

Package ‘rtop’

July 27, 2015

Type Package

Title Interpolation of Data with Variable Spatial Support

Version 0.5-5

Date 2015-07-27

Author Jon Olav Skoien

Maintainer Jon Olav Skoien <jon.skoien@gmail.com>

Depends sp

Imports gstat, graphics, stats, methods, utils, grDevices

Suggests parallel, igraph, rgdal, intamap, rgeos, spacetime,
data.table, reshape2

Description Geostatistical interpolation of data with irregular spatial support such as runoff related data or data from administrative units.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-07-27 10:52:13

R topics documented:

rtop-package	2
checkVario	4
createRtopObject	6
downloadRtopExampleData	7
gDist	8
getRtopParams	10
netProp	13
plot.rtopVariogramCloud	15
readAreaInfo	16
readAreas	17
rtopCluster	18
rtopDisc	19

rtopFitVariogram	21
rtopKrige	23
rtopSim	26
rtopVariogram	28
sceua	30
useRtopWithIntamap	33
variogramModel	34
varMat	35

Index	38
--------------	-----------

rtop-package	<i>A package providing methods for analysis and spatial interpolation of data with an irregular support</i>
--------------	---

Description

This package provides geostatistical methods for analysis and interpolation of data that has an irregular support, such as runoff characteristics or population health data. The methods in this package are based on the top-kriging approach suggested in Skoien et al (2006), with some extensions from Gottschalk (1993). This package can be used as an add-on package for the automatic interpolation package developed within the intamap project (www.intamap.org).

Workflow

The work flow within the package suggests that the user is interested in a prediction of a process at a series of locations where observations have not been made. The example below shows a regionalization of mean annual runoff in Austria.

Although it is possible to perform each step with all necessary arguments, the easiest interface to the method is to store all variables (such as observations, prediction locations and parameters) in an rtop-object, which is created by a call to `createRtopObject`. The element `params` below consists of changes to the default parameters. A further description can be found in `getRtopParams`. The changes below means that the functions will use geostatistical distance instead of full regularization, and that the variogram model will be fitted to the variogram cloud. Most other functions in the rtop-package can take this object as an argument, and will add the results as one or more new element(s) to this object.

The data in the example below are stored as shape-files in the `extdata`-directory of the rtop-package, use the directory of your own data instead. The observations consist of mean summer runoff from 138 catchments in Upper Austria. The predictionLocations are 863 catchments in the same region. observations and predictionLocations are stored as `SpatialPolygonsDataFrame`-objects.

```
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")

# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM
```

```
predictionLocations = readOGR(rpath,"predictionLocations")
params = list(gDist = TRUE, cloud = TRUE)
rtopObj = createRtopObject(observations, predictionLocations,
                          params = params)
```

There are help-methods available in cases when data are not available as shape-files, or when the observations are not part of the shape-files. See [readAreaInfo](#) and [readAreas](#).

A call to [rtopVariogram](#) adds the sample variogram to the object, whereas [rtopFitVariogram](#) fits a variogram model. The last function will call [rtopVariogram](#) if `rtopObj` does not contain a sample variogram.

```
rtopObj = rtopVariogram(rtopObj)
rtopObj = rtopFitVariogram(rtopObj)
```

The function [checkVario](#) is useful to produce some diagnostic plots for the sample variogram and the fitted variogram model.

```
checkVario(rtopObj)
```

The interpolation function ([rtopKrige](#)) solves the kriging system based on the computed regularized semivariances. The covariance matrices are created in a separate regularization function ([varMat](#)), and are stored in the rtop-object for easier access if it is necessary to redo parts of the analysis, as this is the computationally expensive part of the interpolation. Cross-validation can be called with the argument `cv=TRUE`, either in `params` or in the call to [rtopKrige](#).

```
rtopObj = rtopKrige(rtopObj)
splot(rtopObj$predictions, col.regions = bpy.colors(),
      c("var1.pred", "var1.var"))
rtopObj = rtopKrige(rtopObj, cv = TRUE)
splot(rtopObj$predictions, col.regions = bpy.colors(),
      c("var1.pred", "var1.var"))
```

References

L. Gottschalk. Interpolation of runoff applying objective methods. *Stochastic Hydrology and Hydraulics*, 7:269-281, 1993.

Skoien J. O., R. Merz, and G. Bloschl. Top-kriging - geostatistics on stream networks. *Hydrology and Earth System Sciences*, 10:277-287, 2006.

 checkVario

Plot variogram fitted to data with support

Description

The function will create diagnostic plots for analysis of the variograms fitted to sample variograms of data with support

Usage

```
## S3 method for class 'rtop'
checkVario(object, acor = 1, log = "xy", cloud = FALSE,
           gDist = TRUE, params = list(), ...)

## S3 method for class 'rtopVariogramModel'
checkVario(object,
           sampleVariogram = NULL, observations = NULL,
           areas = NULL, dists = NULL, acomp = NULL,
           params = list(), compVars = list(), acor = 1,
           log = "xy", legx = NULL, legy = NULL,
           plotNugg = TRUE, ...)
```

Arguments

object	either: object of class <code>rtop</code> (see rtop-package), or an object of type <code>rtopVariogram</code>
acor	unit correction factor in the key, e.g. to see numbers more easily interpretable for large areas. As an example, <code>ucor = 0.000001</code> when area is given in square meters and should rather be shown as square kilometers. Note that this parameter also changes the value of the nugget to the new unit.
log	text variable for log-plots, default to log-log "xy", can otherwise be set to "x", "y" or ""
cloud	logical; whether to look at the cloud variogram instead of the binned variogram
gDist	logical; whether to use ghosh-distance for semivariogram regularization instead of full integration of the semivariogram
sampleVariogram	a sample variogram of the data
observations	a set of observations
areas	either an array of areas that should be used as examples, or the number of areas per order of magnitude (similar to the parameter <code>amul</code> of the standard parameters, see getRtopParams . <code>amul</code> from <code>rtopObj</code> or from the standard parameter set will be used if not defined here.
dists	either an array of distances that should be used as examples, or the number of distances per order of magnitude (similar to the parameter <code>amul</code> of the standard parameters, see getRtopParams . <code>amul</code> from <code>rtopObj</code> or from the standard parameter set will be used if not defined here.

acomp	either a matrix with the area bins that should be visualized, or a number giving the number of pairs to show. If a sample variogram is given, the acomp pairs with highest number of pairs will be used
params	list of parameters to modify the standard parameters of rtopObj or the default parameters found from getRtopParams
compVars	a list of variograms of gstat-type for comparison, see vgm . The names of the variograms in the list will be used in the key.
legx	x-coordinate of the legend for fine-tuning of position, see x-argument of Legend
legy	y-coordinate of the legend for fine-tuning of position, see y-argument of Legend
plotNugg	logical; whether the nugget effect should be added to the plot or not
...	arguments to lower level functions

Value

The function gives diagnostic plots for the fitted variograms, where the regularized variograms are shown together with the sample variograms and possibly also user defined variograms. In addition, if an rtopObject is submitted, the function will also give plots of the relationship between variance and area size and a scatter plot of the fit of the observed and regularized variogram values. The sizes of the dots are relative to the number of pairs in each group.

