Package 'IceCast'

August 16, 2017

Type Package

Title Apply Statistical Post-Processing to Improve Sea Ice Predictions

Version 1.2.0

Author Hannah M. Director, Adrian E. Raftery, and Cecilia M. Bitz

Maintainer Hannah M. Director <direch@uw.edu>

Description Tools for modeling and correcting biases in sea ice predictions obtained from dynamical models. This package depends on the 'ncdf4' and 'rgeos' R packages. These packages respectively require installing externally from R Unidata's 'NetCDF' library and Geometry Engine - Open Source ('GEOS'). (See the 'rgeos' and 'ncdf4' packages for details.) References, Bivand and Rundel (2017) <https://CRAN.R-project.org/package=rgeos>, Open Source Geospatial Foundation (2017) <https://trac.osgeo.org/geos>, Pierce (2017) <https://CRAN.R-project.org/package=ncdf4>, Unidata (2017) <https://www.unidata.ucar.edu/software/netcdf/>.

License GPL (>= 2)

Depends R (>= 2.10), rgeos, sp

Imports ncdf4, maptools, MASS, raster, methods, utils

Suggests geosphere, fields, knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-16 06:54:06 UTC

R topics documented:

`allRegions` *Polygon of all regions*

Description

All regions except the non-regional ocean converted into a single `SpatialPolygons` object. Regions are modified from the region masks provided by the National Snow and Ice Data Center (NSIDC)

Usage

```
allRegions
```

Format

SpatialPolygons object

References

Region Mask: National Snow and Ice Data Center, 2017: Region mask for the northern hemisphere. http://nsidc.org/data/polar-stereo/tools_masks.html.

Examples

```
data(allRegions)
plot(allRegions)
```

---

anyIntersect *Check if a line has intersecting segments*

Description

Determine if there are any intersecting line segments in a matrix of coordinates representing a line

Usage

```
anyIntersect(line)
```

Arguments

line	matrix of coordinates corresponding to the line of interest

Value

list where list$any is a boolean indicating if there are any intersections and list$val is an index corresponding to the first intersection found

Examples

```
checkResults <- anyIntersect(currSecEx)
checkResults$any #true/false
checkResults$val #indices of first intersection found
```

bgWater *Polygon of the non-regional ocean*

Description

The non-regional ocean converted into a single SpatialPolygons object. Thee boundaries of the non-regional ocean were defined by modifying the region masks provided by the National Snow and Ice Data Center (NSIDC).

Usage

```
bgWater
```

Format

SpatialPolygons object

References

Region Mask: National Snow and Ice Data Center, 2017: Region mask for the northern hemisphere. http://nsidc.org/data/polar-stereo/tools_masks.html.

Examples

```
data(bgWater)
plot(bgWater)
```

checkIntersect *Check if line segments intersect*

Description

Find if two line segments intersect

Usage

```
checkIntersect(a, b, c, d, seq = FALSE)
```

Arguments

a	first coordinate of first line segment
b	second coordinate of first line segment
c	first coordinate of second line segment
d	second coordinate of second line segment
seq	indicator for whether the two line segments are intersecting

Value

boolean indicating if there is an intersection

Examples

```
checkIntersect(c(0, 0), c(1, 1), c(2, 2), c(3, 3))
checkIntersect(c(0, 0), c(1, 1), c(0.5, 0.5), c(2, 2))
```

---

contourShift *Apply contour-shifting to bias correct*

Description

Apply contour-shifting to bias correct a predicted contour using existing mappings.

Usage

```
contourShift(maps, predicted, bcYear, predStartYear, regions, level,
  datTypePred, myLandMat = landMat, myAllRegions = allRegions,
  myLand = land)
```

Arguments

maps	object obtained from the createMappings function (see details)
predicted	array of predicted values of dimension year x month x longitude x latitude
bcYear	year to be bias-corrected
predStartYear	year prediction array starts in
regions	region information list (see details)
level	concentration level for which to build contour
datTypePred	string indicating the format of the prediction: either "gfdl" or "simple" (see details)
myLandMat	binary matrix specifying land locations
myAllRegions	a single SpatialPolygons object given the region under consideration
myLand	SpatialPolygons corresponding to the land

Details

The object maps is obtained from running the createMapping function. It is a list of five objects where month, startYear, and endYear give the month, first year, and last year that were mapped. The variables obsList and predList are lists of arrays with one 3-dimensional array for each region. The first dimension is for the year. The other two dimensions are for the fixed points' y-coordinates, the mapped points' x-coordinates, the mapped points' y-coordinates, the length of the mapping vectors in the x-direction, the length of the vectors in the y-direction, and the angles of the mapping vectors.

