Package: scfmutils (via r-universe)

August 19, 2024

Title Tools and Utilities for Working With the SCFM Wildfire Simulation Model

Description The original fire model is described by Cumming et al. (1998), and more accessibly in Armstrong and Cumming (2003). It has recently been implemented as a collection of 'SpaDES' modules by Cumming, McIntire, Eddy, and Chubaty, available from <https://github.com/PredictiveEcology/scfm>.

URL https://scfmutils.predictiveecology.org, https://github.com/PredictiveEcology/scfmutils, https://predictiveecology.github.io/scfmutils/

Date 2024-08-19

Version 2.0.3

Depends R (>= 4.2)

Imports data.table, dplyr, fpCompare, ggplot2, graphics, grDevices, methods, purrr, raster, reproducible (>= 2.1.0), rlang, scam, sf, SpaDES.core (>= 2.1.5), SpaDES.tools (>= 2.0.7), stats, terra, tidyterra, tools, viridis

Suggests bcdata, googledrive, gridExtra, knitr, rmarkdown, roxygen2, sp, testthat

Remotes PredictiveEcology/reproducible@development, PredictiveEcology/SpaDES.core@development, PredictiveEcology/SpaDES.tools@development

Encoding UTF-8

Language en-CA

License GPL (>= 3)

VignetteBuilder knitr, rmarkdown

BugReports https://github.com/PredictiveEcology/scfmutils/issues

ByteCompile yes

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://predictiveecology.r-universe.dev

RemoteUrl https://github.com/PredictiveEcology/scfmutils

RemoteRef development

RemoteSha a2751c7f341f6d76b6491932687ef02203b9b43e

Contents

`scfmutils-package` `scfmutil` *package*

Description

The original fire model is described by Cumming et al. (1998), and more accessibly in Armstrong and Cumming (2003). It has recently been implemented as a collection of 'SpaDES' modules by Cumming, McIntire, Eddy, and Chubaty, available from https://github.com/PredictiveEcology/scfm.

Author(s)

Maintainer: Alex M Chubaty <achubaty@for-cast.ca> (ORCID)

Authors:

- Steve Cumming <Steve.Cumming@sbf.ulaval.ca>
- Eliot J B McIntire <eliot.mcintire@nrcan-rncan.gc.ca> (ORCID)
- Ian M S Eddy <ian.eddy@nrcan-rncan.gc.ca> (ORCID)

Other contributors:

- His Majesty the King in Right of Canada, as represented by the Minister of Natural Resources Canada [copyright holder]

See Also

Useful links:

- https://scfmutils.predictiveecology.org
- https://github.com/PredictiveEcology/scfmutils
- https://predictiveecology.github.io/scfmutils/
- Report bugs at https://github.com/PredictiveEcology/scfmutils/issues

---

`calcZonalRegimePars` `scfmRegime: calcZonalRegimePars`

Description

scfmRegime: calcZonalRegimePars

Usage

```
calcZonalRegimePars(
  polygonID,
  firePolys,
  firePoints,
  epochLength,
  maxSizeFactor,
  fireSizeColumnName,
  targetBurnRate = NULL,
  targetMaxFireSize = NULL
)
```

Arguments

polygonID	TODO
firePolys	TODO
firePoints	TODO
epochLength	TODO
maxSizeFactor	TODO
fireSizeColumnName	TODO
targetBurnRate	TODO
targetMaxFireSize	TODO

Value

list containing the following elements and their values: ignitionRate (ignition rate), pEscape (escape probability), xBar (mean fire size), lxBar (mean log-fire-size), xMax (maximum observed fire size), emfs_ha (estimated maximum fire size in ha), empiricalBurnRate (empircal burn rate)

---

calibrateFireRegimePolys

scfmDriver: calibrateFireRegimePolys

Description

Calibrate fire regime polygons ... (TODO)

Usage

```
calibrateFireRegimePolys(
  polygonType,
  targetN,
  fireRegimePolys,
  buffDist,
  pJmp,
  pMin,
  pMax,
  flammableMap = NULL,
  plotPath = NULL,
  outputPath = NULL,
  optimizer = "bfgs"
)
```

Arguments

polygonType	the names of polygons, i.e. PolyID
targetN	the number of fires to simulate during calibration
fireRegimePolys	sf polygon or multipolygon object defining the fire regime polygons
buffDist	buffer distance for cells available to be burned outside of each regime polygon
pJmp	numeric. default spread probability for degenerate polygons
pMin	numeric. minimum spread probability
pMax	numeric. maximum allowable spread probability
flammableMap	a packed SpatRaster (see terra::wrap()) with values 0 indicating non-flammable pixels, 1 flammable.
plotPath	character. file name specifying an output directory to use for producing plots of the scam fit for each polygon.
outputPath	character. path to output directory.
optimizer	character. the numerical optimization method to use with scam fitting; see ?scam.

