

Package ‘EMbC’

July 21, 2025

Type Package

Title Expectation-Maximization Binary Clustering

Version 2.0.4

Date 2023-09-26

Author Joan Garriga, John R.B. Palmer, Aitana Oltra, Frederic Bartumeus

Maintainer Joan Garriga <jgarriga@ceab.csic.es>

Description Unsupervised, multivariate, binary clustering for meaningful annotation of data, taking into account the uncertainty in the data. A specific constructor for trajectory analysis in movement ecology yields behavioural annotation of trajectories based on estimated local measures of velocity and turning angle, eventually with solar position covariate as a day-time indicator, (‘`Expectation-Maximization Binary Clustering for Behavioural Annotation’’).

URL <doi:10.1371/journal.pone.0151984>

License GPL-3 | file LICENSE

Imports Rcpp (>= 0.11.0), sp, methods, RColorBrewer, mnormt, suntools

Suggests move, sf, rgl, knitr

LinkingTo Rcpp, RcppArmadillo

LazyData true

VignetteBuilder knitr

RoxygenNote 7.2.3

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-10-03 13:40:02 UTC

Contents

EMbC-package	2
binClst-class	4
binClstPath-class	4
binClstPath_instance	5
binClstStck-class	5

binClst_instance	6
bkml	6
bmap	7
chkp	8
cnfm	9
embc	11
expth	12
lblp	12
lkhp	13
pkml	14
pmap	15
rlbl	16
sct3	17
sctr	18
setc	19
slct	20
smth	20
stbc	21
stts	22
varp	23
view	24
x2d	25
Index	26

EMbC-package

Expectation-Maximization binary Clustering package.

Description

The Expectation-maximization binary clustering (EMbC) is a general purpose, unsupervised, multi-variate, clustering algorithm, driven by two main motivations: (i) it looks for a good compromise between statistical soundness and ease and generality of use - by minimizing prior assumptions and favouring the semantic interpretation of the final clustering - and, (ii) it allows taking into account the uncertainty in the data. These features make it specially suitable for the behavioural annotation of animal's movement trajectories.

Details

The method is a variant of the well sounded Expectation-Maximization Clustering (EMC) algorithm, - i.e. under the assumption of an underlying Gaussian Mixture Model (GMM) describing the distribution of the data-set - but constrained to generate a binary partition of the input space. This is achieved by means of the **delimiters**, a set of parameters that discretizes the input features into high and low values and define the binary regions of the input space. As a result, each final cluster includes a unique combination of either low or high values of the input variables. Splitting the input features into low and high values is what favours the semantic interpretation of the final clustering.

The initial assumptions implemented in the EMbC algorithm aim at minimizing biases and sensitivity to initial conditions: (i) each data point is assigned a uniform probability of belonging to

each cluster, (ii) the prior mixture distribution is uniform (each cluster starts with the same number of data points), (iii) the starting partition, (*i.e.* initial delimiters position), is selected based on a global maximum variance criterion, thus conveying the minimum information possible.

The number of output clusters is 2^m determined by the number of input features m . This number is only an upper bound as some of the clusters can be merged along the likelihood optimization process. The EMbC algorithm is intended to be used with not more than 5 or 6 input features, yielding a maximum of 32 or 64 clusters. This limitation in the number of clusters is consistent with the main motivation of the algorithm of favouring the semantic interpretation of the results.

The algorithm deals very intuitively with data reliability: the larger the uncertainty associated with a data point, the smaller the leverage of that data point in the clustering.

Compared to close related methods like EMC and Hidden Markov Models (HMM), the EMbC is specially useful when: (i) we can expect bi-modality, to some extent, in the conditional distribution of the input features or, at least, we can assume that a binary partition of the input space can provide useful information, and (ii) a first order temporal dependence assumption, a necessary condition in HMM, can not be guaranteed.

