

Package: movegroup (via r-universe)

July 2, 2026

Title Visualizing and Quantifying Space Use Data for Groups of Animals

Version 2026.07.01

Description Offers an easy and automated way to scale up individual-level space use analysis to that of groups. Contains functions from the 'move' package to calculate either a dynamic Brownian bridge movement model or dynamic Bivariate Gaussian bridges 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 7.3.2

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>>), Vital Heim [aut]

(ORCID: <<https://orcid.org/0000-0002-8492-9196>>)

Maintainer Simon Dedman <simonededman@gmail.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-02 03:20:14 UTC

RemoteUrl <https://github.com/cran/movegroup>

RemoteRef HEAD

RemoteSha 58790fab5064b666de009b89171e02a89ab7efb8

Contents

alignraster	2
argosFiltered	4
movegroup	5
moveLocErrorCalc	11
plotraster	13
scaleraster	19
stitchraster	21
TracksCleaned	22

Index	24
--------------	-----------

alignraster	<i>Combines region-specific group-level UD rasters into a single raster.</i>
-------------	------------------------------------------------------------------------------

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	<i>Data: Tracks of two great hammerhead sharks with position confidence intervals</i>
---------------	---------------------------------------------------------------------------------------

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/SimonDedman/SavingTheBlue/blob/main/R/06A_Filter_SPOT_data.R and https://github.com/SimonDedman/SavingTheBlue/blob/main/R/06A_Filter_SPOT_data.R.

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	<i>Automates dynamic Brownian bridge movement model and dynamic bivariate Gaussian bridge construction across individuals</i>
-----------	-------------------------------------------------------------------------------------------------------------------------------

Description

Automates dynamic Brownian bridge movement model (dBBMM) and dynamic bivariate Gaussian bridge (dBBGB) 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 Kranstauber, 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,  
  model = "dbbmm",  
  ID = NULL,  
  Datetime = NULL,  
  Lat = NULL,  
  Lon = NULL,  
  dat.TZ = "US/Eastern",  
  proj = sp::CRS("+proj=longlat +datum=WGS84"),  
  sensor = "acoustic-telemetry",  
  integrationStep = NULL,  
  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 = NULL,  
  dbbext = 3,  
  dbbwindowsize = NULL,  
  writeRasterFormat = "ascii",  
  writeRasterExtension = ".asc",  
  writeRasterDatatype = "FLT4S",  
  absVolumeAreaSaveName = "VolumeArea_AbsoluteScale.csv",  
  savedir = tempdir(),  
  saveAreaCT = FALSE,  
  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.
model	Character. Movement model to use for utilization distribution estimation. Either "dbbmm" (dynamic Brownian bridge movement model, default) or "dbgb" (dynamic bivariate Gaussian bridge). Please familiarize yourself with the different models using the references provided in the details before making a decision on which model you choose. Remember, in dBBMMs the animal's movement between two locations is equally likely to happen in all directions, i.e. is an isotropic diffusive process. In dBGBs, the diffusion is factorized in two directions, i.e. the direction towards the next location, and the direction perpendicular to it. The uncertainty in your animals whereabouts between locations when the time gap between two locations is large, can therefore have different extents and patterns depending on your model choice, and depending on the temporal resolution and dynamics, your data may be more suitable for one model or the other. Please also see our suggestion on how to quickly and qualitatively compare the two models and their suitability for your data in the Details below.
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 <code>sp::CRS("+proj=longlat +datum=WGS84")</code> .
sensor	Sensor for move function. Single character or vector with length of the number of coordinates. Optional. Default "acoustic-telemetry". See https://www.movebank.org/cms/movebank-content/movebank-attribute-dictionary#event_attributes for possible sensor types. This is just for personal record, and not used in any calculations.
integrationStep	corresponds to <code>time.step</code> and <code>timeStep</code> in <code>move::brownian.bridge.dyn</code> and <code>move::dynBGB</code> , respectively. I.e. corresponds to the size of the timer intervals taken for every integration step in minutes. If left NULL, this function uses the default values from the 'move' package where 15 (for dBBMM) or 20.1 (for dBGB) steps are taken in the shortest time interval For example, in dBBMM-calculations this means that if there is a location recorded e.g. every 60 minutes, the function divides each segment into 4 minute chunks upon which it does the calculation. However, please be advised, that if you have even just one very short time interval, each segment of the track will be divided in the nr. of chunks based on this one short time interval. This can markedly increase computing time. Check the time intervals in your data first. As quote from the 'move' vignette: "If the track contains time intervals much shorter than the scheduled on the tag, set the <code>time.step</code> e.g. to the scheduled time interval at which the tag was set (in minutes). Higher values to this argument reduce calculation time." The here provided function automatically divides your chosen duration by 15 or 20.1 depending on your model choice.