Author(s)

Jon Olav Skoien

See Also

[rtop-package](#)

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
setwd(rpath)
observations = readOGR(".", "observations")
# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM
predictionLocations = readOGR(".", "predictionLocations")
params = list(cloud = TRUE, gDist = TRUE)
rtopObj = createRtopObject(observations, predictionLocations,
                           params = params)

# Fit a variogram (function also creates it)
rtopObj = rtopFitVariogram(rtopObj)
checkVario(rtopObj,
           compVar = list(first = vgm(5e-6, "Sph", 30000,5e-8),
                          second = vgm(2e-6, "Sph", 30000,5e-8)))
```

```

rtopObj = checkVario(rtopObj, acor = 0.000001,
  acomp = data.frame(ac11 = c(2,2,2,2,3,3,3,4,4),
    ac12 = c(2,3,4,5,3,4,5,4,5)))
rtopObj = checkVario(rtopObj, cloud = TRUE, identify = TRUE,
  acor = 0.000001)

## End(Not run)

```

createRtopObject *Create an object for interpolation within the rtop package*

Description

This is a help function for creating an object (see [rtop-package](#) to be used for interpolation within the rtop package)

Usage

```

createRtopObject(observations, predictionLocations,
  formulaString, params=list(), ainfo, areas, overlapObs,
  overlapPredObs, ...)

```

Arguments

observations	SpatialPolygonsDataFrame with observations
predictionLocations	a SpatialPolygons or a SpatialPolygonsDataFrame -object with prediction locations
formulaString	formula that defines the dependent variable as a linear model of independent variables; suppose the dependent variable has name z, for ordinary and simple kriging use the formula $z \sim 1$; for universal kriging, suppose z is linearly dependent on x and y, use the formula $z \sim x + y$. The formulaString defaults to "value~1" if value is a part of the data set. If not, the first column of the data set is used. Universal kriging is not yet properly implemented in the rtop-package, this element is mainly used for defining the dependent variable.
params	parameters to modify the standard parameters of the rtop-package, set internally in this function by a call to getRtopParams
ainfo	SpatialPointsDataFrame with information about the observations. Only used here if it contains information for separating the observations into observations and predictionLocations or combining ainfo with areas to find observations and predictionLocations(deprecated method). Typically from a call to readAreaInfo
areas	SpatialPolygonsDataFrame with areal information, typically from a call to readAreas , deprecated
overlapObs	matrix with observations that overlap each other
overlapPredObs	matrix with observations and predictionLocations that overlap each other
...	Extra parameters to getRtopParams and possibility to pass deprecated arguments

Arguments

folder the folder to which the downloaded data set will be copied

Value

The function will have as a side effect that additional example data is downloaded from Vienna University of Technology. This will for the default case replace the existing example data-set in the rtop package. Alternatively the user can specify a separate directory for the data set.

Author(s)

Jon Olav Skoien

Examples

```
## Not run:
downloadRtopExampleData()
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")

## End(Not run)
```

gDist

calculate geostatistical distances between areas

Description

Calculate geostatistical distances (Ghosh-distances) between areas

Usage

```
## S3 method for class 'rtop'
gDist(object, params = list(), ...)
## S3 method for class 'SpatialPolygonsDataFrame'
gDist(object, object2 = NULL, ...)
## S3 method for class 'SpatialPolygons'
gDist(object, object2 = NULL, ...)
## S3 method for class 'list'
gDist(object, object2 = NULL, diag = FALSE, debug.level = 0, ...)
```

Arguments

object object of class [SpatialPolygons](#) or [SpatialPolygonsDataFrame](#) with boundaries of areas; or list of discretized areas, typically from a call to [rtopDisc](#); or object of class rtop with such boundaries and/or discretized elements (the individual areas)

params	a set of parameters, used to modify the standard parameters for the <code>rtop</code> package, set in <code>getRtopParams</code> . The argument <code>params</code> can also be used for the other methods, through the <code>...</code> -argument.
object2	an object of same type as <code>object</code> , except for <code>rtop</code> ; for calculation of geostatistical distances also between the elements in the two different objects
diag	logical; if TRUE only calculate the geostatistical distances between each element and itself, only when the objects are lists of discretized areas and <code>object2 = object</code> or <code>object2 = NULL</code>
debug.level	<code>debug.level = 0</code> will suppress output from the call to <code>varMat</code> , done for calculation of the geostatistical distances
...	other parameters, for <code>gDist.list</code> when calling one of the other methods, or for <code>varMat</code> , in which the calculations take place

Value

If called with one list of discretized elements, a matrix with the geostatistical distances between the elements within the list. If called with two lists of discretized elements, a matrix with the geostatistical distances between the elements in the two lists. If called with `diag = TRUE`, the function returns an array of the geostatistical distance within each of the elements in the list.

If called with one `SpatialPolygons` or `SpatialPolygonsDataFrame` or the function returns a list with one matrix with geostatistical distances between the elements of the object. If called with two objects, the list will also contain a matrix of the geostatistical distances between the elements of the two objects, and an array of the geostatistical distances within the elements of the second object.

If called with an `rtop`-object, the function will return the object, amended with the list above.

Note

The geostatistical distance can be seen as the average distance between points in two elements, or the average distance within points in a single element. The distance measure is also sometimes referred to as Ghosh-distance, from Ghosh (1951) who found analytical expressions for these distances between blocks with regular geometry.

The use of geostatistical distances within `rtop` is based on an idea from Gottschalk (1993), who suggested to replace the traditional regularization of variograms within block-kriging (as done in the original top-kriging application of Skoien et al (2006)) with covariances of the geostatistical distance. The covariance between two areas can then be found as $C(a1, a2) = cov(gd)$ where `gd` is the geostatistical distance between the two areas `a1` and `a2`, instead of an integration of the covariance function between the two areas.

`rtop` is based on semivariograms instead of covariances, and the semivariogram value between the two areas can be found as $gamma(a1, a2) = g(gd) - 0.5 (g(gd1) + g(gd2))$ where `g` is a semivariogram valid for point support, `gd1` and `gd2` are the geostatistical distances within each of the two areas.

Author(s)

Jon Olav Skoien

References

- Ghosh, B. 1951. Random distances within a rectangle and between two rectangles. Bull. Calcutta Math. Soc., 43, 17-24.
- Gottschalk, L. 1993. Correlation and covariance of runoff. Stochastic Hydrology and Hydraulics, 7, 85-101.
- Skoien, J. O., R. Merz, and G. Bloschl. 2006. Top-kriging - geostatistics on stream networks. Hydrology and Earth System Sciences, 10, 277-287.

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
gDist = gDist(observations)

## End(Not run)
```

getRtopParams

Setting parameters for the intamap package

Description

This function sets a range of the parameters for the intamap package, to be included in the object described in [rtop-package](#)

Usage

```
getRtopParams(params,newPar, observations, formulaString, ...)
```

Arguments

- | | |
|---------------|--|
| params | An existing set of parameters for the interpolation process, of class <code>intamapParams</code> or a list of parameters for modification of the default parameters |
| newPar | A list of parameters for updating <code>params</code> or for modification of the default parameters. Possible parameters with their defaults are given below |
| observations | SpatialPolygonsDataFrame with observations, used for setting some of the default parameters |
| formulaString | formula that defines the dependent variable as a linear model of independent variables, see e.g. createRtopObject for more details. |
| ... | Individual parameters for updating <code>params</code> or for modification of the default parameters. Possible parameters with their defaults are given below <ul style="list-style-type: none"> • <code>model = "Ex1"</code> - variogram model type. Currently the following models are implemented: |