Value

data.table with columns:

- PolyID: polygon ID;
- pSpread: spread probability;
- p0: TODO;
- naiveP0: TODO;
- pIgnition: ignition probability;
- maxBurnCells: maximum number of burned cells.

---

checkForIssues *Check for various issues with* fireRegimePolys

---

Description

Check for various issues with fireRegimePolys

Usage

```
checkForIssues(
  fireRegimePolys,
  studyArea,
  rasterToMatch,
  flammableMap,
  sliverThresh,
  cacheTag
)
```

Arguments

fireRegimePolys	sf polygon or multipolygon object defining the fire regime polygons
studyArea	sf object corresponding to the study area of interest.
rasterToMatch	SpatRaster object covering the spatial extent of studyArea, used as a template for raster layer creation.
flammableMap	SpatRaster with values 0 indicating non-flammable pixels, 1 flammable.
sliverThresh	minimum sliver size
cacheTag	character specifying additional user tags for caching

Value

a cleaned up fireRegimePolys object

---

comparePredictions_summaryDT

Create data.table *to compare scfm predictions with historical observations*

Description

Create data.table to compare scfm predictions with historical observations

Usage

```
comparePredictions_summaryDT(
  fireRegimePoints = NULL,
  burnSummary = NULL,
  fireRegimePolys = NULL,
  times = NULL
)

comparePredictions_meanFireSize(dt)

comparePredictions_fireReturnInterval(dt, times)

comparePredictions_annualIgnitions(dt)

comparePredictions_annualEscapes(dt)

comparePredictions_fireDistribution(
  fireRegimePoints = NULL,
  burnSummary = NULL,
  size
)
```

Arguments

fireRegimePoints	sf object produced scfmRegime module
burnSummary	data.table, produced by scfmSpread module
fireRegimePolys	sf polygon or multipolygon object defining the fire regime polygons
times	list of simulation start and end times (i.e., output from times(sim))
dt	scfm summary data.table produced by comparePredictions_summaryDT()
size	minimum fire size (ha)

Value

comparePredictions_summaryDT returns a data.table object; other functions return ggplot objects.

Author(s)

Ian Eddy

Examples

```
## Not run:
## assumes user has run scfm to produce the simList `mySimOut`
dt <- comparePredictions_summaryDT(fireRegimePoints = mySimOut$fireRegimePoints,
                                   burnSummary = mySimOut$burnSummary,
                                   fireRegimePolys = mySimOut$fireRegimePolys,
                                   times = times(mySimOut))

gg_mfs <- comparePredictions_meanFireSize(dt)
gg_fri <- comparePredictions_fireReturnInterval(dt)
gg_ign <- comparePredictions_annualIgnitions(dt)
gg_frp <- plot_fireRegimePolys(mySimOut$fireRegimePolys)

gridExtra::grid.arrange(fps,  gg_mfs,  gg_fri,  gg_ign, nrow = 2, ncol = 2)

## End(Not run)
```

---

deSliver *Merge sliver polygons into non-sliver neighbours*

Description

The threshold is applied to the area of the multipolygon object, not each individual polygon. Non-sliver polygons keep their original attributes. Intended to be used when it is important to retain the original extent of an area while removing sliver polygons.

Usage

```
deSliver(x, threshold)
```

Arguments

x	an sf POLYGONS or MULTIPLOYGONS object
threshold	the minimum area below which a polygon is considered a sliver

Value

an object of class sf with sliver polygons merged to their nearest valid neighbour.

---

executeDesign scfmDriver: executeDesign

Description

DESCRIPTION NEEDED

Usage

```
executeDesign(L, dT, maxCells)
```

Arguments

L	TODO
dT	TODO
maxCells	TODO

Value

TODO

`fireRegimePolyTypes` `prepInputsFireRegimePolys`

Description

Create fire regime polygons for `scfmRegime`.

Usage

```
fireRegimePolyTypes()

prepInputsFireRegimePolys(
  url = NULL,
  destinationPath = tempdir(),
  studyArea = NULL,
  rasterToMatch = NULL,
  type = "ECOREGION"
)
```

Arguments

`url`	character. URL from which to download and prepare fire regime polygons. Defaults are provided for Canadian ecodistrict, ecoregion, ecoprovince, and ecozone, as well as national Fire Regime Types and Fire Regime Units from Erni et al. (2020) doi:10.1139/cjfr20190191.
`destinationPath`	character. Path to directory where data will be downloaded.
`studyArea`	`sf` object corresponding to the study area of interest.
`rasterToMatch`	`SpatRaster` object covering the spatial extent of `studyArea`, used as a template for raster layer creation.
`type`	character. The polygon type to use: Must be one of "ECODISTRICT", "ECOREGION" (default), "ECOPROVINCE", "ECOZONE", "FRT", or "FRU". If `url` to BEC shapefile is provided, can also be one of: "BECNDT", "BECSUBZONE", or "BECZONE".

