Package 'movegroup'

May 9, 2026

Title Visualizing and Quantifying Space Use Data for Groups of Animals

Version 2024.03.05

Description Offers an easy and automated way to scale up individual-level space use analysis to that of groups. Contains a function from the 'move' package to calculate a dynamic Brownian bridge movement model from movement data for individual animals, as well as functions to visualize and quantify space use for individuals aggregated in groups. Originally written with passive acoustic telemetry in mind, this package also provides functionality to account for unbalanced acoustic receiver array designs, and satellite tag data.

Depends R (>= 4.1.0)

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

LazyData true

Imports ggplot2 (>= 3.3.5), utils, beepr (>= 1.3), dplyr (>= 1.0.8), ggmap (>= 3.0.0), knitr (>= 1.45), lubridate (>= 1.8.0), magick (>= 2.8.2), methods, move (>= 4.1.6), purrr (>= 1.0.2), raster (>= 3.5-15), rlang (>= 1.0.2), sf (>= 1.0-7), sp (>= 1.4-6), stars (>= 0.5-5), starsExtra (>= 0.2.7), stringr (>= 1.4.0), terra (>= 1.7-39), tidyr (>= 1.2.0), tidyselect (>= 1.2.0), viridis (>= 0.6.4)

RoxygenNote 7.3.1

Suggests rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Simon Dedman [aut, cre] (ORCID: <https://orcid.org/0000-0002-9108-972X>), Maurits van Zinnicq Bergmann [aut] (ORCID: <https://orcid.org/0000-0002-8414-5025>)

Maintainer Simon Dedman <simondedman@gmail.com>

Repository CRAN

Date/Publication 2024-03-07 10:10:02 UTC

Contents

`alignraster` *Combines region-specific group-level UD rasters into a single raster.*

Description

Extends the spatial extent of each area-specific group-level raster to the spatial extent shared by all rasters. This will only be required if you have multiple individuals (e.g. different sharks) divided amongst a few discrete areas (e.g. around different islands) and the effort (e.g. receiver coverage) is different among islands. Not required for multiple individuals all within the same region or sampling regime.

Usage

```
alignraster(
  folderroots = c("/myfolder/H", "/myfolder/L", "/myfolder/M"),
  foldernames = c("H", "L", "M"),
  pattern = ".asc",
  scalefolder = "Scaled",
  scaledweightedname = "All_Rasters_Scaled_Weighted",
  savefolder = tempdir(),
  format = "ascii",
  datatype = "FLT4S",
  bylayer = TRUE,
  overwrite = TRUE,
  returnObj = FALSE
)
```

Arguments

`folderroots`	Character vector of locations of folder roots output by movegroup. Function expects CRS.Rds file and a subfolder with the scaled raster.
`foldernames`	Character vector names of folders corresponding to files in folderroots, i.e. the names of the objects, arrays, regions, etc.
`pattern`	For input rasters from scaleraster. Default ".asc".
`scalefolder`	For input rasters from scaleraster. Default "Scaled".

`scaledweightedname`
: For input rasters from scaleraster. Default "All_Rasters_Scaled".

`savefolder`
: E.g. "/myfolder/Aligned". Single character entry of folder to save outputs, no trailing slash.

`format`
: Character. Output file type for raster::writeRaster param format. Default ascii, other options have generally not worked well in SD's experience.

`datatype`
: Character. Data type for writing values to disk for raster::writeRaster param datatype. Default FLT4S.

`bylayer`
: For raster::writeRaster param bylayer. Default TRUE.

`overwrite`
: For raster::writeRaster param overwrite. Default TRUE.

`returnObj`
: Logical. Return the scaled object to the parent environment to be assigned as an object? Default FALSE.

Details

When used in a movegroup pipeline, the order would be: movegroup.R, scaleraster.R, alignraster.R if required, plotraster.R.