- Exp - Exponential model
- Ex1 - Multiplication of a modified exponential and fractal model, the same model as used in Skoien et al(2006).
- Gau - Gaussian model
- Ga1 - Multiplication of gaussian and fractal model
- Sph - Spherical model
- Sp1 - Multiplication of spherical and fractal model
- Fra - Fractal model
- parInit - the initial parameters and the limits of the variogram model to be fitted, given as a matrix with three columns, where the first column is the lower limit, the second column is the upper limit and the third column are starting values.
- nugget = TRUE - logical; if TRUE nugget effect should be estimated
- unc = TRUE - logical; if TRUE the standard deviation of observations are in column unc
- rresol = 100 - minimum number of discretization points in each area
- hresol = 5 - number of discretization points in one direction for elements in binned variograms
- cloud = FALSE - logical; if TRUE use the cloud variogram for variogram fitting
- amul = 1 - defines the number of areal bins within one order of magnitude. Numbers between 1 and 3 are possible, as this parameter refers to the axp parameter of [axTicks](#).
- dmul = 3 - defines the number of distance bins within one order of magnitude. Numbers between 1 and 3 are possible, as this parameter refers to the axp parameter of [axTicks](#).
- fit.method = 8 - defines the type of Least Square method for fitting of variogram. The methods 1-7 correspond to the similar methods in [gstat](#).
 - 1 - weighted least squares with number of pairs per bin:
 $err = n * (yobs - ymod)^2$
 - 2 - weighted least squares difference according to Cressie (1985):
 $err2 = abs(yobs/ymod - 1)$
 - 6 - ordinary least squares difference: $err = (yobs - ymod)^2$
 - 7 - similar to default of [gstat](#), where higher weights are given to shorter distances $err = n/h^2 * (yobs - ymod)^2$
 - 8 - Opposite of weighted least squares difference according to Cressie (1985): $err3 = abs(ymod/yobs - 1)$
 - 9 - neutral WLS-method - $err = min(err2, err3)$
- gDistEst = FALSE - use geostatistical distance when fitting variograms
- gDistPred = FALSE - use geostatistical distance for semivariogram matrices
- gDist - parameter to set jointly $gDistEst = gDistPred = gDist$
- nmax = 10 for local kriging: the number of nearest observations that should be used for a kriging prediction or simulation, where nearest is defined in terms of the space of the spatial locations. By default, 10 observations are used.

- `maxdist = Inf` - for local kriging: only observations within a distance of `maxdist` from the prediction location are used for prediction or simulation; if combined with `nmax`, both criteria apply
- `hstype = "regular"` - sampling type for binned variograms
- `rstype = "rtop"` - sampling type for the elements, see also [rtopDisc](#)
- `nclus = 1` - number of CPUs to use if parallel processing is wanted; `nclus = 1` means no parallelization
- `cnAreas = 100` - limit whether parallel processing should be applied; the minimum number of areas in [varMat](#), and also controlling when to use parallel processing in [rtopDisc](#), when $nAreas * params\$rresol / 100 > cnAreas$
- `clusType = NULL` - the cluster type to be started for parallel processing; uses the default type of the system when `clusType = NULL`
- `wlim = 1.5` - an upper limit for the norm of the weights in kriging, see [rtopKrige](#)
- `wlimMethod = "all"` which method to use for reducing the norm of the weights if necessary. Either "all", which modifies all weights equally or "neg" which reduces negative weights and large weights more than the smallest weights
- `cv = FALSE` - logical; for cross-validation of observations
- `debug.level = 1` - used in some functions for giving additional output. See individual functions for more information.
- `partialOverlap = FALSE` whether to work with partially overlapping areas
- `olim = 1e-4` smallest overlapping area to be used for partial overlap, relative to the smallest of the areas
- `nclus = 1` option to use parallel processing, `nclus > 1` defines the number of workers to be started
- `clusType = NA` which cluster type to start if `nclus > 1`; the default is used if `nclusType = NA`
- `cnAreas = 200` The minimum number of observations or observations plus predictions allowing parallelization in the creation of the covariance matrix
- `cDlim = 1e6` The minimum number of discretization points for allowing parallelization in the discretization process
- `observations` - used for initial values of parameters if supplied
- `formulaString` - used for initial values of parameters if supplied

Value

A list of the parameters with class `rtopParams` to be included in the object described in [rtop-package](#)

Note

This function will mainly be called by [createRtopObject](#), but can also be called by the user to create a parameter set or update an existing parameter set. If none of the arguments is a list of class `rtopParams`, the function will assume that the argument(s) are modifications to the default set

of parameters. The function can also be called by other functions in the rtop-package if the users chooses not to work with an object of class rtop.

If the function is called with two lists of parameters (but the first one is not of class rtopParams) they are both seen as modifications to the default parameter set. If they share some parameters, the parameter values from the second list will be applied.

Parallel processing has been included for some of the functions. The default is no parallel procesing, and the package also attempts to decide whether it is sensible to start a set of clusters and distribute jobs to them based on the size of the job. The default limit might not be the best for every system.

Author(s)

Jon Olav Skoien

References

Cressie, N. 1985. Fitting variogram models by weighted least squares. *Mathematical Geology*, 17 (5), 563-586

Skoien J. O., R. Merz, and G. Bloschl. Top-kriging - geostatistics on stream networks. *Hydrology and Earth System Sciences*, 10:277-287, 2006

See Also

[createRtopObject](#) and [rtop-package](#)

Examples

```
# Create a new set of intamapParameters, with default parameters:
params = getRtopParams()
# Make modifications to the default list of parameters
params = getRtopParams(newPar = list(gDist = TRUE, nugget = FALSE))
# Make modifications to an existing list of parameters
params = getRtopParams(params = params, newPar = list(gDist = TRUE,
  nugget = FALSE))
```

netProp

Propagate values along a river network. This function does not work as intended for several cases, including the example. It is therefore deprecated and will be removed in one of the next versions.

Description

Pass values along a river network when the river network has more segments than the prediction polygons.

Usage

```
netProp(network, from = "FROMJCT", to = "TOJCT", pred = "pred",
  iprint = 1)
```

Arguments

network	object of class SpatialLinesDataFrame describing the river network
from	name of the column giving the endpoint ID of each line segment
to	name of the column giving the start ID of each line segment
pred	name of column with predictions
iprint	if iprint >= 1 the function will give some information about the convergence of the value propagation. Use iprint = 0 to suppress this output.

Value

The function will propagate the predictions upwards along the river network. The result is a [SpatialLinesDataFrame](#) with predictions for all line segments, which can easier be plotted.

Note

This function works when the topology of the river network is similar to the example here, that the from-column is always the upstream part of a river segment, and that all segments are actually connected.

Author(s)

Jon Olav Skoien

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
predictionLocations = readOGR(rpath,"predictionLocations")
observations$obs = observations$QSUMMER/observations$AREASQKM

# Setting some parameters
params = list(geoDist = TRUE, rresol = 25, cloud = FALSE, model = "Sph")
# Build an object
rtopObj = createRtopObject(observations,predictionLocations,
                          formulaString = obs~1, params = params)
# Fit a variogram (function also creates it)
rtopObj = rtopFitVariogram(rtopObj)
# Check the variogram fit
rtopObj = checkVario(rtopObj, cloud = TRUE, identify = TRUE)
# Predicting at prediction locations
rtopObj = rtopKrige(rtopObj)
# Cross-validation
rtopObj = rtopKrige(rtopObj,cv=TRUE)
cor(rtopObj$predictions$observed, rtopObj$predictions$var1.pred)

rnet = readOGR(".", "rnet")
pred = rtopObj$predictions
rnet$pred =
```

```

    pred$var1.pred[match(rnet$TOJCT, pred$JCTID)]

# will only plot for a few discontinous river segments
spplot(rnet, "pred", col.regions = bpy.colors())
rnet = netProp(rnet)
# will show a prediction for all segments
spplot(rnet, "pred", col.regions = bpy.colors())

## End(Not run)

```

```
plot.rtopVariogramCloud
```

Plot and Identify Data Pairs on Sample Variogram Cloud

Description

Plot a sample variogram cloud, possibly with identification of individual point pairs

Usage

