function fhProcess
`myHDMSpp_NO`	vector of TAXON_IDs for species to be included in output
`writeSpRasters`	logical: whether to also write species abundance rasters to disk
`myLU_List`	list of species abundance lookup arrays created by function make_Spp_LU_list()
`ResultsDir`	path of directory where results will be written usually generated by FAME script
`HDMVals`	matrix of cell values for Habitat Distribution Model rasters for (at least) all TAXON_ID in myHDMSpp_NO generally provided in settings file and read loaded by FAME script
`TaxonList`	data.frame of species attributes (read from default or user provided .csv)
`writeYears`	vector for SEASONS for which rasters are to be written otherwise if writeSpRasters == TRUE, if writeYears == NULL then all SEASONS are written out
`writeSp`	vector of TAXON_IDs provided if only subset of species rasters are required as output.

Details

Calculates the relative abundance of species for each raster cell in analysis and summaries these as summed abundance each season. Optionally it also write relative abundance rasters for species to disk

Value

data frame wide format summary of relative abundance of species by SEASONS

`make_Spp_LU_list` *Generate list of species abundance lookup arrays*

Description

Generate list of species abundance lookup arrays

Usage

```
make_Spp_LU_list(myHDMSpp_NO = HDMSpp_NO, myAbundDataLong = ExpertDataLong)
```

Arguments

`myHDMSpp_NO`	vector of VBA IDs for species to be included in analysis
`myAbundDataLong`	long format input lookup table of species abundance x YSF xEFG_NO x FIRETYPE_NO

Details

function creates a list of Lookup arrays for each taxon (VBA_CODE/TAXON_ID) for YSF x EFGNO x FireTypeNo these are then used in spatial calculation of species abundance functions

Value

list of 3D arrays named by TAXON_ID of relative abundance value for YSF x EFG x FIRETYPE_NO

`notAllIn` *Checks whether all values in one vector are in another vector*

Description

Checks whether all values in one vector are in another vector

Usage

```
notAllIn(x, v = V)
```

Arguments

`x`	Vector of values to check if all are in second vector
`v`	Second vector of values that may or may not contain all values in x

Value

logical

`removeEmptyDirs` *remove empty directories from path*

Description

remove empty directories from path

Usage

```
removeEmptyDirs(rootDir = "./Results")
```

Arguments

`rootDir`	relative path to remove all empty subdirectories from. Default value "./Results"

Details

Removes all empty subdirectories from the nominated path does not remove the nominated path directory even if empty

`unlistPivot_wider` *Fix Pivot_wider list of lists columns*

Description

Fix Pivot_wider list of lists columns

Usage

```
unlistPivot_wider(df)
```

Arguments

`df`	wide format data frame with fields that are lists of lists

Details

Supporting function to deal with pivot_wider returning list of lists in some cases

Value

wide format data frame without fields that are lists of lists

Index

add_xystring, 2

calc_TFI_2, 6
calcBBTFI_2, 3
calcDeltaAbund, 4
calcDraftSpList, 5
calcU_All_Combs, 6
cellsToHectares, 7
cropNAborder, 8

fhProcess, 9

get_Spp_No, 10

inputRasters, 11

Join_Names, 11

LBY_f, 12

make_Spp_LU_list, 15
makeGS_LU, 12
makeGS_Summary, 13
makeSppYearSum2, 14

notAllIn, 15

removeEmptyDirs, 16

unlistPivot_wider, 16