moveLocError	location.error (m) in the 'brownian.bridge.dyn' function and locErr (m) in the 'dynBGB' 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 units set via 'timeDiffUnits' designating the length of long breaks in re-locations. Allows setting movement segments that correspond to long time gaps to FALSE so they are ignored in subsequent model calculations avoiding model and UD inflation. Default of 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 = 50m ² = 0.00005 km ² (divide by 1000000) = 0.00045 degrees. 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. It is also used in move::dynBGBvariance for the behavioural change point analysis and in move::dynBGB. Must be an odd number. Depending on model choice there are different defaults: dbbmm - 11, dbgb - 15. Motion variance based on only

the middle section of the trajectory; the ends of the movement trajectory where no changes are allowed because at some stage you want to have a few locations to base your estimation of the variance on and how many locations in either side of the window we use for this, is called the margin. Smaller values for window size and margin is expected to give a higher frequency of behavioural changes; make these large for looking at migrations.

dbbext	Ext param in the 'brownian.bridge.dyn' and 'dynBGB' functions in the 'move' package. Extends bounding box around track. Numeric single (all edges), double (x & y), or 4 (xmin xmax ymin ymax) . Default 0.3 - extends bounding box by 30 percent. Excessive buffering will get cropped automatically.
dbbwindowSize	The window.size param in the 'brownian.bridge.dyn' and windowSize param in the 'dynBGB' functions 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. For example: a window size of 31 means the sliding window is moved every 31 locations or every 31 hours (via timeDiffUnits). This number must be odd. Must be $\geq 2 * \text{movemargin}$. Given the different 'movemargin' defaults for dBBMM (11) and dBGB (15), defaults for this parameter are set to 23 and 31 depending on your model choice 'dbbmm' and 'dbgb', respectively. Individuals with fewer detections than the window size will be removed.
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. Default tempdir().
saveAreaCT	Save tiny individual core and general use areas tables to disk. These are the only things retained in the per-individual loop, so if your large dataset causes memory crashes, you can run it in chunks and stitch the results together later with stitchraster. Default FALSE.
alerts	Audio warning for failures. Default TRUE.

Details

We strongly recommend that prior to choosing any of the model options, you familiarize yourself with the underlying concepts of dBBMMs and dBGBs by reading the corresponding publications from Kranstauber et al. (2012) for dBBMMs and Kranstauber et al. (2014) for dBGBs. Please see the full reference details in the 'References' section below.

Here we provide a very short overview of the most important points from the listed references. Dynamic BBMMs and dBGBs are an improvement over traditional UD calculation methods, as they are able to integrate a temporal component to the movement data and explicitly model the movement trajectory between two consecutive locations. The dBBMM does this by defining the variance of the Brownian motion between two locations and estimating how irregular the movement path is. Importantly, in dBBMMs the movement between locations is assumed to be diffusive and isotropic. Effectively, this means that the movement between two locations is equally likely to happen in all directions. Dynamic BGBs allow the incorporation of a directional bias in the movement path analysis of bridging models. Kranstauber et al.'s Bivariate Gaussian Bridges model does this by factorizing the diffusion, i.e. the Brownian variance estimates, into the direction towards and perpendicular to a straight line to the next location, resulting in two normally distributed probability densities for each direction.

In either model choice, long time gaps between two consecutive detections can therefore present an issue, as these long time gaps create a very high uncertainty in regard to your animal's whereabouts between two locations. However, given the different underlying assumptions in regard to the diffusion process in each model, longer time gaps can cause different problematic outputs. We would like to thank Dr. Anne K. Scharf for the further explanations regarding the model choice. In short, dBBMM does not distinguish between the animal changing movement in direction or speed between two locations. But, dBGBs make a difference between the animal changing movement in direction and in speed between two locations. Long time gaps in dBGBs can therefore create stronger uncertainty in change in direction so that the outputs of your model runs might show biologically unrealistic, very narrow and long peaks along or perpendicular to the trajectory between two locations, whereas such a case in dBBMMs would create a uninformative, inflated circular area with a large diameter within which your animal might have been (i.e. pretty much anywhere, which is not informative).