```
## S3 method for class 'rtopVariogramCloud'
plot(x, ...)
```

Arguments

x	object of class <code>variogramCloud</code>
...	parameters that are passed through to plot.variogramCloud The most important are: <ul style="list-style-type: none"> • <code>identify</code> logical; if TRUE, the plot allows identification of a series of individual point pairs that correspond to individual variogram cloud points (use left mouse button to select; right mouse button ends) • <code>digitize</code> logical; if TRUE, select point pairs by digitizing a region with the mouse (left mouse button adds a point, right mouse button ends) • <code>xlim</code> limits of x-axis • <code>ylim</code> limits of y-axis • <code>xlab</code> x axis label • <code>ylab</code> y axis label • <code>keep</code> logical; if TRUE and <code>identify</code> is TRUE, the labels identified and their position are kept and glued to object x, which is returned. Subsequent calls to plot this object will now have the labels shown, e.g. to plot to hardcopy

Note

This function is mainly a wrapper around [plot.variogramCloud](#), necessary because of different column names and different class names. The description of arguments and value can therefore be found in the help page of [plot.variogramCloud](#).

Author(s)

Jon Olav Skoien

References<http://www.gstat.org/>**See Also**[plot.gstatVariogram](#)**Examples**

```
## Not run:
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
observations$obs = observations$QSUMMER_OB/observations$AREASQKM

# Create the sample variogram
rtopVario = rtopVariogram(observations, params = list(cloud = TRUE))
plot(rtopVario)

## End(Not run)
```

readAreaInfo	<i>create SpatialPointsDataFrame with observations of data with a spatial support</i>
--------------	---

Description

readAreaInfo will read a text file with observations and descriptions of data with a spatial support.

Usage

```
readAreaInfo(fname = "ainfo.txt", id = "id",
             iobs = "iobs", obs = "obs", unc = "unc",
             filenames= "filenames", sep = "\t",
             debug.level = 1, moreCols = list(NULL))
```

Arguments

fname	name of file with areal information
id	name of column with observation id
iobs	name of column with number of observations
obs	name of column with observations
unc	name of column with possible uncertainty of observation

filenames	name of column with filenames of areas if different names than id should be used.
sep	separator in csv-file
debug.level	used for giving additional output
moreCols	name of other column names the user wants included in ainfo

Details

The function is of particular use when data are not available as shape-files, or when the observations are not part of the shape-files. This function is mainly for compatibility with the former FORTRAN-version. The simplest way to read the data in that case is through `readShapePoly` in the `maptools`-package or `readOGR` in the `rgdal`-package. See also [rtop-package](#).

Value

[SpatialPointDataFrame](#) with information about observations and/or `predictionLocations`.

Author(s)

Jon Olav Skoien

readAreas	<i>help file for creating SpatialPolygonsDataFrame with observations and/or predictionLocations of data with a spatial support</i>
-----------	--

Description

`readAreas` will read area-files, add observations and convert the result to [SpatialPolygonsDataFrame](#)

Usage

```
readAreas(object, adir=".", ftype = "xy", projection = NA, ...)
```

Arguments

object	either name of file with areal information or SpatialPointsDataFrame with observations
adir	directory where the files with areal information are to be found
ftype	type of file, the only type supported currently is "xy", referring to x- and y-coordinates of boundaries
projection	add projection to the object if input is boundary-files
...	further parameters to be passed to readAreaInfo

Details

If object is a file name, [readAreaInfo](#) will be called. If it is a [SpatialPointsDataFrame](#) with observations and/or predictionLocations, the function will read areal data from files according to the ID associated with each observation/predictionLocation.

The function is of particular use when data are not available as shape-files, or when the observations are not part of the shape-files. This function is mainly for compatibility with the former FORTRAN-version. The simplest way to read the data in that case is through [readShapePoly](#) in the [mapproj](#)-package or [readOGR](#) in the [rgdal](#)-package. See also [rtop-package](#).

Value

The function creates a [SpatialPolygonsDataFrame](#) of observations and/or predictionLocations, depending on the information given in object.

Author(s)

Jon Olav Skoien

rtopCluster	<i>start, access, stop or restart a cluster for parallel computation with rtop</i>
-------------	--

Description

Convenience function for using parallel computation with rtop. The function is usually not called by the user.

Usage

```
rtopCluster(nclus, ..., action = "start", type)
```

Arguments

nclus	The number of workers in the cluster
...	Arguments for clusterEvalQ ; commands to be evaluated for each worker, such as library-calls
action	Defines the action of the function. There are three options: <ul style="list-style-type: none"> • "start" Starts a new cluster if necessary, reuses an existing if it has already been started • "restart" Stops the cluster and starts it again. To be used in case there are difficulties with the cluster, or if the user wants to change the type of the cluster
type	The type of cluster; see makeCluster for more details. The default of makeCluster is used if type is missing or NA.

Details

It is usually not necessary for the user to call this function for starting or accessing a cluster. This is done automatically by the different rtop-functions when needed if the parameter nclus is larger than one (see [getRtopParams](#)). If the user actually starts the cluster by a call to this function, it will also be necessary to set the nclus parameter to a value larger than one for the cluster to be used by different functions.

Restarting the cluster might be necessary if the cluster has a problem (e.g. does not return memory) or if the user wants to change to a different cluster type.

Stopping the cluster is useful when the user does not want to continue with parallel computation and wants to close down the workers.

Value

If the function is called with action equal to "start" or "restart", the result is a cluster with nclus workers. The cluster is also added to the global options with the name rtopCluster (`getOption("rtopCluster")`).

If the function is called with action equal to "stop", the function stops the cluster, sets the rtopCluster of options to NULL and returns NULL to the user.

Author(s)

Jon Olav Skoien

 rtopDisc

Discretize areas

Description

rtopDisc will discretize an area for regularization or calculation of Ghosh-distance

Usage

```
## S3 method for class 'rtop'
rtopDisc(object, params = list(),...)
## S3 method for class 'SpatialPolygonsDataFrame'
rtopDisc(object, params = list(), bb = bbox(object), ...)
## S3 method for class 'SpatialPolygons'
rtopDisc(object, params = list(), bb = bbox(object), ...)
## S3 method for class 'rtopVariogram'
rtopDisc(object, params = list(), ...)
```

Arguments

object	object of class SpatialPolygons or SpatialPolygonsDataFrame or rtopVariogram , or an object with class <code>rtop</code> that includes one of the above
bb	boundary box, usually modified to be the common boundary box for two spatial object
params	possibility to pass parameters to modify the standard parameters for the <code>rtop</code> package, set in getRtopParams . Typical parameters to modify for this function are: <ul style="list-style-type: none"> • <code>rresol = 100</code>; minimum number of discretization points in areas • <code>hresol = 5</code>; number of discretization points in one direction for areas in binned variograms • <code>hstype = "regular"</code>; sampling type for binned variograms • <code>rstype = "rtop"</code>; sampling type for real areas
...	Possibility to pass individual parameters

Details

There are different options for discretizing the objects. When the areas from the bins are discretized, the options are random or regular sampling, regular sampling is the default.

For the real areas, regular sampling appears to have computational advantages compared with random sampling. In addition to the traditional regular sampling, `rtop` also offers a third type of sampling which assures that the same discretization points are used for overlapping areas.

Starting with a coarse grid covering the region of interest, this will for a certain support be refined till a requested minimum number of points from the grid is within the support. In this way, for areal supports, the number of points in the area with the largest number of points will be approximately four times the requested minimum number of points. This methods also assure that points used to discretize a large support will be reused when discretizing smaller supports within the large one, e.g. subcatchments within larger catchments.

Value

The function returns a list of discretized areas, or if called with an `rtop`-object as argument, the object with lists of discretizations of the observations and prediction locations (if part of the object). If the function is called with an `rtopVariogram` (usually this is an internal call), the list contains discretized pairs of hypothetical objects from each bin of the semivariogram with a centre-to-centre distance equal to the average distance between the objects in a certain bin.

Author(s)

Jon Olav Skoien

See Also

[rtop-package](#) and [rtopVariogram](#)

rtopFitVariogram *Fit variogram model to sample variogram of data with spatial support*

Description

rtopFitVariogram will fit a variogram model to the estimated binned variogram or cloud variogram of data with an areal support.

Usage