A region information list is a list six objects: `regions`, `lines`, `out`, `centRegion`, `centLines`, and `centFrom`. The first three objects are ordered lists giving information about each of the regions that will be mapped outside the Central Arctic region. The variable `regions` gives `SpatialPolygons` objects for the corresponding regions and the variable `lines` gives the `SpatialLines` objects for the corresponding fixed lines. The variable `out` gives `SpatialPolygons` objects that are outside the corresponding regions, but that border the fixed lines. These are used when building new polygons to determine if points are being mapped outside the region of interest. The variable `centRegions` is the `SpatialPolygons` object corresponding to the central Arctic region, `centLines` is the `SpatialLines` object for the fixed line, and `centFrom` is an n x 2 matrix with each row repeatedly giving the coordinates of the center point from which mapping vectors will emanate. The package contains `regionInfo` in the `regionInfo.rda` file, which can be used unless you want to define your own regions.

The predicted data array, `predicted`, should be single array of dimension: years x longitude (304) x latitude (448). If `datTypePred = "simple"`, the values in the array should indicate whether each grid box is categorized to contain ice (1: ice-covered, 0: no ice, NA: land). If `datTypePred = "gfdl"`. #' If `datTypePred = "gfdl"` , the values in the `predicted` array correspond to the raw ice concentrations values predicted (including indicators for missing data, land etc.) formatted as in the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory converted to a Polar Stereographic grid (Vecchi et al. 2014; Msadek et al. 2014). Weights for converting to a polar stereograhic grid were obtained from the spherical coordinate remapping and interpolation package (SCRIP) (Jones 1997).

Value

`SpatialPolygons` object of the adjusted region

References

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model:

Jones, P.W. "A user's guide for SCRIP: A spherical coordinate remapping and interpolation package." Los Alamos National Laboratory, Los Alamos, NM (1997).

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215.

National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Examples

```
## Not run:
adj <- contourShift(maps = discrep, predicted = emFeb2012, bcYear = 2012, predStartYear = 1980,
                    regions = regionInfo, level = 15, datTypePred = "gfdl")
plot(land, col = "grey", border = FALSE)
plot(adj, add = TRUE, col = "blue")

## End(Not run)
```

convToGrid *Convert* SpatialPolygons *object to a grid*

Description

Convert SpatialPolygons object to binary grid. Using the rasterize function, grid boxes whose centers are part of the SpatialPolygons are given value 1 and all other grid boxes are given value 0. Land values are set to NA.

Usage

```
convToGrid(x, myLandMat = landMat)
```

Arguments

x	SpatialPolygon object
myLandMat	binary matrix specifying land locations

Examples

```
grid <- convToGrid(bgWater)
image(grid)
```

createMapping *Map a set of observations and predictions*

Description

Finds all the mappings for a set of observations and predictions often over multiple years

Usage

```
createMapping(startYear, endYear, obsStartYear, predStartYear, observed,
  predicted, regions, month, level, datTypeObs, datTypePred, plotting = FALSE,
  obsOnly = FALSE, predOnly = FALSE, nX = 304, nY = 448, xmn = -3850,
  xmx = 3750, ymn = -5350, ymx = 5850)
```

Arguments

startYear	first year to be mapped
endYear	last year to be mapped
obsStartYear	year in which observation array starts
predStartYear	year in which prediction array starts
observed	array of observed values of dimension year x longitude x latitude

predicted	array of predicted values of dimension year x longitude x latitude
regions	region information list (see detail)
month	month under consideration
level	concentration level for which to build contour
datTypeObs	string of either "bootstrap" or "simple" indicating the file type of the observation (see details)
datTypePred	string of either "gfdl" or "simple" indicating the file type of the prediction (see details)
plotting	boolean indicatng whether maps should be plotted (defaults to false)
obsOnly	run mapping only for observations
predOnly	run mapping only for predictions
nX	dimension in the x (defaults to value for Northern Polar stereographic grid: 304)
nY	dimension in the y (defaults to value for Northern Polar stereographic grid: 448)
xmn	min x value (defaults to value for Northern Polar stereographic grid: -3850)
xmx	max x value (defaults to value for Northern Polar stereographic grid: 3750)
ymn	min y value (defaults to value for Northern Polar stereographic grid: -5350)
ymx	max y value (defaults to value for Northern Polar stereographic grid: 5850)

Details

The object maps is obtained from running the createMapping function. It is a list of five objects where startYear and endYear give first year, and last year that were mapped. The variables obsList and predList are lists of arrays with one 3-dimensional array for each region. The first dimension is for the year. The other two dimensions are for the fixed points' y-coordinates, the mapped points' x-coordinates, the mapped points' y-coordinates, the length of the mapping vectors in the x-direction, the length of the vectors in the y-direction, and the angles of the mapping vectors.

A region information list is a list six objects: regions, lines, out, centRegion, centLines, and centFrom. The first three objects are ordered lists giving information about each of the regions that will be mapped outside the Central Arctic region. The variable regions gives SpatialPolygons objects for the corresponding regions and the variable lines gives the SpatialLines objects for the corresponding fixed lines. The variable out gives SpatialPolygons objects that are outside the corresponding regions, but that border the fixed lines. These are used when building new polygons to determine if points are being mapped outside the region of interest. The variable centRegions is the SpatialPolygons object corresponding to the central Arctic region, centLines is the SpatialLines object for the fixed line, and centFrom is an n x 2 matrix with each row repeatedly giving the coordinates of the center point from which mapping vectors will emanate. The package contains regionInfo in the regionInfo.rda file, which can be used unless you want to define your own regions.