Value

Region-specific group-level UD rasters that share the same spatial extent.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Maurits van Zinnicq Bergmann, <mauritsvzb@gmail.com>

Examples

```
# load data
data("TracksCleaned")
# loop movegroup and scaleraster through tide subsets
tide <- c("H", "M", "L")
for (i in tide) {
dir.create(file.path(tempdir(), i))
  movegroup(
    data = TracksCleaned[TracksCleaned$T.Ph == i, ],
    ID = "Shark",
    Datetime = "Datetime",
    Lat = "Lat",
    Lon = "Lon",
    savedir = file.path(tempdir(), i, "/"))

  scaleraster(path = file.path(tempdir(), i),
              crsloc = file.path(tempdir(), i))
}

alignraster(folderroots = file.path(tempdir(), tide),
            foldernames = tide,
```

```
savefolder = file.path(tempdir(), "Aligned"))
```

`argosFiltered` *Data: Tracks of two great hammerhead sharks with position confidence intervals*

Description

Tracks of 2 great hammerhead sharks tagged in Jupiter, and The Keys, Florida, USA, in 2022 and 2023 respectively, by Saving The Blue (savingtheblue.org), filtered by argosfilter::sdafilter and with state space model applied using aniMotum package, using scripts by Vital Heim, see https://github.com/SimonDedman/SavingTheBlue/blob/main/R/06A_Filter_SPOT_data.R and https://github.com/SimonDed
.

Usage

```
data(argosFiltered)
```

Format

A data frame with 1492 rows and 8 variables:

id Character, shark ID.

date POSIXct datetime, format YYYY-MM-DD HH:MM:SS.

lon Decimal longitudes.

lon025 Decimal longitudes, lower 95% confidence interval bound.

lon975 Decimal longitudes, upper 95% confidence interval bound.

lat Decimal latitudes.

lat025 Decimal latitudes, lower 95% confidence interval bound.

lat975 Decimal latitudes, upper 95% confidence interval bound.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Vital Heim, <vital.heim@gmail.com>

Source

https://www.savingtheblue.org

movegroup *Automates dynamic Brownian bridge movement model construction across individuals*

Description

Automates dynamic Brownian bridge movement model calculation for utilization distribution (UD) estimation for multiple individuals simultaneously, using functions in the 'move' package. The authors are indebted to the move package authors Bart Kraunstauber, Marco Smolla, and Anne K Scharf, and to Sarah Becker for seed code which inspired the development of the movegroup::movegroup function.

Usage

```
movegroup(
  data = NULL,
  ID = NULL,
  Datetime = NULL,
  Lat = NULL,
  Lon = NULL,
  dat.TZ = "US/Eastern",
  proj = sp::CRS("+proj=longlat +datum=WGS84"),
  sensor = "VR2W",
  moveLocError = 1,
  timeDiffLong = 2,
  timeDiffUnits = "hours",
  center = TRUE,
  centre = NULL,
  buffpct = 0.3,
  rasterExtent = NULL,
  rasterCRS = sp::CRS("+proj=utm +zone=17 +datum=WGS84"),
  rasterResolution = 50,
  movemargin = 11,
  dbbext = 3,
  dbbwindowsize = 23,
  writeRasterFormat = "ascii",
  writeRasterExtension = ".asc",
  writeRasterDatatype = "FLT4S",
  absVolumeAreaSaveName = "VolumeArea_AbsoluteScale.csv",
  savedir = tempdir(),
  alerts = TRUE
)
```

Arguments

data	Data frame object containing the data. Requires columns Lat Lon DateTime ID and potentially a grouping column (not currently implemented, email to request). Column names specified in later parameters.

ID	Name of animal tag ID column in data. "Character".
Datetime	Column name in data that contains date/time stamps for each recorded detection. Must be in POSIXct format. "Character".
Lat	Name of latitude column in data. "Character".
Lon	Name of longitude column in data. "Character".
dat.TZ	Timezone of data for as.POSIXct. Default "US/Eastern".
proj	CRS for move function. Default sp::CRS("+proj=longlat +datum=WGS84").
sensor	Sensor for move function. Single character or vector with length of the number of coordinates. Optional. Default "VR2W".
moveLocError	Location error (m) in the 'brownian.bridge.dyn' function in the 'move' package. Numeric. Either single or a vector of length nrow data. If using passive acoustic data this is the detection range of the receiver(s). Default 1. See MoveLocErrorCalc function for satellite data with state space modelled locations with 95% confidence intervals for latlon i.e. lat and lon025 and 975.
timeDiffLong	Single numeric value. Threshold value in timeDiffUnits designating the length of long breaks in re-locations. Used for bursting a movement track into segments, thereby removing long breaks from the movement track. See ?move::bursted for details. Default 2 hours is arbitrary, looping through 18, 24, and 36 hours for satellite data on great hammerhead sharks revealed volume areas for core and general use gradually rise with timeDiffLong increases, multiple small dots of presence get blobbed together, and therefore sometimes this covers land. Ideally one would not discard any data, in which case one should choose a value higher than the largest between-detections gap in their dataset (or just pick a very large number). This parameter is useful when the model would otherwise get stuck trying to calculate a UD for an individual with a very large home range that is inadequately captured by a receiver array. Default 2.
timeDiffUnits	Character. Unit for timeDiffLong. Default "hours".
center	US English alternate to centre. Do you want to center the move object within extent? See spTransform. Default TRUE.
centre	British English alternate to center. Do you want to centre the move object within extent? See spTransform. Default NULL.
buffpct	Buffer extent for raster creation, proportion of 1. Default 0.3, can try e.g. 3 for a large buffer to avoid clipping, at the cost of file size, but later cropping in plotraster.R will remove extraneous blank space.
rasterExtent	Extent of raster created around data. If NULL (default), calculated from data, buffpct, rasterResolution. Else length 4 vector, c(xmn, xmx, ymn, ymx) decimal latlon degrees. Don't go to 90 (degrees) north or south for ymax or ymin. Doesn't prevent constraint to data limits (in plot anyway), but prevents raster clipping crash.
rasterCRS	CRS for raster creation. Default sp::CRS("+proj=utm +zone=17 +datum=WGS84").
rasterResolution	Single numeric value to set raster resolution - cell size (width and height) in metres. 111000: 1 degree lat = 111km. Trade-off between small res = big file & processing time. Should be a function of the spatial resolution of your receivers