```
## S3 method for class 'rtop'
rtopFitVariogram(object, params = list(), ...)
## S3 method for class 'SpatialPolygonsDataFrame'
rtopFitVariogram(object, params=list(), ...)
## S3 method for class 'SpatialPointsDataFrame'
rtopFitVariogram(object, params=list(), ...)
## S3 method for class 'rtopVariogram'
rtopFitVariogram(object, observations, dists = NULL,
                  params=list(), mr = FALSE, aOver = NULL, ...)
## S3 method for class 'rtopVariogramCloud'
rtopFitVariogram(object, observations, dists = NULL,
                  aOver = NULL, params=list(), mr = FALSE, ...)
```

Arguments

object	object of class rtopVariogram or rtopVariogramCloud, or an object with class rtop that includes the sample variograms. The object can also be of class SpatialPolygonsDataFrame or SpatialPointsDataFrame with observations. If object is a SpatialPointsDataFrame , it must have a column with name area.
observations	the observations, passed as a Spatial*DataFrame object, if object is an rtopVariogram or rtopVariogramCloud
params	a set of parameters, used to modify the standard parameters for the rtop package, set in getRtopParams . The argument params can also be used for the other methods, through the ...-argument.
dists	either a matrix with geostatistical distances (created by a call to the function gDist or a list with the areas discretized (from a call to rtopDisc .
mr	logical; defining whether the function should return a list with discretized elements and geostatistical distances, even if it was not called with an rtop-object as argument.
aOver	a matrix with the overlapping areas of the observations, used for computation of the nugget effect. It will normally be recomputed by the function if it is NULL and necessary
...	Other parameters to functions called from rtopFitVariogram

Value

The function creates an object with the fitted variogram Model (`variogramModel`) and a `data.frame` (`varFit`) with the differences between the sample semivariances and the regularized semivariances. If `mr = TRUE`, the function also returns other objects (discretized elements and geostatistical distances, if created) as a part of the returned object. If the function is called with an `rtop`-object as argument, it will return an `rtop`-object with `variogramModel` and `varFit` added to the object, in addition to other objects created.

Note

There are several options for fitting of the `variogramModel`, where the parameters can be set in `params`, which is a list of parameters for modification of the standard parameters of the `rtop`-package given in a call to `getRtopParams`. The first choice is between individual fitting and binned fitting. This is based on the type of variogram submitted, individual fitting is done if a cloud variogram (of class `rtopVariogramCloud`) is passed as argument, and binned fitting if the submitted variogram is of class `rtopVariogram`. If the function is called with an object of class `rtop`, having both `variogram` and `variogramCloud` among its arguments, the variogram model is fitted to the variogram which is consistent with the parameter `cloud`.

Author(s)

Jon Olav Skoien

References

<http://www.intamap.org/>

See Also

[rtop-package](#)

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM
predictionLocations = readOGR(rpath,"predictionLocations")

# Setting some parameters
params = list(gDist = TRUE, cloud = FALSE)
# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM
# Build an object
rtopObj = createRtopObject(observations,predictionLocations,
                           params = params)
# Fit a variogram (function also creates it)
rtopObj = rtopFitVariogram(rtopObj)
```

```

rtopObj$variogramModel

## End(Not run)

```

rtopKrige	<i>Spatial interpolation of data with spatial support</i>
-----------	---

Description

rtopKrige perform spatial interpolation or cross validation of data with areal support.

Usage

```

## S3 method for class 'rtop'
rtopKrige(object, varMatUpdate = FALSE, params = list(), ...)
## S3 method for class 'SpatialPolygonsDataFrame'
rtopKrige(object, predictionLocations = NULL,
           varMatObs, varMatPredObs, varMat, params = list(),
           formulaString, sel, ...)
## S3 method for class 'STSDf'
rtopKrige(object, predictionLocations = NULL,
           varMatObs, varMatPredObs, varMat, params = list(),
           formulaString, sel, olags = NULL, plags = NULL,
           lagExact = TRUE, ...)
## Default S3 method:
rtopKrige(object, predictionLocations = NULL,
           varMatObs, varMatPredObs, varMat, params = list(),
           formulaString, sel, wret = FALSE, ...)

```

Arguments

object	object of class rtop or SpatialPolygonsDataFrame
varMatUpdate	logical; if TRUE, also existing variance matrices will be recomputed, if FALSE, only missing variance matrices will be computed, see also varMat
predictionLocations	SpatialPolygons or SpatialPolygonsDataFrame with prediction locations. NULL if cross validation is to be performed.
varMatObs	covariance matrix of observations, where diagonal must consist of internal variance, typically generated from call to varMat
varMatPredObs	covariance matrix between observation locations and prediction locations, typically generated from call to varMat
varMat	list covariance matrices including the two above

params	a set of parameters, used to modify the standard parameters for the rtop package, set in getRtopParams . Additionally, it is possible overrule some of the parameters in <code>object\$params</code> by passing them as separate arguments.
formulaString	formula that defines the dependent variable as a linear model of independent variables, see e.g. createRtopObject for more details.
sel	array of prediction location numbers, if only a limited number of locations are to be interpolated/crossvalidated
wret	logical; if TRUE, return a matrix of weights instead of the predictions, useful for batch processing of time series, see also details
olags	A vector describing the relative lag which should be applied for the observation locations. See also details
plags	A vector describing the relative lag which should be applied for the prediction locations. See also details
lagExact	logical; whether differences in lagtime should be computed exactly or approximate
...	from <code>rtopKrige.rtop</code> , arguments to be passed to <code>rtopKrige.default</code> . In <code>rtopKrige.default</code> , parameters for modification of the object parameters or default parameters. Of particular interest are <code>cv</code> , a logical for doing cross-validation, <code>nmax</code> , and <code>maxdist</code> for maximum number of neighbours and maximum distance to neighbours, respectively, and <code>wlim</code> , the limit for the absolute values of the weights.

Details

This function is the interpolation routine of the rtop-package. The simplest way of calling the function is with an rtop-object that contains the fitted variogram model and all the other necessary data (see [createRtopObject](#) or [rtop-package](#)).

The function will, if called with covariance matrices between observations and between observations and prediction locations, use these for the interpolation. If the function is called without these matrices, `varMat` will be called to create them. These matrices can therefore be reused if necessary, an advantage as it is computationally expensive to create them.

The interpolation that takes part within `rtopKrige.default` is based on the semivariance matrices between observations and between observations and prediction locations. It is therefore possible to use this function also to interpolate data where the matrices have been created in other ways, e.g. based on distances in geographical space or distances along a stream.

The function returns the weights rather than the predictions if `wret = TRUE`. This is useful for batch processing of time series, e.g. once the weights are created, they can be used to compute the interpolated values for each time step.

rtop is able to take some advantage of multiple cpus, which can be invoked with the parameter `nclus`, setting this larger than one will start a cluster with `nclus` workers, if the `parallel`-package has been installed.