For datTypeObs = "simple" and datTypePred = "simple" the values in the observed and predicted arrays are indicators of whether the grid box contains ice (1: ice-covered, 0: no ice, NA: land). If datTypePred = "gfdl" or datTypeObs = "bootstrap", the values in the observed and predicted arrays correspond to the raw ice concentrations values observed or predicted (including indicators for missing data, land etc.). If datTypePred = "gfdl", the predictions are formatted as in the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced

by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory and converted to a Polar Stereographic grid (Vecchi et al. 2014; Msadek et al. 2014). If `datTypeObs = "bootstrap"` the array values are assumed to be from the monthly sea ice concentration obtained from the National Aeronautics and Space Administration (NASA) satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and processed by the bootstrap algorithm. Weights for converting to a polar stereograhic grid were obtained from the spherical coordinate remapping and interpolation package (SCRIP) (Jones 1997).

Value

`map` object (see details)

References

Bootstrap sea ice concentration: Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model: Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215.

National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Jones, P.W. "A user's guide for SCRIP: A spherical coordinate remapping and interpolation package." Los Alamos National Laboratory, Los Alamos, NM (1997).

Examples

```
## Not run:
createMapping(startYear = 1981, endYear = 1981, obsStartYear = 1981, predStartYear = 1980,
             observed = obsFeb19811982, predicted = emFeb19811982,
             regions = regionInfo, month = 2, level = 15,
             datTypeObs = "bootstrap", datTypePred = "gfdl", plotting = TRUE)

## End(Not run)
```

---

currSecEx *Coordinates of a line segment with self-intersections*

Description

Example of a line segment with self-intersections. We will use it to demonstrate the `untwistSec` function.

Usage

```
currSecEx
```

Format

n x 2 matrix of coordiantes

Examples

```
data(currSecEx)
head(currSecEx)
```

---

discrep *Discrepancy maps for February 1981-2011*

Description

The object `discrep` is obtained from running the `createMapping` function for the month of February from 1981-2011. The predictions are from the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory converted to a Polar Stereographic grid at a 3.5-month lead time (Vecchi et al. 2014; Msadek et al. 2014). Weights for converting to a polar stereograhic grid were obtained from the spherical coordinate remapping and interpolation package (SCRIP) (Jones 1997). The observations are from the monthly sea ice concentration obtained from the National Aeronautics and Space Administration (NASA) satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and processed by the bootstrap algorithm. The results are distributed by the National Snow and Ice Data Center (Comiso 2000, updated 2015).

Usage

```
discrep
```

Format

Object obtained from the `createMapping` function (see details)

Details

The object `discrep` is obtained from running the `createMapping` function. It is a list of five objects where `month`, `startYear`, and `endYear` give the month, first year, and last year that were mapped. The variables `obsList` and `predList` are lists of arrays with one 3-dimensional array for each region. The first dimension is for the year. The other two dimensions are for the fixed points' y-coordinates, the mapped points' x-coordinates, the mapped points' y-coordinates, the length of the mapping vectors in the x-direction, the length of the vectors in the y-direction, and the angles of the mapping vectors.

References

Bootstrap sea ice concentration: Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model: Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215.

National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Jones, P.W. "A user's guide for SCRIP: A spherical coordinate remapping and interpolation package." Los Alamos National Laboratory, Los Alamos, NM (1997).

Examples

```
data(discrep)
names(discrep)
```

`emFeb19811982` *Ensemble mean sea ice concentration for February 1981-1982 (initialized in November)*

Description

The object `emFeb19811982` is an array of the ensemble mean of the predicted sea ice concentration from the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory (Vecchi et al. 2014; Msadek et al. 2014). The data have been converted to a Polar stereographic grid. Weights for converting to a polar stereograhic grid were obtained from the spherical coordinate remapping and interpolation package (SCRIP) (Jones 1997).

Usage

```
emFeb19811982
```

Format

array of dimension of 2 x 304 x 448 (corresponding to year x longitude x 448 latitude)

References

Bootstrap sea ice concentration:

Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model:

Jones, P.W. "A user's guide for SCRIP: A spherical coordinate remapping and interpolation package." Los Alamos National Laboratory, Los Alamos, NM (1997).

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215.

National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Examples

```
data(emFeb19811982)
dim(emFeb19811982)
```

`emFeb2012` *Ensemble mean sea ice concentration for February 2012 (initialized in November)*

Description

The object `emFeb2012` is an array of the ensemble mean of the predicted sea ice concentration from the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory (Vecchi et al. 2014; Msadek et al. 2014). The data have been converted to the Polar stereographic grid. Weights for converting to a polar stereograhic grid were obtained from the spherical coordinate remapping and interpolation package (SCRIP) (Jones 1997).

Usage

```
emFeb2012
```

Format

matrix of dimension 304 x 448 (longitude x latitude)

References

Bootstrap sea ice concentration:

Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model:

Jones, P.W. "A user's guide for SCRIP: A spherical coordinate remapping and interpolation package." Los Alamos National Laboratory, Los Alamos, NM (1997).