The EMbC R-package is mainly intended for the behavioural annotation of animals' movement trajectories where an easy interpretation of the final clustering and the reliability of the data constitute two key issues, and the conditions of bi-modality and unfair temporal dependence usually hold. In particular, the temporal dependence condition is easily violated in animal's movement trajectories because of the heterogeneity in empirical time series due to large gaps, or prefixed sampling scheduling.

Input movement trajectories are given either as a `*data.frame*` or a `*Move*` object from the `**move**` R-package. The package deals also with stacks of trajectories for population level analysis. Segmentation is based on local estimates of velocity and turning angle, eventually including a solar position covariate as a daytime indicator.

The core clustering method is complemented with a set of functions to easily visualize and analyze the output:

- * clustering statistics, * clustering scatterplot (2D and 3D) * temporal labeling profile (ethogram),
- * plotting of intermediate variables, * confusion matrix (numerical validation with respect to an expert's labeling), * visual validation with external information (e.g. environmental data), * generation of kml or webmap docs for detailed inspection of the output.

Also, some functions are provided to further refine the output, either by pre-processing (smoothing) the input data or by post-processing (smoothing, relabeling, merging) the output labeling.

The results obtained for different empirical datasets suggest that the EMbC algorithm behaves reasonably well for a wide range of tracking technologies, species, and ecological contexts (e.g. migration, foraging).

Author(s)

Joan Garriga <jgarriga@ceab.csic.es>

binClst-class	<i>Binary Clustering Class</i>
---------------	--------------------------------

Description

binClst is a generic multivariate binary clustering object.

Slots

- X The input data set. A multivariate matrix where each row is a data point and each column is an input feature (a variable).
- U A multivariate matrix with same dimension as X with the values of certainty associated to each corresponding value in X. Ceartainties assign reliability to the data points so that the less reliable is a data point the less its leverage in the clustering. By default certainties are set to one for all variables of all data points.
- stdv A numeric vector with variable specific values for minimum standard deviation.
- m The number of input features.
- k The number of clusters.
- n The number of observations (data points).
- R A matrix with the values delimiting each binary region (the Reference values).
- P A list with the GMM (Gaussian Mixture Model) parameters. Each element of the list corresponds to a component of the GMM and it is a named-sublist itself, with elements '\$M' (the component's mean) and '\$S' (the component's covariance matrix).
- W A n*k matrix with the likelihood weights.
- A A numeric vector with the clustering labels (annotations) for each data-point (the basic output data). Labels are assigned based on the likelihood weights. Only in case of equal likelihoods the delimiters are used as a further criterion to assign labels.
- L The values of likelihood at each step of the optimization process.
- C Default color palette used for the plots. Can be changed by means of the setc() function.

binClstPath-class	<i>Binary Clustering Path Class</i>
-------------------	-------------------------------------

Description

binClstPath is a binClst subclass for fast and easy speed/turn-clustering of movement trajectories. The input trajectory is given as a data.frame with, at least, the columns (timeStamp,longitude,latitude). This format is described in detail in the class constructor [stbc](#). As a binClst subclass, this class inherits all slots and functionality of its parent class.

Slots

- pth A data.frame with the trajectory timestamps and geolocation coordinates, plus eventual extra columns that were included in the input path data frame, (see the [stbc](#) constructor).
- spn A numeric vector with the time intervals between locations (in seconds).
- dst A numeric vector with the distances between locations (in meters). We use loxodromic computations.
- hdg A numeric vector with local heading directions (in radians from North). We use loxodromic computations.
- bursted A logical value indicating whether the binClstPath instance has already been bursted. As bursting can be computationally demanding for long trajectories, an instance is bursted only when a burst wise representation of the trajectory' is requested for the first time, (unless this value is changed to FALSE).
- tracks If bursted=TRUE, a SpatialLinesDataFrame object ("sp" R-package) with the bursted track segments.
- midPoints If bursted=TRUE, a SpatialPointsDataFrame object ("sp" R-package) with the bursted track midpoints.

binClstPath_instance *binClstPath Instance definition*

Description

Unless otherwise specified, a binClstPath instance refers to a binClstPath object itself, as well as its child class binClstMove. The latter inherits all slots and functionality defined for the former.

binClstStck-class *Binary Clustering Stack Class*

Description

binClstStck is a special class for population level speed/turn-clustering of movement trajectories, given either as path data.frames or move objects.

