

Package ‘rtop’

July 2, 2014

Type Package

Title Interpolation of data with variable spatial support

Version 0.3-45

Date 2014-02-06

Author Jon Olav Skoien

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

Depends sp

Imports gstat

Suggests igrph, rgdal, intamap, rgeos

Description Package for 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 2014-02-10 12:50:38

R topics documented:

rtop-package	2
checkVario	3
createRtopObject	5
downloadRtopExampleData	7
gDist	8
getRtopParams	9
netProp	12
plot.rtopVariogramCloud	14
readAreaInfo	15

readAreas	16
rtopDisc	17
rtopFitVariogram	18
rtopKrige	20
rtopVariogram	23
useRtopWithIntamap	24
variogramModel	25
varMat	26

Index	29
--------------	-----------

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

## 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 rtop (see rtop-package), or an object of type rtopVariogram
acor	unit correction factor in the key, e.g. to see numbers more easily interpretable for large areas. As an example, ucor = 0.000001 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 amul of the standard parameters, see getRtopParams . amul from rtopObj 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 amul of the standard parameters, see getRtopParams . amul from rtopObj 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")
observations = readOGR(rpath,"observations")
# Create a column with the specific runoff:
observations$obs = observations$QSUMMER_OB/observations$AREASQKM
predictionLocations = readOGR(rpath,"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

Value

An object of class rtop with observations, prediction locations, parameters and possible other elements useful for interpolation in the rtop-package. Most other externally visible functions in the package will be able to work with this object, and add the results as a new element.

Author(s)

Jon Olav Skoien

See Also

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

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)

## End(Not run)
```

downloadRtopExampleData

Download additional example data

Description

Download additional example data from Vienna University of Technology

Usage

```
downloadRtopExampleData(folder = system.file("extdata",package="rtop"))
```

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, ...)
## 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 <code>rtop</code> with such boundaries and/or discretized elements (the individual areas)
object2	an object of same type as object, 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 varMat , 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 | <code>SpatialPolygonsDataFrame</code> 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. <code>createRtopObject</code> 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: <ul style="list-style-type: none"> – <code>Exp</code> - Exponential model – <code>Ex1</code> - Multiplication of a modified exponential and fractal model, the same model as used in Skoien et al(2006). – <code>Gau</code> - Gaussian model – <code>Ga1</code> - Multiplication of gaussian and fractal model – <code>Sph</code> - Spherical model – <code>Sp1</code> - Multiplication of spherical and fractal model – <code>Fra</code> - Fractal model • <code>parInit</code> - 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. • <code>nugget = TRUE</code> - logical; if TRUE nugget effect should be estimated • <code>unc = TRUE</code> - logical; if TRUE the standard deviation of observations are in column <code>unc</code> • <code>rresol = 100</code> - minimum number of discretization points in each area • <code>hresol = 5</code> - number of discretization points in one direction for elements in binned variograms • <code>cloud = FALSE</code> - logical; if TRUE use the cloud variogram for variogram fitting • <code>amul = 1</code> - defines the number of areal bins within one order of magnitude. Numbers between 1 and 3 are possible, as this parameter refers to the <code>axp</code> parameter of <code>axTicks</code>. • <code>dmul = 3</code> - defines the number of distance bins within one order of magnitude. Numbers between 1 and 3 are possible, as this parameter refers to the <code>axp</code> parameter of <code>axTicks</code>. • <code>fit.method = 8</code> - defines the type of Least Square method for fitting of variogram. The methods 1-7 correspond to the similar methods in <code>gstat</code>. |

- 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](#)
- 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.
- 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.

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

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 <code>iprint >= 1</code> the function will give some information about the convergence of the value propagation. Use <code>iprint = 0</code> 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, ccloud = 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, ccloud = 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
splot(rnet, "pred", col.regions = bpy.colors())
rnet = netProp(rnet)
# will show a prediction for all segments
splot(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

<code>x</code>	object of class <code>variogramCloud</code>
<code>...</code>	parameters that are passed through to <code>plot.variogramCloud</code> 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 <code>x</code>, 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/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 <code>SpatialPointsDataFrame</code> 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 <code>readAreaInfo</code>

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 `maptools`-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

rtopDisc	<i>Discretize areas</i>
----------	-------------------------

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 rtop 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 rtop package, set in getRtopParams . Typical parameters to modify for this function are: <ul style="list-style-type: none"> • rresol = 100; minimum number of discretization points in areas • hresol = 5; number of discretization points in one direction for areas in binned variograms • hstype = "regular"; sampling type for binned variograms • rstype = "rtop"; 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	<i>Fit variogram model to sample variogram of data with spatial support</i>
------------------	---

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, ...)
## 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 <code>rtopVariogram</code> or <code>rtopVariogramCloud</code> , or an object with class <code>rtop</code> 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 <code>area</code> .
observations	the observations, passed as a <code>Spatial*DataFrame</code> object, if object is an <code>rtopVariogram</code> or <code>rtopVariogramCloud</code>
params	a set of parameters, used to modify the standard parameters for the <code>rtop</code> package, set in getRtopParams .
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 <code>rtop</code> -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 <code>NULL</code> and necessary
...	Other parameters to functions called from <code>rtopFitVariogram</code>

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

Spatial interpolation of data with spatial support

Description

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

Usage

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

```
rtopKrige(object, predictionLocations = NULL,
          varMatObs, varMatPredObs, varMat, params = list(), formulaString,
          sel, wret = FALSE, ...)
```

Arguments

object	object of class <code>rtop</code> 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 <code>rtop</code> 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
...	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. if

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 predictions 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.

See Also

[rtop-package](#)

Examples

```
## Not run:
library(rgdal)
rpath = system.file("extdata",package="rtop")
observations = readOGR(rpath,"observations")
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$AREASQKM
# Build an object
rtopObj = createRtopObject(observations,predictionLocations, params = params)
# Fit a variogram (function also creates it)
rtopObj = rtopFitVariogram(rtopObj)
# Predicting at prediction locations
rtopObj = rtopKrige(rtopObj)
# Cross-validation
rtopObj = rtopKrige(rtopObj,cv=TRUE)
cor(rtopObj$predictions$observed,rtopObj$predictions$var1.pred)

## End(Not run)
```

rtopVariogram	<i>create variogram for data with spatial support</i>
---------------	---

Description

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

Usage

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

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)
...	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.

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 equally distanced in the log10-space of the distances and areas (lengths). The size of the bins is decided from the parameters `amu1` and `dmu1`, 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)
```

useRtopWithIntamap *Integrates the rtop package with the intamap package*

Description

This function checks if the [intamap-package](#) is installed, and if it is, makes it possible to use `rtop`-objects in the functions of the package.

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

```
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, ...)
## 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 'matrix'
varMat(object, variogramModel, ...)
## 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
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 . Typical parameters to modify for this function: <ul style="list-style-type: none"> • <code>rresol</code> = 100 minimum number of discretization points, in call to rtopDisc if necessary • <code>rstype</code> = "rtop" sampling type from areas, in call to rtopDisc if necessary • <code>gDistPred</code> = FALSE use geostatistical distance for semivariogram matrices • <code>gDistparameter</code> to set jointly <code>gDistEst</code> = <code>gDistPred</code> = <code>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.

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