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215.

National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Examples

```
data(emFeb2012)
dim(emFeb2012)
```

---

extractCoords *Function to extract coordinates.*

Description

Function to extract coordinates from a SpatialLines object. If there are breaks in the line, this function connects the closest points to create one line. Note: This differs from the function getCoords in that the ordering of the points is considered.

Usage

```
extractCoords(x)
```

Arguments

x	SpatialLines or SpatialPolygons object

Value

n x 2 matrix of coordinates

Examples

```
coords <- extractCoords(regionInfo$regions[[3]])
par(mfrow = c(1, 2))
plot(regionInfo$regions[[3]], main = "Polygon Object")
plot(coords, type = "p", main = "Coordinates", pch = 20)
```

`findHoles` *Find holes in a polygon*

Description

Function to find and remove holes in a `SpatialPolygons` object. Note that this function differs from the function `rmHoles` in that it is considering gaps between a polygon and region boundaries in addition to holes contained within the polygon itself.

Usage

```
findHoles(myPoly, myReg, myOut)
```

Arguments

`myPoly`	`SpatialPolygons` object of interest
`myReg`	`SpatialPolygons` object for the region which myPoly is in
`myOut`	`SpatialPolygons` object that is outside the region, but that borders the regions' fixed line. This polygon is used to determine if points are being mapped outside the region of interest.

Value

list where `new$holes` gives the holes in the polygons as a `SpatialPolygons` object and `new$newPoly` gives a `SpatialPolygons` object with the holes removed.

Examples

```
holeEx <- findHoles(regionInfo$regions[[1]], regionInfo$regions[[1]], regionInfo$out[[1]])
plot(holeEx$newPoly)
plot(holeEx$holes, col = "red", add = TRUE)
```

getArea *Calculate geodesic area*

Description

Caclulate the geodesic areas of SpatialPolygons object on the Nothern Hemisphere Polar Stereographic projection

Usage

```
getArea(myPoly, byid = FALSE)
```

Arguments

myPoly	SpatialPolygons object to calculate area of
byid	boolean indicating whether areas should be calculated for each polygon or for the whole object together

Details

Area calculations are for the Polar stereographic grid with major axis of 6378273m and ellipsoid flattening of 1/298.2794111.

Value

Area (or vector of areas if byid is set to TRUE)

References

Information on Polar Stereographic North projection: https://nsidc.org/data/atlas/epsg_3411.html

Examples

```
getArea(regionInfo$regions[[1]])
```

getCoords *Extract coordinates from a spatial object of lines and points*

Description

Get coordinates from a spatial object of lines and points. There is no ordering of points returned. Note: This differs from `extractCoords` in that the ordering of the points is NOT considered.

Usage

```
getCoords(myPoints)
```

Arguments

`myPoints`	spatial object of type `SpatialCollections`, `SpatialPoints`, or `SpatialLines`

Value

n x 2 matrix of coordinates

Examples

```
#Load sample line
exampleLine <- as(rmHoles(bgWater[2]), "SpatialLines")
getCoords(exampleLine)
```

getDist *Find euclidean distance*

Description

Finds the euclidean distance between two points (ignoring projection)

Usage

```
getDist(p1, p2)
```

Arguments

`p1`	first point, x and y coordinate pair
`p2`	second point, x and y coordinate pair

Value

distance value

Examples

```
getDist(c(1, 2), c(3, 4))
```

`getInd` *Find indices in matrix*

Description

Function to find to which matrix indices coordinates correspond (on a 304 x 448 grid)

Usage

```
getInd(coords, xmn = -3850, ymn = -5350)
```

Arguments

`coords`	coordinates of interest
`xmn`	min x (defaults to value for Northern Polar stereographic grid: -3850)
`ymn`	min y (defaults to value for Northern Polar stereographic grid: -5350)

Value

n x 2 matrix of coordinates on a 304 x 448 grid

Examples

```
dat <- matrix(nrow = 2, ncol = 2, data = c(-2000, 0, 300, 1000))
getInd(dat)
```

`getInitMonth` *Get initialization month*

Description

Determine initialization month based on month being forecast and lag. Considers lags up to 11 months in advance.

Usage

```
getInitMonth(month, lag)
```

Arguments

`month`	month being forecast (integer from 1 to 12)
`lag`	months in advance prediction is being made (integer from 1 to 11).

Details

Note that this calculation assumes that the prediction for a month is its first day. This differs from the convention used in our paper which rounds up to the nearest full month. In practice, this may not be the case.

Value

integer corresponding to the initialization month

Examples

```
initMonth <- getInitMonth(month = 10, lag = 4)
initMonth
```

---

getMap *Map one observation or prediction*

Description

Find the mapping vectors for one observation or prediction.