The solution to this problem is in both cases to remove the variance of the segments corresponding to the large time gaps. This can be done using the 'move' package, and is directly implemented in this function already (please see params `timeDiffLong` and `timeDiffUnits` above). However, the ideal settings may be different based on your model choice due to the reasons explained above. And, based on your data structure, your data may be inadequate for your model choice. It is therefore imperative that you understand the assumptions of the different models and how they relate to your own data.

Once you have decided, which model is most appropriate for your data, we here provide the `movegroup::movegroup()` function that automates the following steps in calculating dBBMM/dBGB-based UDs:

Step 1. Filters individuals. Remove those individuals for which there are insufficient data i.e. number of re-locations is smaller than the window size parameter value (defaults: 23 for model "dbbmm" and 31 for model "dBGB").

Step 2. Generates universal raster. Based on all remaining data, a universal raster is generated where the calculated UDs are plotted into.

Step 3. Loops 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
```

The movegroup package provides additional functions to automate and simplify the visualisation and quantification of space use data of a group or populations of animals. When used together, the order of these functions would be: movegroup, (stitchraster if required), scaleraster, (alignraster if required then stitchraster again) , plotraster.

Errors and their origins::

1. Error in .local(object, raster, location.error = location.error, ext = ext: Higher y or x 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[[dtm]]: 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 'Date-Time' 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. dbmmwindowsize may be too large relative to nrow of that individual. Try lowering movemargin (default 11, has to be odd) and then lowering dbmmwindowsize (default 23, has to be $\geq 2 * \text{movemargin}$, has to be odd).
8. Error in validityMethod(as(object, superClass)): The used raster is not a UD (sum unequal to 1), sum is NaN. Potentially from memory overrun from large datasets. Close all other programs, restart your session, remove other objects from R, and try again, watching the RAM usage piechart by the Environment tab. If unsuccessful, run function in chunks of individuals until all (of length $\geq \text{windowsize}$) are saved to disc as asc files, then produce final summary stats with stitchraster.
9. Error in validityMethod(as(object, superClass)) : The used raster is not a UD (sum unequal to 1), sum is: NaN One possible cause is loss of accuracy due to writing raster to disk with datatype FLT4S this can be solved preventing disk usage or changing data type. Same error message as in nr. 8 above. This error can also occur if you are calculating dBGBs and set your timeDiffLong shorter than the shortest segment. Check your time gaps between locations and compare to timeDiffLong, check the timeDiffUnits parameter and make sure you chose the correct unit. Try shorter timeDiffLong options.
10. Function runs but no findable output: The default directory into which outputs are saved (i.e. savedir) is set to 'tempdir()'. This means, that if you forget to change this parameter, all outputs will be saved to your system's temporary folder.

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 ("Motion-Variance.csv"), see move::brownian.motion.variance.dyn or move::dynBGBvariance. 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>
Vital Heim, <vital.heim@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., Safi, K., and Bartumeus, F. (2014). Bivariate Gaussian bridges: directional factorization of diffusion in Brownian bridge models. *Movement Ecology*. <https://doi.org/10.1186/2051-3933-2-5>

Kranstauber, B., M. Smolla & A. K. Scharf. 2019. Move: visualising and analysing 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	<i>moveLocError calculator for ARGOS or state space models resulting in 95percent latlon confidence intervals</i>
------------------	-------------------------------------------------------------------------------------------------------------------

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 95% utilization distribution (UD) contours of a group-level UD 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,
  stadiaAPI = NULL,
  expandfactor = 1.6,
  mapzoom = NULL,
  mapsource = "google",
  maptype = "satellite",
  generalUDpct = 0.95,
  coreUDpct = 0.5,
  contour1colour = "red",
  contour2colour = "orange",
  positionscolour = "white",
  positionsalpha = 0.33,
  positionssize = 0.1,
  COAcolour = "black",
  COAalpha = 1,
  COAshape = 4,
  COAsize = 1,
  plottitle = NULL,
  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(), "_UD-contours.png"),
savedir = tempdir(),
receiverlats = NULL,
receiverlons = NULL,
receivernames = NULL,
receiverrange = NULL,
recpointscol = "black",
recpointssfill = "white",
recpointssalpha = 0.5,
recpointsssize = 1,
recpointssshape = 21,
recbufcol = "grey75",
recbuffill = "grey",
recbufalpha = 0.5,
reclabcol = "black",
reclabfill = NA,
reclabnudge = 0,
reclabnudgey = -200,
reclabpad = 0,
reclabrad = 0.15,
reclabbord = 0,
surface = TRUE,
surfaceUDpct = 0.95,
cropsavedimage = FALSE,
savewidth = 6,
saveheight = NULL
)