or positioning tags. Higher resolution will lead to more precision in the volume areas calculations. Try using 2*dbblocationerror, if dbblocationerror is a single value. Default 50 = 50m = 50m² = 0.00005 km² (divide by 1000000). Try around the median of your moveLocError.

`movemargin` Margin size for variance calc in move::brownian.motion.variance.dyn and behavioural change point analysis in move::brownian.bridge.dyn. Must be an odd number. Default 11.

`dbbext` Ext param in the 'brownian.bridge.dyn' function in the 'move' package. Extends bounding box around track. Numeric single (all edges), double (x & y), or 4 (xmin xmax ymin ymax) . Default 3. Excessive buffering will get cropped automatically.

`dbbwindowsize` The window.size param in the 'brownian.bridge.dyn' function in the 'move' package. The size of the moving window along the track. Larger windows provide more stable/accurate estimates of the brownian motion variance but are less well able to capture more frequent changes in behaviour. Number must be odd. Code will not run if total detections of individual < window size (default 23), which must be >= 2*movemargin (default 11).

`writeRasterFormat`
Character. Output file type for raster::writeRaster param format. Default "ascii". TO DEPRECIATE.

`writeRasterExtension`
Character. Output file extension for raster::writeRaster param extension. Default ".asc". TO DEPRECIATE.

`writeRasterDatatype`
Character. Data type for writing values to disk for raster::writeRaster param Datatype. Default "FLT4S". TO DEPRECIATE.

`absVolumeAreaSaveName`
File name plus extension where UD estimates are saved. Default "VolumeArea_AbsoluteScale.csv".

`savedir` Save outputs to a temporary directory (default) else change to desired directory e.g. "/home/me/folder". Do not use getwd() for this. Do NOT include terminal slash. Directory must exist. Default tempdir().

`alerts` Audio warning for failures. Default TRUE.

Details

Step 1. Filter individuals. Remove those individuals for which there are insufficient data i.e. number of re-locations is smaller than the window size parameter value (default = 31). Step 2. Generate universal raster. Based on all remaining data, a universal raster is generated where the calculated UDs are plotted into.

Step 3. Loop through individuals. Individuals are looped through to construct individual-level movement models (on an absolute scale).

See www.GitHub.com/SimonDedman/movegroup for issues, feedback, and development suggestions.

install_git('https://gitlab.com/bartk/move.git') #Installs 'move' development version

When used together, the order of functions would be: movegroup, scaleraster, alignraster if required, plotraster.