Usage

```
getMap(ice, plotting = FALSE, reg = regionInfo$regions,
  lines = regionInfo$lines, out = regionInfo$out, main = "",
  myLand = land, nSpace = 7)
```

Arguments

ice	SpatialPolygon object corresponding to the region of ice
plotting	boolean indicating if map should be plotted
reg	region information list (defaults to regionInfo$regions)
lines	lines (defaults to regionInfo$lines)
out	back stop (defaults to regionInfo$out)
main	character vector giving the title for the plotting map (defaults to no title)
myLand	SpatialPolygon corresponding to the land
nSpace	Spacing between points and lines that should be plotted (defaults to every seventh point and arrow)

Details

Often reg, lines, and out are taken from a region information list. A region information list is a list of six objects: regions, lines, out, centRegion, centLines, and centFrom. The first three objects are ordered lists giving information about each of the regions that will be mapped outside the Central Arctic region. The variable regions gives SpatialPolygons objects for the corresponding regions and the variable lines gives the SpatialLines objects for the corresponding fixed lines. The variable out gives SpatialPolygons objects that are outside the corresponding regions, but that border the fixed lines. These are used when building new polygons to determine if points are being mapped outside the region of interest. The variable centRegions is the SpatialPolygons object corresponding to the central Arctic region, centLines is the SpatialLines object for the fixed line, and centFrom is an n x 2 matrix with each row repeatedly giving the coordinates of the center point from which mapping vectors will emanate. The package contains regionInfo in the regionInfo.rda file, which can be used unless you want to define your own regions.

Value

List of the length of the number of regions. Each item in the list is a matrix. Each row of each matrix corresponds to a point in the region's fixed line. The seven columns give the fixed point's x-coordinate, the fixed point's y-coordinate, the mapped point's x-coordinate, the mapped point's y-coordinate, the length of the mapping vectors in the x-direction, the length of the vectors in the y-direction, and the angles of the mapping vectors.

Examples

```
## Not run:
obs <- getRegion(dat = obsFeb19811982[1,,], datType = "bootstrap", level = 15)
obsMap <- getMap(ice = obs, plotting = TRUE,
                 main = "Observed Mapping \n February 1985")

## End(Not run)
```

---

getRegion *Get polygons corresponding to regions*

Description

Takes in a matrix and returns polygons representing contiguous regions in the matrix. Typically these regions are either where the ice concentration is above a certain level or where there is land.

Usage

```
getRegion(dat, datType, level, myLandMat = landMat,
  myAllRegions = allRegions, useAll = FALSE, landInd = FALSE,
  xmn = -3850, xmx = 3750, ymn = -5350, ymx = 5850)
```

Arguments

dat	matrix of one of the allowed data types ("gfdl", "bootstrap", or "simple) (see details)
datType	string indicating the format of the data: either "gfdl", "bootstrap", or "simple" (see details)
level	concentration level of interest
myLandMat	binary matrix specifying land locations
myAllRegions	SpatialPolygons object specifying region that will be considered
useAll	boolean, if true indicates to use the full area (overrides myLandMat)
landInd	boolean, if true indicates that the region of interest is the land
xmn	min x (defaults to value for polar stereographic grid, -3850)
xmx	max x (defaults to value for polar stereographic grid, 3750)
ymn	min y (defaults to value for polar stereographic grid, -5350)
ymx	max y (defaults to value for polar stereographic grix, 5850)

Details

For datType = "simple" the values in the dat matrix are indicators of whether the grid box contains ice (1: ice-covered, 0: no ice, NA: land). If datType = "gfdl" or datType = "bootstrap", the values in the matrix correspond to the raw ice concentrations values observed or predicted (including indicators for missing data, land etc.). If datType = "gfdl", the predictions are formatted as in the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory converted to a Polar Stereographic grid (Vecchi et al. 2014; Msadek et al. 2014). If datType = "bootstrap" the array values are formatted the same as the ice concentration values obtaned from the National Aeronautics and Space Administration (NASA) satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and processed by the bootstrap algorithm.

Value

region of interest as a SpatialPolygons object

References

Bootstrap sea ice concentration: Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model: Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215.

National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Examples

```
## Not run:
obsExample <-  getRegion(dat = obsFeb2012, datType = "bootstrap", level = 15)
plot(land, col = 'grey', border = FALSE)
plot(obsExample, col = "lightblue", add = TRUE)

## End(Not run)
```

---

interEx *Example of a line that contains self-intersections*

Description

Example of a line that contains self-intersections. We will use it to demonstrate the functions that address these intersections.

Usage

```
interEx
```

Format

n x 2 matrix of coordinates

Examples

```
data(interEx)
plot(interEx)
```

---

intLine *Space points along a line*

Description

The function evenly spaces the number of points that are on one line, `predL`, on a different line, `obsL`

Usage

```
intLine(predL, obsL, plotting = FALSE)
```

Arguments

`predL`	predicted line (n1 x 2 matrix of coordinates)
`obsL`	predicted line (n2 x 2 matrix of coordinates)
`plotting`	boolean indicating whether maps should be plotted

Value

n x 2 matrix of evenly-spaced coordinates

Examples

```
lineSpace <- intLine(predLEx, obsLEx, plotting = TRUE)
```

---

keepLine *Keep only spatial lines*

Description

Keep only SpatialLines from a spatial object.