Kriging of time series is possible when observations and predictionLocations are spatiotemporal objects of type `STSDF`. The interpolation is still spatial, in the sense that the regularisation of the variograms are just done using the spatial extent of the observations, not a possible temporal extent, such as done by Skoien and Bloschl (2007). However, it is possible to make predictions

based on observations from different time steps, through the use of the lag-vectors. These vectors describe a typical "delay" for each observation and prediction location. This delay could for runoff related variables be similar to travel time to each gauging location. For a certain prediction location, earlier time steps would be picked for neighbours with shorter travel time and later time steps for neighbours with slower travel times.

The lagExact parameter indicates whether to use a weighted average of two time steps, or just the time step which is closest to the difference in lag times.

The use of lag times should in theory increase the computation time, but might, due to different computation methods, even speed up the computation when the number of neighbours to be used (parameter nmax) is small compared to the number of observations. If computation is slow, it can be useful to test `olags = rep(0, dim(observations)[1])` and similar for predictionLocations.

Value

If called with [SpatialPolygonsDataFrame](#), the function returns a [SpatialPolygonsDataFrame](#) with predictions, either at the locations defined in predictionLocations, or as leave-one-out cross-validation predicitions at the same locations as in object if `cv = TRUE`

If called with an rtop-object, the function returns the same object with the predictions added to the object.

Author(s)

Jon Olav Skoien

References

Skoien J. O., R. Merz, and G. Bloschl. Top-kriging - geostatistics on stream networks. *Hydrology and Earth System Sciences*, 10:277-287, 2006.

Skoien, J. O. and G. Bloschl. Spatio-Temporal Top-Kriging of Runoff Time Series. *Water Resources Research* 43:W09419, 2007.

See Also

[rtop-package](#)

Examples

```
## Not run:
library(rgdal)
# The following command will download the complete example data set
# downloadRtopExampleData()
# observations$obs = observations$QSUMMER_OB/observations$AREASQKM

rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
predictionLocations = readOGR(rpath,"predictionLocations")

# Setting some parameters; nclus > 1 will start a cluster with nclus
```

```

# workers for parallel processing
params = list(gDist = TRUE, cloud = FALSE, nclus = 1, rresol = 25)

# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM

# Build an object
rtopObj = createRtopObject(observations, predictionLocations,
                           params = params)

# Fit a variogram (function also creates it)
rtopObj = rtopFitVariogram(rtopObj)

# Predicting at prediction locations
rtopObj = rtopKrige(rtopObj)

# Cross-validation
rtopObj = rtopKrige(rtopObj, cv=TRUE)
cor(rtopObj$predictions$observed, rtopObj$predictions$var1.pred)

## End(Not run)

```

rtopSim

Spatial simulation of data with spatial support

Description

rtopSim will conditionally or unconditionally simulate data with areal support.

Usage

```

## S3 method for class 'rtop'
rtopSim(object, varMatUpdate = FALSE, beta = NA,
         largeFirst = TRUE, params = list(), ...)
## Default S3 method:
rtopSim(object = NULL, predictionLocations,
         varMatObs, varMatPredObs, varMatPred, variogramModel, ...)

```

Arguments

object	object of class rtop or SpatialPolygonsDataFrame or NULL
varMatUpdate	logical; if TRUE, also existing variance matrices will be recomputed, if FALSE, only missing variance matrices will be computed, see also varMat
beta	The expected mean of the data, for unconditional simulations
largeFirst	Although the simulation method follows a random path around the prediction-Locations, simulating the largest area first will assure that the true mean of the simulated values will be closer to beta

params	a set of parameters, used to modify the standard parameters for the rtop package, set in getRtopParams . The argument params can also be used for the other methods, through the ...-argument.
predictionLocations	SpatialPolygons or SpatialPolygonsDataFrame with locations for simulations.
varMatObs	covariance matrix of possible observations, where diagonal must consist of internal variance, typically generated from call to varMat
varMatPredObs	covariance matrix between possible observation locations and simulation locations, typically generated from call to varMat
varMatPred	covariance matrix between simulation locations, typically generated from a call to varMat
variogramModel	a variogram model of type rtopVariogramModel
...	possible modification of the object parameters or default parameters.

Details

This function can do constrained or unconstrained simulation for areas. The simplest way of calling the function is with an rtop-object that contains the fitted variogram model and all the other necessary data (see [createRtopObject](#) or [rtop-package](#)). rtopSim is the only function in rtop which does not need observations. However, a variogram model is still necessary to perform simulations.

The arguments beta and largeFirst are only used for unconditional simulations.

Value

If called with [SpatialPolygons](#) as predictionLocations and either [SpatialPolygonsDataFrame](#) or NULL for observations, the function returns a [SpatialPolygonsDataFrame](#) with simulations at the locations defined in predictionLocations

If called with an rtop-object, the function returns the same object with the simulations added to the object.

Author(s)

Jon Olav Skoien

References

Skoien J. O., R. Merz, and G. Bloschl. Top-kriging - geostatistics on stream networks. Hydrology and Earth System Sciences, 10:277-287, 2006.

See Also

[rtop-package](#)

Examples

```

## Not run:
library(rgdal)
# The following command will download the complete example data set
# downloadRtopExampleData()

rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
predictionLocations = readOGR(rpath,"predictionLocations")

# Setting some parameters; nclus > 1 will start a cluster with nclus
# workers for parallel processing
params = list(gDist = TRUE, cloud = FALSE, nclus = 1, rresol = 25)

# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM

# Build an object
rtopObj = createRtopObject(observations, predictionLocations,
                          params = params)

# Fit a variogram (function also creates it)
rtopObj = rtopFitVariogram(rtopObj)

# Conditional simulations for two new locations
rtopObj10 = rtopSim(rtopObj, nsim = 5)
rtopObj11 = rtopObj

# Unconditional simulation at the observation locations
# (These are moved to the predictionLocations)
rtopObj11$predictionLocations = rtopObj11$observations
rtopObj11$observations = NULL
# Setting varMatUpdate to TRUE, to make sure that covariance
# matrices are recomputed
rtopObj12 = rtopSim(rtopObj11, nsim = 10, beta = 0.01,
                  varMatUpdate = TRUE)

rtopObj10$simulations@data
rtopObj12$simulations@data

## End(Not run)

```

rtopVariogram

create variogram for data with spatial support

Description

rtopVariogram will create binned variogram or cloud variogram of data with an areal support.

Usage

```
## S3 method for class 'rtop'
rtopVariogram(object, params = list(), ...)
## S3 method for class 'SpatialPolygonsDataFrame'
rtopVariogram(object, ...)
## S3 method for class 'SpatialPointsDataFrame'
rtopVariogram(object, formulaString, params=list(), cloud,
              abins, dbins, ...)
## S3 method for class 'STSDf'
rtopVariogram(object, formulaString, params=list(), cloud,
              abins, dbins, data.table = FALSE, ...)
```

Arguments

object	object of class rtop (see rtop-package) or a SpatialPolygonsDataFrame or SpatialPointsDataFrame with information about observations. If object is a SpatialPointsDataFrame , it must have a column with name area.
formulaString	formula that defines the dependent variable as a linear model of independent variables; suppose the dependent variable has name z, for ordinary and simple kriging use the formula $z \sim 1$; for universal kriging, suppose z is linearly dependent on x and y, use the formula $z \sim x + y$. The formulaString defaults to "value~1" if value is a part of the data set. If not, the first column of the data set is used.
params	a set of parameters, used to modify the standard parameters for the rtop package, set in getRtopParams .
cloud	logical; if TRUE, calculate the semivariogram cloud, can be used to overrule the cloud parameter in params.
abins	possibility to set areal bins (not yet implemented)
dbins	possibility to set distance bins (not yet implemented)
data.table	an option to use data.table internally for the variogram computation for STSDf -objects
...	parameters to other functions called, e.g. gstat's variogram -function and to rtopVariogram.SpatialPointsDataFrame when the method is called with an object of a different class

Value

The function creates a variogram, either of type `rtopVariogram` or `rtopVariogramCloud`. This variogram is based on the [variogram](#) function from `gstat`, but has additional information about the spatial size or length of the observations. An rtop-object with the variogram added is returned if the function is called with an rtop-object as argument.