Errors and their origins::

1. Error in .local(object, raster, location.error = location.error, ext = ext: Higher y grid not large enough, consider extending the raster in that direction or enlarging the ext argument. Increase buffpct, e.g. to 3.
2. Error in .data[[dttm]]: Must subset the data pronoun with a string, not a <POSIXct/POSIXt> object. Use "ColName" not dataframe$ColName syntax for Datetime, ID, Lat, Lon.
3. Error in splice(dot_call(capture_dots, frame_env = frame_env, named = named,: object 'DateTime' not found. Use "ColName" not ColName syntax for Datetime, ID, Lat, Lon.
4. Error in .local(object, raster, location.error = location.error, ext = ext: Higher x grid not large enough, consider extending the raster in that direction or enlarging the ext argument. Try "buffpct = 1," , then larger e.g. 3, if still getting the error.
5. cannot allocate vector of size (BIG) Gb: Increase rasterResolution value.
6. In min/max: No non-missing arguments to min; returning Inf: likely not enough memory, increase rasterResolution value.
7. Error in tmp[[i]]: subscript out of bounds. dbbmmwindowsize may be too large relative to nrow of that individual. Try lowering movemargin (default 11, has to be odd) and then lowering dbbmmwindowsize (default 23, has to be >=2*movemargin, has to be odd).

Value

Individual-level utilization distributions, saved as rasters, as well as calculated volume area estimates for 50 and 95pct contours, saved in a .csv file. Motion variance csvs per individual ("MotionVariance.csv"), see move::brownian.motion.variance.dyn. No processed object is returned, i.e. bad: "objectname <- movegroup()", good: "movegroup()".

Author(s)

Simon Dedman, <simondedman@gmail.com>

Maurits van Zinnicq Bergmann, <mauritsvzb@gmail.com>

References

Kranstauber, B., Kays, R., LaPoint, S. D., Wikelski, M. and Safi, K. (2012), A dynamic Brownian bridge movement model to estimate utilization distributions for heterogeneous animal movement. Journal of Animal Ecology. doi: 10.1111/j.1365-2656.2012.01955.x

Kranstauber, B., M. Smolla & A. K. Scharf. 2019. Move: visualizing and analyzing animal track data. R package version 4.2.4 (at 2023-08-15). https://CRAN.R-project.org/package=move.

Examples

```
# load data
data("TracksCleaned")
# run function
movegroup(
 data = TracksCleaned,
 ID = "Shark",
 Datetime = "Datetime",
 Lat = "Lat",
```

```
Lon = "Lon",
savedir = tempdir())
```

`moveLocErrorCalc` *moveLocError calculator for ARGOS or state space models resulting in 95percent latlon confidence intervals*

Description

Builds a dataframe of original locations plus rowmeans of mean distance of location extremities lon975, lat; lon025, lat; lon, lat975; lon, lat025 from the centre point lon, lat.

Usage

```
moveLocErrorCalc(
  x,
  loncol = "lon",
  latcol = "lat",
  latloncrs = 4326,
  projectedcrs = 32617,
  lon025 = "lon025",
  lon975 = "lon975",
  lat025 = "lat025",
  lat975 = "lat975"
)
```

Arguments

`x`	Data frame or tibble with lats and lons and their high and low confidence interval counterparts.
`loncol`	Name of longitude column in x, character. Default "lon".
`latcol`	Name of latitude column in x, character. Default "lat".
`latloncrs`	CRS of x, default 4326, numeric.
`projectedcrs`	CRS to project to, should match your region, default 32617, numeric. See movegroup projectedCRS.
`lon025`	Name of low 2.5% confidence interval longitude column in x, character. Default "lon025".
`lon975`	Name of high 2.5% confidence interval longitude column in x, character. Default "lon975".
`lat025`	Name of low 2.5% confidence interval latitude column in x, character. Default "lat025".
`lat975`	Name of high 2.5% confidence interval latitude column in x, character. Default "lat975".

Details

Use on your data object from movegroup::movegroup(data).

Value

Dataframe of original locations plus rowmeans of mean distance of location extremities, for use in movegroup::movegroup(moveLocError).

Author(s)

Simon Dedman, <simondedman@gmail.com>

Examples

```
data(argosFiltered)
myMoveLocError <- moveLocErrorCalc(argosFiltered)
```

---

plotraster *Plots a group-level utilization distribution*

Description

Plots 50 and 95pct contours of a group-level utilization distribution raster on a spatial map background. Contains functionality to also visualize geographic locations of individual listening stations (e.g., acoustic receivers) as well as the entire surface UD.