Usage

```
keepLine(myPoly)
```

Arguments

myPoly	SpatialCollections, SpatialPolygons, SpatialPoints, or SpatialLines object

Value

SpatialPolygons object

Examples

```
par(mfrow = c(1, 2))
plot(spatialCollEx, col = "blue", main = "Spatial Collections Object")
lineOnly <- keepLine(spatialCollEx)
plot(lineOnly, col = "blue", main = "Spatial Line Only")
```

---

keepPoly *Keep only spatial polygons*

Description

Keep only SpatialPolygons from a spatial object.

Usage

```
keepPoly(myPoly)
```

Arguments

myPoly	SpatialCollections, SpatialPolygons, SpatialPoints, or SpatialLines object

Value

SpatialPolygons object

Examples

```
par(mfrow = c(1, 2))
plot(spatialCollEx, col = "blue", main = "Spatial Collections Object")
polyOnly <- keepPoly(spatialCollEx)
plot(polyOnly, col = "blue", main = "Spatial Polygon Only")
```

---

land *Polygon of land*

Description

Land mask as a single SpatialPolygons object. The land mask was obtained from the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory converted to a Polar Stereographic grid. (Vecchi et al. 2014; Msadek et al. 2014). Weights for converting to a polar stereograhic grid were obtained from the spherical coordinate remapping and interpolation package (SCRIP) (Jones 1997).

Usage

```
land
```

Format

SpatialPolygons object

References

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model: Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215. National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Jones, P.W. "A user's guide for SCRIP: A spherical coordinate remapping and interpolation package." Los Alamos National Laboratory, Los Alamos, NM (1997).

Region Mask: National Snow and Ice Data Center, 2017: Region mask for the northern hemisphere. http://nsidc.org/data/polar-stereo/tools_masks.html.

Examples

```
data(land)
plot(land)
```

`landMat` *Binary matrix indicating where there is land*

Description

Binary matrix of dimension 304 x 448 with value for 1 for land grid boxes and 0 otherwise. Data are on a north Polar Stereographic grid with the land mask simplified to match model output from the CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model produced by the National Oceanic and Atmospheric Administration's Geophysical Fluid Dynamics Laboratory converted to a Polar Stereographic grid (Vecchi et al. 2014; Msadek et al. 2014). Weights for converting to a polar stereograhic grid were obtained from the spherical coordinate remapping and interpolation package (SCRIP) (Jones 1997).

Usage

```
landMat
```

Format

304 x 448 matix

References

CM2.5 Forecast-oriented Low-Ocean Resolution (FLOR) model:

Jones, P.W. "A user's guide for SCRIP: A spherical coordinate remapping and interpolation package." Los Alamos National Laboratory, Los Alamos, NM (1997).

Msadek, R., et al. "Importance of initial conditions in seasonal predictions of Arctic sea ice extent." Geophysical Research Letters 41.14 (2014): 5208-5215.

National Center for Atmospheric Research, 2017: Earth system grid at NCAR. https://www.earthsystemgrid.org/home.html.

Vecchi, Gabriel A., et al. "On the seasonal forecasting of regional tropical cyclone activity." Journal of Climate 27.21 (2014): 7994-8016.

Examples

```
data(landMat)
image(landMat, xaxt = "n", yaxt = "n")
```

makePolygons *Create polygon from mapped points*

Description

Create a new polygon from the coordinates of mapped points

Usage

```
makePolygons(myEnd, myFixedLine, myOut, myLand = land,
  polyName = "unspecified")
```

Arguments

myEnd	n x 2 list of mapped points, i.e. the points to which the polygon should extend
myFixedLine	SpatialLines object from which the mapping vectors emanate
myOut	SpatialPolygons object outside the current region that borders the region's fixed line. It is used to determine if points are being mapped outside the region of interest.
myLand	SpatialPolygons object for the land
polyName	character string to name the new polygon (defaults to "unspecified")

Value

SpatialPolygons object created from the mapped points

Examples

```
newPoly <- makePolygons(myEnd = mappedPoints, myFixedLine = regionInfo$lines[[1]],
                        myOut = regionInfo$out[[1]])
plot(newPoly)
```

mappedPoints *Example of mapped points*

Description

Example of a set of mapped points organized as an n x 2 matrix of coordinates. This is used to demonstrate the makePolygons function.

Usage

```
mappedPoints
```

Format

matrix of 1027 x 2

Examples

```
data(mappedPoints)
head(mappedPoints)
plot(mappedPoints, type = "l")
```

---

mapxy *Get geodetic latitudes and longitudes*

Description

Get corresponding latitude and longitude values for coordinates on a Polar Stereographic North projection grid

Usage

```
mapxy(X, Y, sgn = 1, slat = 70, re = 6378.273, e2 = 0.006693883,
  degrees = TRUE)
```

Arguments

X	Polar Stereographic X Coordinate (km)
Y	Polar Stereographic Y Coordinate (km)
sgn	indicator for Northern hemisphere (defaults to 1)
slat	standard latitude (defaults to 70)
re	Earth's radius (defaults to 6378.273)
e2	eccentricity squared (defaults to 0.006693883)
degrees	boolean indicating whether result should be returned in degrees or radians

Value

list with elements `coords$aLat`, the geodetic latitude (degrees, +90 to -90), and `coords$aLon`, the geodetic longitude (degrees, -180 to 180)

References

The equations for this calculation are from Snyder, J. P., 1982, Map Projections Used by the U.S. Geological Survey, Geological Survey Bulletin 1532, U.S. Government Printing Office. See JPL Technical Memorandum 3349-85-101 for further details.