Examples

```
library(terra)
library(SpaDES.tools)

## random study area in central Alberta
studyAreaAB <- vect(cbind(-115, 55), crs = "epsg:4326") |>
  project(paste("+proj=lcc +lat_1=49 +lat_2=77 +lat_0=0 +lon_0=-95",
                "+x_0=0 +y_0=0 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0")) |>
  randomStudyArea(seed = 60, size = 1e10)

studyAreaBC <- vect(cbind(-122.14, 52.14), crs = "epsg:4326") |>
```

```
  project(paste("+proj=lcc +lat_1=49 +lat_2=77 +lat_0=0 +lon_0=-95",
                "+x_0=0 +y_0=0 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0")) |>
  randomStudyArea(seed = 60, size = 1e10)


frpEcoregion <- prepInputsFireRegimePolys(studyArea = studyAreaAB, type = "ECOREGION")
plot(frpEcoregion)



frpBECNDT <- prepInputsFireRegimePolys(studyArea = studyAreaBC, type = "BECNDT")
plot(frpBECNDT)


frpFRT <- prepInputsFireRegimePolys(studyArea = studyAreaAB, type = "FRT")
plot(frpFRT)

frpFRU <- prepInputsFireRegimePolys(studyArea = studyAreaAB, type = "FRU")
plot(frpFRU)
```

`genFireMapAttr` `scfmLandCoverInit: genFireMapAttr`

Description

`scfmLandCoverInit: genFireMapAttr`

Usage

```
genFireMapAttr(flammableMap, fireRegimePolys, neighbours)
```

Arguments

`flammableMap`	`SpatRaster` with values `0` indicating non-flammable pixels, `1` flammable.
`fireRegimePolys`	`sf` polygon or multipolygon object defining the fire regime polygons
`neighbours`	integer of value `4` or `8` specifying the number of neighbouring pixels

Value

TODO

genSimLand scfmDriver: genSimLand

Description

Buffers polygon, generates index raster

Usage

```
genSimLand(coreLand, buffDist, flammableMap = NULL)
```

Arguments

coreLand	TODO
buffDist	distance to buffer coreLand
flammableMap	SpatRaster with values 0 indicating non-flammable pixels, 1 flammable.

Value

list containing fireRegimePoly, landscapeIndex, flammableMap objects.

---

getFirePoints_NFDB_scfm *Get fire points data from the Canadian National Fire Database*

Description

Get fire points data from the Canadian National Fire Database

Usage

```
getFirePoints_NFDB_scfm(
  url = NULL,
  studyArea = NULL,
  rasterToMatch = NULL,
  redownloadIn = 2,
  NFDB_pointPath = NULL
)
```

Arguments

`url`	URL from which to download the fire points data. Default `NULL` fetches data from http://cwfis.cfs.nrcan.gc.ca/downloads/nfdb/fire_pnt/current_version/NFDB_point.zip.
`studyArea`	`sf` object corresponding to the study area of interest.
`rasterToMatch`	`SpatRaster` object covering the spatial extent of `studyArea`, used as a template for raster layer creation.
`redownloadIn`	time in years that we tolerate the data to be "old", and require redownload. I.e. 0.5 would mean "redownload data older than 6 months". Default 2.
`NFDB_pointPath`	file path to save the download data. Must be provided.

`hatP0` `scfmDriver`*: escape probability*

Description

```
1 - (1-p0)**N = pEscape
1 - pEscape = (1-p0)**N
(1 - pEscape)**1/N = 1 - p0
p0 = 1 - (1 - pEscape)**1/N
```

Usage

```
hatP0(pEscape, n = 8)

escapeProbDelta(p0, w, hatPE)
```

Arguments

`pEscape`	TODO
`n`	TODO
`p0`	TODO
`w`	TODO
`hatPE`	TODO

Value

TODO

---

makeAndExecuteDesign scfmDriver: makeAndExecuteDesign

Description

This is a wrapper around makeDesign and executeDesign.

Usage

```
makeAndExecuteDesign(...)
```

Arguments

...	objects to pass through to makeDesign and executeDesign.