Usage

```
plotraster(
  x = file.path("Scaled", "All_Rasters_Scaled_Weighted_UDScaled.asc"),
  crsloc = NULL,
  xlatlon = NULL,
  locationpoints = NULL,
  calcCOA = FALSE,
  pointsincontourssave = NULL,
  trim = TRUE,
  myLocation = NULL,
  googlemap = FALSE,
  gmapsAPI = NULL,
  expandfactor = 1.6,
  mapzoom = NULL,
  mapsource = "google",
  maptype = "satellite",
  contour1colour = "red",
  contour2colour = "orange",
  positionscolour = "white",
```

```
  positionsalpha = 0.33,
  positionssize = 0.1,
  COAcolour = "black",
  COAalpha = 1,
  COAshape = 4,
  COAsize = 1,
  plottitle = "Aggregated 95% and 50% UD contours",
  plotsubtitle = "Scaled contours",
  legendtitle = "Percent UD Contours",
  plotcaption = paste0("movegroup, ", lubridate::today()),
  axisxlabel = "Longitude",
  axisylabel = "Latitude",
  legendposition = c(0.105, 0.8),
  fontsize = 12,
  fontfamily = "Times New Roman",
  filesavename = paste0(lubridate::today(), "_dBBMM-contours.png"),
  savedir = tempdir(),
  receiverlats = NULL,
  receiverlons = NULL,
  receivernames = NULL,
  receiverrange = NULL,
  recpointscol = "black",
  recpointsfill = "white",
  recpointsalpha = 0.5,
  recpointssize = 1,
  recpointsshape = 21,
  recbufcol = "grey75",
  recbuffill = "grey",
  recbufalpha = 0.5,
  reclabcol = "black",
  reclabfill = NA,
  reclabnudgex = 0,
  reclabnudgey = -200,
  reclabpad = 0,
  reclabrad = 0.15,
  reclabbord = 0,
  surface = TRUE,
  cropsavedimage = FALSE
)
```

Arguments

x	Path to scaled raster generated by movegroup (/scaleraster/alignraster). Likely file.path(movegroupsavedir, "Scaled", "All_Rasters_Scaled_Weighted_UDScaled.asc").
crsloc	Location of saved CRS Rds file from movegroup.R. Likely the same as path. Likely movegroup's savedir.
xlatlon	If you want to also return a csv of your original locations labelled with which UD contours they fall within, include the location of the LatLon raster here. Try

file.path(crsloc, "Scaled", "All_Rasters_Scaled_Weighted_LatLon.asc") . Default NULL will not produce the csv output.

`locationpoints` If you want to also return a csv of your original locations labelled with which UD contours they fall within, include the original input location points of animals, for xlatlon. This should be a data frame which MUST have columns labelled "lat" and "lon".

`calcCOA` Calculate the centre of activity i.e. mean lat & lon point. Requires locationpoints. Default FALSE.

`pointsincontourssave`
Location and name to save the 'location in contours' csv related to xlatlon and locationpoints, including the ".csv".

`trim` Remove NA & 0 UD values and crop the plot to remaining date extents? Shrinks lots of dead space at the edges of the raster. Default TRUE.

`myLocation` Location for extents, format c(xmin, ymin, xmax, ymax). Default NULL, extents auto-created from data. Set this if you want to expand or crop your map to cover a specific area.

`googlemap` If pulling basemap from Google maps, this sets expansion factors since Google Maps tiling zoom setup doesn't align to myLocation extents. Default FALSE.

`gmapsAPI` Enter your google maps API here, quoted character string. Can leave NULL if already registered with ggmap::register_google(). See Details for instructions. If you don't have an API or don't want to get one, leave NULL, ensure mapsource is stamen, and maptype is stamen-compatible.

`expandfactor` Extents expansion factor for basemap. 1.3 to 1.5 are the same zoom as 1. 1.6 is a big leap up in zoom out. 1.9 & maybe 1.7 or 1.8 is another step out. Ignored if not using Google Maps.

`mapzoom` Highest number = zoomed in. Google: 3 (continent) - 21 (building). stamen: 0-18. Default NULL is calculated based on your data.

`mapsource` Source for ggmap::get_map; google needs Google Maps API present. Options "google", "osm", "stadia".

`maptype` Type of map for ggmap::get_map param maptype. Options: Google mapsource: "terrain", "terrain-background", "satellite", "roadmap", "hybrid". Stadia mapsource: "stamen_terrain", "stamen_toner", "stamen_toner_lite", "stamen_watercolor", "stamen_terrain_background", "stamen_toner_background", "stamen_terrain_lines", "stamen_terrain_labels", "stamen_toner_lines", "stamen_toner_labels".

`contour1colour` Colour for contour 1, typically 95pct, default "red".

`contour2colour` Colour for contour 2, typically 50pct, default "orange".

`positionscolour`
Colour for original animal locations, if xlatlon not NULL. Default "white".