Examples

```
new <- mapxy(100, 300)
new$aLat
new$aLon
```

obsFeb19811982 *Observed sea ice February 1981-1982*

Description

The object observed is an array obtained from the function readMonthlyBS for startYear = 1981 and endYear = 1982. It gives the observed sea ice concentrations arranged in an array of dimension of year x month x lon x lat. The observations are from the monthly sea ice concentration obtained from the National Aeronautics and Space Administration (NASA) satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and processed by the bootstrap algorithm. The results are distributed by the National Snow and Ice Data Center (Comiso 2000, updated 2015).

Usage

```
obsFeb19811982
```

Format

array of dimension of 2 x 12 x 304 x 448 (year x month x longitude x latitude)y

References

Bootstrap sea ice concentration:

Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

Examples

```
data(obsFeb19811982)
dim(obsFeb19811982)
```

obsFeb2012 *Observed sea ice February 2012*

Description

The object observed is an array obtained from the readMonthlyBS function for startYear = 2012 and endYear = 2012. It gives the observed sea ice concentrations arranged in an array of dimension of year x month x lon x lat

Usage

```
obsFeb2012
```

Format

array of dimension of 2 years x 12 months x 304 longitudes x 448 latitudes

References

Bootstrap sea ice concentration: Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

Examples

```
data(obsFeb2012)
dim(obsFeb2012)
```

---

obsLEx *Coordinates of an observed line segment*

Description

Example of the coordinates for an observed line segment. We will use it to demonstrate the `intLine` function.

Usage

```
obsLEx
```

Format

n x 2 matrix of coordinates

Examples

```
data(obsLEx)
head(obsLEx)
```

predLEx *Coordinates of a predicted line segment*

Description

Example of the coordinates for a predicted line segment. We will use it to demonstrate the `intLine` function.

Usage

```
predLEx
```

Format

n x 2 matrix of coordinates

Examples

```
data(predLEx)
head(predLEx)
```

quickRun *Simple evaluation of contour-shifting*

Description

Reads in netCDF files of observations and predictions, performs bias correction, and exports a new netCDF file with bias-corrected predictions

Usage

```
quickRun(obsNCDF, predNCDF, predYears, startYear, month, outputFile, level,
  datTypeObs = "bootstrap")
```

Arguments

`obsNCDF`	filepath for observed data array (see details for info about array structure)
`predNCDF`	filepath for predicted data array (see details for info about array structure)
`predYears`	vectors of years for which to make prediction
`startYear`	first year to use when learning model
`month`	month of prediction
`outputFile`	filepath for where bias-corrected netCDF file should be stored
`level`	concentration level for which to build contour
`datTypeObs`	string of either "bootstrap" or "simple" indicating the file type of the observation (see details for info about array structure)

Details

The predicted data array, predNCDF, should be a netCDF file with a single array of dimension: years x longitude (304) x latitude (448). The variable should be named iceInd. The values in the array should indicate whether each grid box is categorized to contain ice (1: ice-covered, 0: no ice, NA: land). The observed data array, obsNCDF, should be a netCDF file with a single array of dimension: years x longitude (304) x latitude (448). The observed data array, obsNCDF, can be formatted the same as predNCDF if datTypeObs = "simple". Alternatively, if datTypeObs = "bootstrap" the array values can be ice concentration values obtained from the National Aeronautics and Space Administration (NASA) satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and processed by the bootstrap algorithm. Data should be retained in the same format as given by bootstrap (including indicators for missing data, land etc.). The variable should be named "conc".

Value

netCDF file of dimension years by longitude (304) by latitude (448) with indicators for where ice is predicted after bias correction. (1: ice-covered, 0: not ice, NA: land). Grid boxes have been categorized as ice if their centers are ice covered (within R the bias-corrected contours are not restricted to align to a grid).

References

Bootstrap sea ice concentration: Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

Examples

```
## Not run:
quickRun(obsNCDF = "/obs.nc", predNCDF = "/pred.nc", predYears = c(2001:2013),
         startYear = 1980, month = 2, outputFile = "/outputFile.nc", level = 15,
         datTypeObs = "simple")

## End(Not run)
```

---

readBootstrap *Read individual bootstrap binary file*

Description

Read in individual binary files of monthly observation data. The observations are from the monthly sea ice concentration obtained from the National Aeronautics and Space Administration (NASA) satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and processed by the bootstrap algorithm. The results are distributed by the National Snow and Ice Data Center (Comiso 2000, updated 2015). Functions assume file name conventions are the same as used by NSIDC.