Value

output of executeDesign (TODO)

---

makeDesign scfmDriver: makeDesign

Description

scfmDriver: makeDesign

Usage

```
makeDesign(indices, targetN, pEscape = 0.1, pmin, pmax, q = 1)
```

Arguments

indices	TODO
targetN	TODO
pEscape	TODO
pmin	TODO
pmax	TODO
q	TODO

Value

data.frame with columns igLoc, p0, and p.

Note

This version of makeDesign is the simplest possible.

plot_ageMap *Plot age map*

Description

Plot age map

Usage

```
plot_ageMap(x, title, maxAge)
```

Arguments

x	SpatRaster object corresponding to stand age or time since disturbance map
title	character, the plot title
maxAge	the maximum age to plot

Value

ggplot object

plot_burnMap *Plot burn maps*

Description

Plot burn maps

Usage

```
plot_burnMap(x, title)
```

Arguments

x	SpatRaster object corresponding to a current or cumulative burn map.
title	character, the plot title

Value

ggplot object

plot_fireRegimePolys *Plot fire regime polygons*

Description

Plot fire regime polygons

Usage

```
plot_fireRegimePolys(fireRegimePolys)
```

Arguments

fireRegimePolys	sf polygon or multipolygon object defining the fire regime polygons

Value

a ggplot object

plot_fireRegimeRas *Plot fire regime raster*

Description

Plot fire regime raster

Usage

```
plot_fireRegimeRas(x, title)
```

Arguments

x	SpatRaster object corresponding to a fire regime raster
title	character, the plot title

Value

ggplot object

plot_flammableMap *Plot flammable map*

Description

Plot flammable map

Usage

```
plot_flammableMap(x, title)
```

Arguments

x	SpatRaster object corresponding to a flammability map.
title	character, the plot title

Value

ggplot object

pTE *Various tools for a (right) Truncated Exponential Distribution*

Description

$ZTE(\theta, T)$: pdf is $\frac{\theta \exp(-z\theta)}{1-\exp(-T\theta)}$

Usage

```
pTE(x, theta, T)

dTE(x, theta, T)

qTE(p, theta, T)

rTE(n, theta, T)

qqTE(
  x,
  theta = stop("missing theta"),
  T = stop("missing T"),
  plot.it = TRUE,
  xlab = deparse(substitute(x)),
  ylab = deparse(substitute(y)),
  ...
```

```
)

ETE(theta, T, a = 0)

ETEx(theta, T, a = 0, shift = 200)

ExpBar(Z)

HannonDayiha(Z, Tspec = 0)
```

Arguments

x	TODO
theta	TODO
T	TODO
p	TODO
n	TODO
plot.it	logical indicating whether to plot the resulting figure
xlab, ylab	label text for x- and y-axes
...	additional arguments passed to plot()
a	TODO
shift	TODO
Z	TODO
Tspec	TODO

Details

where T is the truncation point or upper bound and θ is the shape parameter in this application, x are fire sizes >= shift, which is a lower bound and z = log(x / shift) are the scaled log transformed sizes which seem to fit a truncated exponential distribution fairly well.

Originally written by Steve in 1999 in support of Cumming CJFR 2001. Has been in use by BEACONs and was acquired from Pierre Vernier in May 17 2014.

- pTE() is the distribution function;
- dTE() is the density function;
- qTE() is the quantile function;
- rTE() is the random generation function. In fire size applications exp(rTE(n, theta, T)) * shift will generate n random fire sizes;
- qqTE() produces a quantile-quantile plot of vector x against a TE(theta, T);
- ETE() is TODO;
- ETEx() is TODO;
- ExpBar() is TODO;
- HannonDayiha() implements the estimator of Hannon and Dayiha (1999), ported from 1999 C language implementation by SGC June 2004.

References

Patrick M. Hannona & Ram C. Dahiyaa (1999) Estimation of parameters for the truncated exponential distribution. Communications in Statistics - Theory and Methods 28(11): 2591-2612. doi:10.1080/03610929908832440.

`ratioPartition` *Ratio partition*

Description

Ratio partition

Usage

```
ratioPartition(targetBurnRate, empiricalBurnRate, pEscape, xBar, rate)

ratioPartition2(targetBurnRate, empiricalBurnRate, pEscape, xBar, rate)
```

Arguments

`targetBurnRate`	target burn rate
`empiricalBurnRate`	burn rate estimated by the model
`pEscape`	escape probability
`xBar`	mean fire size
`rate`	ignition rate

Value

a list containing the following elements: `rate` (ignition rate), `pEscape` (escape probability), `xBar` (mean fire size).

`unirootFunction` `unirootFunction`

Description

`unirootFunction`

Usage

```
unirootFunction(x, cM, xBar)
```

Arguments

x	TODO
cM	TODO
xBar	TODO

Value

TODO

Index