`positionsalpha` Alpha value for positions, default 0.33, values from 0 (fully transparent) to 1 (fully parent).

`positionssize` Point size for positions, default 0.1.

`COAcolour` Colour for Centre of Activity marker, if plotted. Default "black".

`COAalpha` Alpha value for Centre of Activity point, default 1, values from 0 (fully transparent) to 1 (fully parent).

COAshape	Shape of Centre of Activity marker, default 4, an X. Permissible values 0 to 25.
COAsize	Size of COA point, default 1.
plottitle	Title of the resultant plot, default "Aggregated 95pct and 50pct UD contours".
plotsubtitle	Plot subtitle, default "Scaled contours". Can add the n of your individuals.
legendtitle	Legend title, default "Percent UD Contours".
plotcaption	Plot caption, default "movegroup" + today's date.
axisxlabel	Default "Longitude".
axisylabel	Default "Latitude".
legendposition	Vector of 2, format c(1,2), Proportional distance of (middle?) of legend box from L to R, percent distance from Bottom to Top. Values 0 to 1. Default c(0.11, 0.85).
fontsize	Font size, default 12.
fontfamily	= Font family, default "Times New Roman".
filesavename	File savename, default today's date + "_dBBMM-contours.png".
savedir	Save outputs to a temporary directory (default) else change to current directory e.g. "/home/me/folder". Do not use getwd() here. No terminal slash. E.g. file.path(movegroupsavedir, "Plot") . Auto-created if it doesn't exist.
receiverlats	Vector of latitudes for receivers to be plotted.
receiverlons	Vector of longitudes for receivers to be plotted. Same length as receiverlats.
receivernames	Vector of names for receivers to be plotted. Same length as receiverlats.
receiverrange	Single (will be recycled), or vector (same length as receiverlats) of detection ranges in metres for receivers to be plotted. If you have a max and a (e.g.) 90 percent detection range, probably use max.
recpointscol	Colour of receiver centrepoint outlines. Default "black".
recpointsfill	Colour of receiver centrepoint fills. Default "white".
recpointsalpha	Alpha value of receiver centrepoint fills, 0 (invisible) to 1 (fully visible). Default 0.5.
recpointssize	Size of receiver points. Default 1.
recpointsshape	Shape of receiver points, default 21, circle with outline and fill.
recbufcol	Colour of the receiver buffer circle outlines. Default "grey75"
recbuffill	Colour of the receiver buffer circle fills. Default "grey".
recbufalpha	Alpha value of receiver buffer fills, 0 (invisible) to 1 (fully visible). Default 0.5.
reclabcol	Receiver label text colour. Default "black".
reclabfill	Receiver label fill colour, NA for no fill. Default NA.
reclabnudgex	Receiver label offset nudge in X dimension. Default 0.
reclabnudgey	Receiver label offset nudge in Y dimension. Default -200.
reclabpad	Receiver label padding in lines. Default 0.
reclabrad	Receiver label radius in lines. Default 0.15.
reclabbord	Receiver label border in mm. Default 0.

surface	Plot complete UD surface along with contours. Default TRUE.
cropsavedimage	Crop the output image with knitr::plot_crop which uses pdfcrop on PDFs, otherwise magick::image_trim. magick requires system pre-install. deb: libmagick++-dev (Debian, Ubuntu), rpm: ImageMagick-c++-devel (Fedora, CentOS, RHEL), csw: imagemagick_dev (Solaris), brew: imagemagick@6 (MacOS). Default FALSE.

Details

For plottitle, you can use the term 'home range' when an animal can be detected wherever it goes i.e. using GPS, satellite or acoustic telemetry whereby it is known that acoustic receivers cover the entire home range of the study species. This term is problematic when applied to a passive acoustic telemetry setting where an array of non-overlapping receivers are used to assess local space use patterns i.e. the home range is bigger than the coverage by the acoustic array.

Errors and their origins:

1. Error in gzfile(file, "rb") : cannot open the connection. In addition: Warning message: In gzfile(file, "rb"): cannot open compressed file '/var/folders/dl/etc/ggmap/index.rds', probable reason 'No such file or directory'. Cause: index.rds may not have been created, due to a problem with ggmap::get_map, likely due to your API key failing silently. Filename too long - solve with filename = "whatever" but doesn't do anything. Added issue in github: https://github.com/dkahle/ggmap/issues/346 . API key help: https://github.com/dkahle/ggmap/issues/235 .
2. trying to read file: All_Rasters_Scaled_Weighted_UDScaled.asc: Error in CPL_read_gdal(as.character(x), as.character(options), as.character(driver),: file not found. Check x is correct.
3. Error in grid.Call.graphics...: Empty raster: mapzoom likely set too low, returning no tiles. Increase mapzoom number.
4. Not Found (HTTP 404). Failed to acquire tile /...png: Tiles are unavailable for ocean, and may be unavailable at the chosen zoom level for the specific region of interest. Inspect the output map and try a lower level (number) of mapzoom.