For spatio-temporal objects ([STSDf](#)), the variogram is the spatially variogram, averaged for all time steps. There is a possibility to use `data.table` internally in this function, which can improve computation time for some cases.

Note

The variogram cloud is similar to the variogram cloud from [gstat](#), with the area/length added to the resulting data.frame. The binned variogram is also based on the area or length, in addition to the distance between observations. The bins are equally distanced in the log10-space of the distances and areas (lengths). The size of the bins is decided from the parameters `amul` and `dmul`, defining the number of bins per order of magnitude (1:10, 10:100, and so on).

The distances between areas are in this function based on the centre of gravity.

Author(s)

Jon Olav Skoien

See Also

[rtop-package](#)

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM

vario = rtopVariogram(observations, cloud = TRUE)

## End(Not run)
```

sceua

Optimisation with the Shuffle Complex Evolution method

Description

Function to minimize the value of an objective function for calibration

Usage

```
sceua(OFUN, pars, lower, upper, maxn = 10000, kstop = 5, pcento = 0.01,
      ngs = 5, npg = 5, nps = 5, nspl = 5, mings = 5, iniflg = 1, iprint = 0, iround = 3,
      peps = 0.0001, plog = rep(FALSE,length(pars)), implicit = NULL, ...)
```

Arguments

OFUN	A function to be minimized, with first argument the vector of parameters over which minimization is to take place. It should return a scalar result as an indicator of the error for a certain parameter set
pars	a vector with the initial guess the parameters
lower	the lower boundary for the parameters
upper	the upper boundary for the parameters
maxn	the maximum number of function evaluations
kstop	number of shuffling loops in which the criterion value must change by the given percentage before optimization is terminated
pcento	percentage by which the criterion value must change in given number (kstop) of shuffling loops to continue optimization
ngs	number of complexes in the initial population
npg	number of points in each complex
nps	number of points in a sub-complex
nspl	number of evolution steps allowed for each complex before complex shuffling
mings	minimum number of complexes required, if the number of complexes is allowed to reduce as the optimization proceeds
iniflg	flag on whether to include the initial point in population = 0, not included = 1, included
iprint	flag for controlling print-out after each shuffling loop = 0, print information on the best point of the population = 1, print information on every point of the population
iround	number of significant digits in print-out
peps	convergence level for parameter set (lower number means smaller difference between parameters of the population required for stop)
plog	whether optimization should be done in log10-domain. Either a single TRUE value for all parameters, or a vector with TRUE/FALSE for the different parameters
implicit	function for implicit boundaries for the parameters (e.g. $\text{sum}(\text{pars}[4]+\text{pars}[5]) < 1$). See below for details
...	arguments for the objective function, must be named

Details

sceua is an R-implementation of the Shuffle Complex Evolution - University of Arizona (Duan et al., 1992), a global optimization method which "combines the strengths of the simplex procedure of Nelder and Mead (1965) with the concepts of controlled random search (Price, 1987), competitive evolution (Holland, 1975)" with the concept of complex shuffling, developed by Duan et al. (1992).

This implementation follows the Fortran implementation relatively close, but adds the possibility of searching in log-space for one or more of the parameters, and it uses the capability of R to pass functions as arguments, making it possible to pass implicit conditions to the parameter selection.

The objective function OFUN is a function which should give an error value for each parameter set. It should never return non-numeric values such as NA, NULL, or Inf. If some parameter combinations can give such values, the return value should rather be a large number.

The function works with fixed upper and lower boundaries for the parameters. If the possible range of a parameter might span several orders of magnitude, it might be better to search in log-space for the optimal parameter, to reduce the risk of being trapped in local optima. This can be set with the argument plog, which is either a single value (FALSE/TRUE) or a vector for all parameters. plog = c(TRUE, FALSE, FALSE, TRUE, TRUE) means that the search for parameters 1,4 and 5 should be in log10-space, whereas the search for parameters 2 and 3 are in normal space.

Implicit boundaries can be evoked by passing a function implicit to sceua. This function should give 0 when parameters are acceptable and 1 if not. If, for example, the condition is that the following sum of parameters four and five should be limited:

$$\text{sum}(\text{pars}[4]+\text{pars}[5]) \leq 1$$

then the function will be implicit = function(pars) (2*pars[4] + pars[5]) > 1

Value

The function returns a list with the following elements

- para vector of the best parameters combination
- valuevalue of the objective function for this parameter set
- convergencelist of two values
 - funConvergencefunction convergence relative to pcento
 - parConvergenceparameter convergence relative to peps
- countsnumber of function evaluations
- iterationsnumber of shuffling loops

Author(s)

Jon Olav Skoien

References

- Duan, Q., Sorooshian, S., and Gupta, V.K., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour. Res.* 28 (4), 1015-1031.
- Holland, H.H., 1975. *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor.
- Nelder, J.A. and Mead, R., 1965. A simplex method for function minimization, *Comput. J.*, 7(4), 308-313.
- Price, W.L., 1987. Global optimization algorithms for a CAD workstation, *J. Optim. Theory Appl.*, 55(1), 133-146.

Examples

```

set.seed(1)
# generate example data from a function with three parameters
# with some random noise
fun = function(x, pars) pars[2]*sin(x*pars[1])+pars[3]
x = rnorm(50, sd = 3)
y = fun(x, pars = c(5, 2, 3)) + rnorm(length(x), sd = 0.3)
plot(x,y)

# Objective function, summing up squared differences
OFUN = function(pars, x, yobs) {
  yvals = fun(x, pars)
  sum((yvals-yobs)^2)
}

sceuares = sceua(OFUN, pars = c(0.1,0.1,0.1), lower = c(-10,0,-10),
                upper = c(10,10,10), x = x, yobs = y)

sceuares
xx = seq(min(x), max(x), 0.1)
lines(xx, fun(xx, pars = sceuares$par))

```

useRtopWithIntamap *Integrates the rtop package with the intamap package*

Description

This function makes it possible to use rtop-objects in the functions of the package. It is necessary to load the intamap-package before calling this function.

Usage

```
useRtopWithIntamap()
```

Value

The function will have as side effect that the intamap package is loaded, and that rtop-methods are registered for the intamap-functions estimateParameters, spatialPredict and methodParameters.

Author(s)

Jon Olav Skoien

Examples

```

library(intamap)
useRtopWithIntamap()

```

variogramModel	<i>create or update variogram model</i>
----------------	---

Description

This gives an easier interface to the parameters of the variogram model

Usage