```

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.
stadiaAPI	Enter your stadia API here, quoted character string. Can leave NULL if already registered with ggmap::register_stadiamaps(). 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".
generalUDpct	numerical. Utilization Distribution % in decimals for the contour level depicting general space use areas within the UD surface. Typically 95% UD, i.e. 0.95. Default 0.95. Contourlines colored in 'contour1color'.
coreUDpct	numerical. Utilization Distribution % in decimals for the contour level depicting core space use areas within the UD surface. Typically 50% UD, i.e. 0.5. Default 0.5. Contourlines colored in 'contour2color'.
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, Optional. Default NULL. Optional plot title. If NULL, title will be generated automatically as "Aggregated 'generalUDpct' % and 'coreUDpct' % 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 + "_UD-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".
recpointfill	Colour of receiver centrepoint fills. Default "white".
recpointsalph	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 the surface of the 'surfaceUDpct' UD on a continuous scale along with contours. Default TRUE.
surfaceUDpct	numerical; Percent of UD surface to be plotted if 'surface' is TRUE in decimals. Tracking datasets of gill-breathing animals are often characterized by relatively long time gaps between consecutive detections. These long gaps can cause higher uncertainties in the animals' whereabouts between these detections causing the inclusion of raster cells with relatively low UD values within the complete UD surface. These very low UD values contain little information and this parameters provides the option to trim the UD surface to a desired UD %. Default is 0.95. This means the 95% UD surface is plotted on a continuous scale.
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.
savewidth	Width of saved output image in inches, default 6.
saveheight	Height of saved output image in inches, default NULL, will calculate optimal height based on aspect ratio.

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.

Please be aware,

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_UDSscaled.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/dashboard>. 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>

How to get Stadia Maps basemaps:

<https://client.stadiamaps.com/signup/> Get an account, copy your API key.

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>

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

Examples

```
# Having run the movegroup and scaleraster function examples:
plotraster(
  x = file.path(tempdir(), "Scaled", "All_Rasters_Scaled_Weighted_UDSscaled.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	<i>Scales individual utilization distribution (UD) rasters and volume area estimates</i>
-------------	------------------------------------------------------------------------------------------

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 UD's are saved. Likely the same as savedir from movegroup. Default NULL.
pathsubsets	Path to parent directory that contains all UD's 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 UD's in directory and pathsubsets directory. Default ".asc".
weighting	Addresses unbalanced receiver array design after receivers have first been partitioned into regions, and group-level UD's 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.
scalegroup	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

We here provide the `movegroup::scaleraster()` function that automates the following steps to scale individual-level utilisation distribution (UD) rasters from 0 to 1 for interpretation as relative intensity of utilization, and to subsequently aggregate scaled individual-level rasters into one single group-level UD raster:

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>

Vital Heim, <vital.heim@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)
```

stitchraster

Stitch together movegroup data individuals core and general use areas

Description

If over-large datasets cause RAM crashes for movegroup, one can run batches of individuals in movegroup then join the individual saved area.ct csv files.

Usage

```

stitchraster(
  data = NULL,
  ID = NULL,
  absVolumeAreaSaveName = "VolumeArea_AbsoluteScale.csv",
  savedir = tempdir()
)

```

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".
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().

Details

See www.GitHub.com/SimonDedman/movegroup for issues, feedback, and development suggestions. Install 'move' development version with: `remotes::install_git('https://gitlab.com/bartk/move.git')`

Parameters values should match those used in `movegroup`.

Value

Calculated volume area estimates for 50 and 95pct contours csv.

Author(s)

Simon Dedman, <simondedman@gmail.com>

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, <simonedman@gmail.com>

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

Source

<https://www.biminisharklab.com>

Index

* datasets

- argosFiltered, [4](#)
- TracksCleaned, [22](#)

alignraster, [2](#)
argosFiltered, [4](#)

movegroup, [5](#)
moveLocErrorCalc, [11](#)

plotraster, [13](#)

scaleraster, [19](#)
stitchraster, [21](#)

TracksCleaned, [22](#)