How to get Google map basemaps:

(from https://www.youtube.com/watch?v=O5cUoVpVUjU):

1. Sign up with dev console: a. You must enter credit card details, but won't be charged if your daily API requests stay under the limit. b. Follow the link: https://console.cloud.google.com/projectselector2/apis/dash c. Sign up for Google cloud account (it may auto populate your current gmail), click agree and continue. d. Click the navigation email in the top left corner and click on Billing. e. Create a billing account – they will NOT auto charge after trial ends. f. Enter information, click on 'start my free trial'. They may offer a free credit for trying out their service. More pricing details: https://mapsplatform.google.com/pricing/ . g. Click "Select a Project" then "New project" in the top right corner. h. Enter Project Name, leave Location as is, click "Create". i. You should now see your project name at the top, where the drop-down menu is.
2. Enable Maps and Places API: a. Click 'Library' on the left. b. In the search field type "Maps" . c. Scroll down, click "Maps Java Script API". d. Click Enable. e. Click 'Library' again, search "Places", click on "Places API". f. Click Enable.
3. Create Credentials for API Key: a. Return to 'APIs & Services' page. b. Click on Credentials. c. At the top click 'Create Credentials > API Key'. d. API key should pop up with

option to copy it. e. You can restrict the key if you want by following steps 4 & 5 here: https://www.youtube.com/watch?v=O5cUoVpVUjU&t=232s

Value

Individual-level utilization distributions, saved as rasters, as well as calculated volume area estimates for 50 and 95pct contours, saved in a .csv file.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Maurits van Zinnicq Bergmann, <mauritsvzb@gmail.com>

Examples

```
# Having run the movegroup and scaleraster function examples:
plotraster(
  x = file.path(tempdir(), "Scaled", "All_Rasters_Scaled_Weighted_UDScaled.asc"),
  mapzoom = 14,
  mapsource = "stamen",
  maptype = "terrain",
  savedir = file.path(tempdir(), "Plot"),
  xlatlon = file.path(tempdir(), "Scaled", "All_Rasters_Scaled_Weighted_LatLon.asc"),
  locationpoints = TracksCleaned |> dplyr::rename(lat = "Lat", lon = "Lon"),
  pointsincontourssave = file.path(tempdir(), "Scaled", "pointsincontours.csv"))
```

---

scaleraster *Scales individual utilization distribution rasters and volume area estimates*

Description

Scales individual-level utilization distribution (UD) rasters from 0 to 1 to facilitate interpretation as relative intensity of utilization (as opposed to absolute), making comparisons across individuals and interpretations at the group level more straightforward. Subsequently, scaled individual-level rasters are aggregated to create a single group-level UD raster. See www.GitHub.com/SimonDedman/movegroup for issues, feedback, and development suggestions. There is an option to account for bias in acoustic receiver array spatial representation (see Details).

Usage

```
scaleraster(
  path = NULL,
  pathsubsets = NULL,
  pattern = ".asc",
  weighting = 1,
  format = "ascii",
```

```
  datatype = "FLT4S",
  bylayer = TRUE,
  overwrite = TRUE,
  scalefolder = "Scaled",
  scaledweightedname = "All_Rasters_Scaled_Weighted",
  crsloc = NULL,
  returnObj = FALSE
)
```