Usage

```
readBootstrap(fileName, nX = 304, nY = 448)
```

Arguments

fileName	File name for binary bootstrap data
nX	dimension in the x (defaults to value for Northern Polar stereographic grid: 304)
nY	dimension in the y (defaults to value for Northern Polar stereographic grid: 448)

Value

numeric vector of concentrations

References

Bootstrap sea ice concentration:

Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

Examples

```
## Not run:
#fileName should be the binary file
rawData <- readBootstrap(fileName)

## End(Not run)
```

---

readMonthlyBS *Read in a set of bootstrap observations over a set of year*

Description

Function to process monthly bootstrap data over multiple years. The observations are from the monthly sea ice concentration obtained from the National Aeronautics and Space Administration (NASA) satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and processed by the bootstrap algorithm. The results are distributed by the National Snow and Ice Data Center (Comiso 2000, updated 2015). Functions assume file name conventions are the same as used by NSIDC.

Usage

```
readMonthlyBS(startYear, endYear, fileFolder, nX = 304, nY = 448)
```

Arguments

startYear	first year to read in
endYear	lastYear to read in
fileFolder	Folder in which binary files are stored
nX	longitude dimension
nY	latitude dimension

Details

Raw binary files for 2012-2013 are included in the package

Value

Bootstrap observations sorted into array of dimension: year x month x lon x lat

References

Bootstrap sea ice concentration:

Comiso, J., 2000, updated 2015: Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS. version 2. http://nsidc.org/data/nsidc-0079

Examples

```
## Not run:
#myFilePath should be a file path where the 1983 binary files are stored
observedDemo <- readMonthlyBS(startYear = 1983, endYear = 1983, fileFolder = myFilePath)

## End(Not run)
```

---

regionInfo *Information on the regions*

Description

Information on the regions

Usage

```
regionInfo
```

Format

This region information list is a list six objects: `regions`, `lines`, `out`, `centRegion`, `centLines`, and `centFrom`. The first three objects are ordered lists giving information about each of the regions that will be mapped outside the Central Arctic region. The variable `regions` gives `SpatialPolygons` objects for the corresponding regions and the variable `lines` gives the `SpatialLines` objects for the corresponding fixed lines. The variable `out` gives `SpatialPolygons` objects that are outside the corresponding regions, but that border the fixed lines. These are used when building new polygons to determine if points are being mapped outside the region of interest. The variable `centRegions` is the `SpatialPolygons` object corresponding to the central Arctic region, `centLines` is the `SpatialLines` object for the fixed line,

References

The regions in this object have been modified from the following region mask:

National Snow and Ice Data Center, 2017: Region mask for the northern hemisphere http://nsidc.org/data/polar-stereo/tools_masks.html.

Examples

```
data(regionInfo)
names(regionInfo)
```

`rmHoles` *Remove holes in a polygon*

Description

Remove holes from a `SpatialPolygons` object. Note that this function differs from the function `findHoles` in that it only removes holes contained within the polygon itself, not gaps between the polygon and region boundaries

Usage

```
rmHoles(myPoly, polyName = "notSpecified")
```

Arguments

`myPoly`	`SpatialPolygons` object
`polyName`	character string to name polygon (defaults to "notSpecified")

Value

`SpatialPolygons` object with holes removed

Examples

```
withHoles <- bgWater[2]
plot(withHoles, col = "blue", main = "Polygon with Holes")
noHoles <- rmHoles(withHoles)
plot(noHoles, col = "blue", main = "Holes removed")
```

spatialCollEx *Spatial collection example*

Description

Example of a SpatialCollections object that contains a SpatialPolygons object and SpatialLines object

Usage

```
spatialCollEx
```

Format

SpatialCollections object

Examples

```
data(SpatialCollEx)
plot(spatialCollEx)
plot(spatialCollEx@lineobj, col = "red", add = TRUE)
plot(spatialCollEx@polyobj, col = "blue", add = TRUE)
```

untwist *Remove self-intersections*

Description

Function to remove all self-intersections from a contour.

Usage

```
untwist(myPoly, plotting = FALSE, polyName = "unspecified")
```

Arguments

myPoly	SpatialPolygons object from which self-intersections need to be removed
plotting	boolean indicating if results should be plotted
polyName	name for SpatialPolygons object to return (defaults to "unspecified")

Value

SpatialPolygons object with self-intersections removed

Examples

```
## Not run:
par(mfrow = c(1, 2))
plot(interEx, main = "Original Contour")
noInter <- untwist(interEx, polyName = "interEx")
plot(noInter, main = "Final Contour")

## End(Not run)
```

`untwistSec` *Remove self-intersections from one section of a contour*

Description

Function to correct self-intersections in a section of a line.

Usage

```
untwistSec(line, tol = 0, eps = 0.25)
```

Arguments

`line`	N x 2 matrix of coordinates
`tol`	how much of a difference between the original line and the simplified line is allowed
`eps`	how much to increase `tol` by on each iteration

Value

n x 2 matrix of the new coordinates with self-intersections removed

Examples

```
par(mfrow = c(1, 2))
plot(currSecEx, type = "l", main = "Original Line Section", xlab = "", ylab = "")
newSec <- untwistSec(currSecEx)
plot(newSec, type = "l", main = "New Line Section", xlab = "", ylab =  "")
```

Index