```
rtopVariogramModel(model = "Ex1", sill = NULL, range = NULL,
  exp = NULL, nugget = NULL, exp0 = NULL,
  observations = NULL, formulaString = obs~1)
## S3 method for class 'rtop'
updateRtopVariogram(object, ...)
## S3 method for class 'rtopVariogramModel'
updateRtopVariogram(object, action = "mult", ...,
  checkVario = FALSE,
  sampleVariogram = NULL, observations = NULL)
```

Arguments

model	variogram model, currently "Ex1" is the only implemented, see Skoien et al (2006)
sill	sill of variogram
range	range of variogram
exp	the exponent of the fractal part of the variogram, see Skoien et al (2006)
exp0	gives the angle of the first part of the variogram in a log-log plot (weibull type), should be between 0 and 2. See Skoien et al (2006)
nugget	nugget of point variogram
formulaString	formula that defines the dependent variable as a linear model of independent variables, see e.g. createRtopObject for more details.
object	either: object of class rtop (see rtop-package), or an rtopVariogramModel.
action	character variable defining whether the new parameters should be add(-ed), mult(-iplied) or replace the former parameters. Leaving the parameters equal to NULL will cause no change.
checkVario	logical, will issue a call to checkVario if TRUE
sampleVariogram	a sample variogram of the data
observations	a set of observations
...	parameters to lower level functions

Value

The function helps creating and updating the parameters of the variogram, by using common names and simple update methods. This is mainly for manual fitting of the variogram. The automatic call to checkVario makes it easier to visualize the effect of the changes to the variogram

Author(s)

Jon Olav Skoien

See Also

[rtop-package](#)

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM
predictionLocations = readOGR(rpath,"predictionLocations")
rtopObj = createRtopObject(observations,predictionLocations)
# Fit a variogram (function also creates it)
rtopObj = rtopFitVariogram(rtopObj)
rtopObj = updateRtopVariogram(rtopObj, exp = 1.5, action = "mult",
                             checkVario = TRUE)

## End(Not run)
```

varMat	<i>create a semivariogram matrix between a set of locations, or semivariogram matrices between and within two sets of locations</i>
--------	---

Description

varMat will create a semivariogram matrix between all the supports in a set of locations (observations or prediction locations) or semivariogram matrices between all the supports in one or two sets of locations, and also between them.

Usage

```
## S3 method for class 'rtop'
varMat(object, varMatUpdate = FALSE, params = list(), ...)
## S3 method for class 'SpatialPolygonsDataFrame'
varMat(object, object2 = NULL,...)
## S3 method for class 'SpatialPolygons'
varMat(object, object2 = NULL, variogramModel,
```

```

overlapObs, overlapPredObs, ...)

## S3 method for class 'list'
varMat(object, object2 = NULL, coor1, coor2, maxdist = Inf,
        variogramModel, diag = FALSE, sub1, sub2,
        debug.level = 1, ...)

```

Arguments

object	either: 1) an object of class <code>rtop</code> (see rtop-package) or 2) a SpatialPolygonsDataFrame , or SpatialPolygons , or 3) a matrix with geostatistical distances (see gDist or 4) a list with discretized supports
varMatUpdate	logical; if TRUE, also existing variance matrices will be recomputed, if FALSE, only missing variance matrices will be computed
params	a set of parameters, used to modify the standard parameters for the <code>rtop</code> package, set in getRtopParams .
object2	if object is not an object of class <code>rtop</code> ; an object of the same class as object with a possible second set of locations with support
variogramModel	variogramModel to be used in calculation of the semivariogram matrix (matrices)
...	typical parameters to modify from the default parameters of the <code>rtop</code> -package (or modifications of the previously set parameters for the <code>rtop</code> -object), see also getRtopParams . These can also be passed in a list named <code>params</code> , as for the <code>rtop</code> -method. Typical parameters to modify for this function: <ul style="list-style-type: none"> • <code>rresol = 100</code> minimum number of discretization points, in call to rtopDisc if necessary • <code>rstype = "rtop"</code> sampling type from areas, in call to rtopDisc if necessary • <code>gDistPred = FALSE</code> use geostatistical distance for semivariogram matrices • <code>gDist</code> parameter to set jointly <code>gDistEst = gDistPred = gDist</code>
overlapObs	matrix with observations that overlap each other
overlapPredObs	matrix with observations and predictionLocations that overlap each other
coor1	coordinates of centroids of object
coor2	coordinates of centre-of-gravity of object2
maxdist	maximum distance between areas for inclusion in semivariogram matrix
diag	logical; if TRUE only the semivariogram values along the diagonal will be calculated, typical for semivariogram matrix of prediction locations
sub1	semivariogram array for subtraction of inner variances of areas
sub2	semivariogram array for subtraction of inner variances of areas
debug.level	<code>debug.level >= 1</code> will give output for every element

Value

The lower level versions of the function calculates a semivariogram matrix between locations in object or between the locations in object and the locations in object2. The method for object of type rtop calculates semivariogram matrices between observation locations, between prediction locations, and between observation locations and prediction locations, and adds these to object.

Note

The argument varMatUpdate is typically used to avoid repeated computations of the same variance matrices. Default is FALSE, which will avoid recomputation of the variance matrix for the observations if the procedure is cross-validation before interpolation. Should be set to TRUE if the variogram Model has been changed, or if observation and/or prediction locations have been changed.

If an rtop-object contains observations and/or predictionLocations of type STSDF, the covariance matrix is computed based on the spatial properties of the object.

Author(s)

Jon Olav Skoien

See Also

[gDist](#), [rtop-package](#)

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
gDist = gDist(observations)
vmod = list(model = "Ex1", params = c(0.00001,0.007,350000,0.9,1000))
vm = varMat(gDist$gDistObs, variogramModel = vmod)

## End(Not run)
```

Index

- *Topic **dplot**
 - plot.rtopVariogramCloud, 15
- *Topic **plot**
 - downloadRtopExampleData, 7
 - netProp, 13
 - useRtopWithIntamap, 33
- *Topic **spatial**
 - checkVario, 4
 - createRtopObject, 6
 - gDist, 8
 - getRtopParams, 10
 - readAreaInfo, 16
 - readAreas, 17
 - rtop-package, 2
 - rtopCluster, 18
 - rtopDisc, 19
 - rtopFitVariogram, 21
 - rtopKrige, 23
 - rtopSim, 26
 - rtopVariogram, 28
 - variogramModel, 34
 - varMat, 35
- axTicks, 11
- checkVario, 3, 4, 34
- clusterEvalQ, 18
- createRtopObject, 2, 6, 10, 12, 13, 24, 27, 34
- data.frame, 22
- downloadRtopExampleData, 7
- gDist, 8, 21, 36, 37
- getRtopParams, 2, 4–7, 9, 10, 19–22, 24, 27, 29, 36
- gstat, 11, 30
- legend, 5
- list, 36
- makeCluster, 18
- matrix, 36
- netProp, 13
- parallel, 24
- plot.gstatVariogram, 16
- plot.rtopVariogramCloud, 15
- plot.variogramCloud, 15
- readAreaInfo, 3, 6, 16, 17, 18
- readAreas, 3, 6, 17
- rtop-package, 2, 12, 29, 34, 36
- rtopCluster, 18
- rtopDisc, 8, 12, 19, 21, 36
- rtopFitVariogram, 3, 21
- rtopKrige, 3, 12, 23
- rtopSim, 26
- rtopVariogram, 3, 20, 28
- rtopVariogram.SpatialPointsDataFrame, 29
- rtopVariogramModel, 27
- rtopVariogramModel (variogramModel), 34
- sceua, 30
- SpatialLinesDataFrame, 14
- SpatialPointDataFrame, 17
- SpatialPointsDataFrame, 6, 17, 18, 21, 29
- SpatialPolygons, 6, 8, 9, 20, 23, 27, 36
- SpatialPolygonsDataFrame, 2, 6, 8–10, 17, 18, 20, 21, 23, 25–27, 29, 36
- STSDF, 24, 29, 37
- updateRtopVariogram (variogramModel), 34
- useRtopWithIntamap, 33
- variogram, 29
- variogramModel, 34
- varMat, 3, 9, 12, 23, 24, 26, 27, 35
- vgm, 5