Arguments

path	Path to directory where the individual-level UDs are saved. Likely the same as savedir from movegroup. Default NULL.
pathsubsets	Path to parent directory that contains all UDs across spatial groups or subsets, i.e. if you ran movegroup multiple times for different areas in a connected system, this would be the parent folder within which all the movegroup savedir's are located. Default NULL.
pattern	Extension pattern used to read in all UDs in directory and pathsubsets directory. Default ".asc".
weighting	Addresses unbalanced receiver array design after receivers have first been partitioned into regions, and group-level UDs estimated per region. Numeric. Weights area-specific scaled group-level UD raster by value. This then means that estimated scaled individual-level volume areas also become weighted. Default is 1 for no weighting.
format	Character. Output file type for raster::writeRaster param format. Default "ascii".
datatype	Character. Data type for writing values to disk for raster::writeRaster param Datatype. Default "FLT4S".
bylayer	For raster::writeRaster param bylayer. Default TRUE.
overwrite	For raster::writeRaster param overwrite. Default TRUE.
scalefolder	Folder to save outputs to. Default "Scaled".
scaledweightedname	Name of chunk for scaled and weighted output rasters. Default "All_Rasters_Scaled_Weighted".
crsloc	Location of saved CRS Rds file from movegroup.R. Should be same as path. Default NULL.
returnObj	Logical. Return the scaled object to the parent environment? Default FALSE.

Details

Step 1. Scale rasters. Individual-level UD rasters are scaled from 0 to 1 by dividing each raster by the maximum probability density value occurring within the raster set.

Step 2. Aggregate into a group-level raster. Scaled individual-level rasters are summed to create a single group-level UD raster.

Step 3. Re-scale to 0 to 1. The group-level raster is divided by its own maximum value.

Step 4. Weight raster (optional). The scaled group-level UD raster is divided by the specified weighting factor(s). Note that this is only useful if you want to account for an unbalanced listening

station (e.g., acoustic receivers) array and have split up the study site and receivers in regions, and have run the movegroup() for each regional data set separately. See van Zinnicq Bergmann et al. 2022 (https://doi.org/10.1016/j.biocon.2022.109469) for example. If not applicable, choose a value of "1".

Step 5. Standardize raster. Standardize the potentially weighted and scaled group-level UD raster so that its values sum to 1.

Step 6. Export as latlon CRS. Change crs to latlon for plotting and calculation purposes, save file, continue.

Step 7. Estimate 50 and 95pct contour volume areas. For each scaled individual-level UD raster, estimate 50 and 95pct contour volume areas, as well as their mean and standard deviation. Additionally, the 50 and 95pct volume area is estimated for the group-level UD raster.

Step 8. Export the projected-CRS group-level raster.

Errors and their origins:

1. Error in (function (cond): error in evaluating the argument 'x' in selecting a method for function 'res': subscript out of bounds. Probably path can't find any files of type=pattern: check you used a terminal slash in savedir in movegroup, and that path has files of type=pattern.
2. Error in if (substr(x = pathsubsets, start = nchar(pathsubsets), stop = nchar(pathsubsets))==: argument is of length zero: pathsubsets is wrong. Try setting to same as path. NULL does this.
3. Error in gzfile(file, "rb"): cannot open compressed file 'CRS.Rds', probable reason 'No such file or directory': crsloc is wrong. Try setting to same as path. NULL does this.
4. In min/max: No non-missing arguments to min; returning Inf: likely not enough memory, increase rasterResolution value.

Value

Scaled and weighted individual-level and group-level utilization distributions saved as rasters. Scaled 50 and 95pct contour volume area estimates (in km2) for individuals and the group , saved in .csv format. Latlon raster.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Maurits van Zinnicq Bergmann, <mauritsvzb@gmail.com>

Examples

```
# Having run the movegroup function example:
scaleraster(path = tempdir())

# Weighted by number of positions per ID, fewer locations = lower Weighting value = higher final
# UD values after dividing by Weighting. This scales all IDs up to match the group max.
Weighting <- TracksCleaned |>
 dplyr::group_by(Shark) |>
 dplyr::summarise(N = n()) |>
 dplyr::filter(N > 23) |>
```

```
  dplyr::mutate(N = N / max(N, na.rm = TRUE)) |>
  dplyr::pull(N)

  scaleraster(path = tempdir(), weighting = Weighting)
```

TracksCleaned *Data: Tracks of lemon sharks off Bimini, Bahamas*

Description

Tracks of 17 lemon sharks (Negaprion brevirostris) tagged off Bimini, Bahamas, 2012:2014, by Bimini Biological Field Station employees and volunteers, with accompanying tidal phase.

Usage

```
data(TracksCleaned)
```

Format

A data frame with 1308 rows and 5 variables:

Datetime POSIXct datetime, format YYYY-MM-DD HH:MM:SS.

Shark Individual shark ID code.

T.Ph Tidal phase, H M L High Medium Low.

Lat Decimal latitudes.

Lon Decimal longitudes.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Maurits van Zinnicq Bergmann, <mauritsvzb@gmail.com>

Source

https://www.biminisharklab.com

Index