Slots

- bCS A list of either binClstPath or binClstMove objects, depending on how the input paths are given.
- bC A binClst instance with the global speed/turn clustering of the paths in the stack.

binClst_instance	<i>binClst Instance definition</i>
------------------	------------------------------------

Description

Unless otherwise specified, a binClst instance refers to any of the binary clustering objects defined in the package, either a binClst object itself, or any of its child classes, a binClstPath or a binClstMove instance. The latter inherit all slots and functionality defined for the former.

bkml	<i>Generate a burstwise .kml file of a binClstPath_instance.</i>
------	--

Description

bkml generates a burstwise .kml file of a [binClstPath_instance](#), which can be viewed using Google Earth or other GIS software. At first issue, this command can take some time because bursted segmentation has to be computed.

Usage

```
bkml(obj, folder = "embcDocs", markerRadius = 15, display = FALSE)
```

```
## S4 method for signature 'binClstPath'
```

```
bkml(obj, folder = "embcDocs", markerRadius = 15, display = FALSE)
```

Arguments

obj	A binClstPath_instance .
folder	A character string indicating the name of the folder in which the .kml file will be saved. If the folder does not exist it is automatically created, (defaults to '~/embcDocs').
markerRadius	A numeric value indicating the radius of the markers to be plotted, (defaults to 5 pixels).
display	A boolean value (defaults to FALSE) to automatically launch Google-Earth from within R to visualize the generated .kml document. (Google Earth must already be installed on the system. In Windows, it must be associated with the .kml file type.)

Value

The path/name of the saved kml file.

See Also

[pkml](#)

Examples

```
## Not run:
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- generate a burstwise kml of the output --
bkml(mybcp)

## End(Not run)
```

bmap

Generate an *HTML* burstwise webmap of a *binClstPath_instance*.

Description

bmap generates a burstwise .html file map of a *binClstPath_instance* in HTML5, using Google Maps JavaScript API v3 (<https://developers.google.com/maps/documentation/javascript/>). The resulting file can be viewed locally in most browsers (an internet connection is required for displaying the map tiles) or posted online.

Usage

```
bmap(
  obj,
  folder = "embcDocs",
  apiKey = "",
  mapType = "SATELLITE",
  markerRadius = 15,
  display = FALSE
)

## S4 method for signature 'binClstPath'
bmap(
  obj,
  folder = "embcDocs",
  apiKey = "",
  mapType = "SATELLITE",
  markerRadius = 15,
  display = FALSE
)
```

Arguments

obj	A <i>binClstPath_instance</i> .
folder	A character string indicating the name of the folder in which the .html file will be saved. If the folder does not exist it is automatically created, (defaults to '~/embcDocs').

apiKey	A character string specifying the API Key to be passed to the Google Maps server. No Key is needed for using Google Maps JavaScript API v3, but users may wish to specify a key in order to monitor web traffic if the document is being posted online.
mapType	A character string specifying the type of map to be used in the background. This value is passed directly to the Google Maps server, and currently can be set to ROADMAP, SATELLITE, HYBRID, or TERRAIN. (See the Google Maps API documentation for more information.)
markerRadius	A numeric value indicating the radius of the markers to be plotted, (defaults to 5 pixels).
display	A boolean value (defaults to FALSE) to automatically launch the system's default browser from within R to visualize the generated .html document.

Value

The path/name of the saved .html file.

Examples

```
## Not run:
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- generate a burstwise HTML of the output --
bmap(mybcp)

## End(Not run)
```

chkp

Check labeling profile

Description

Plots the labeling profile of a [binClst_instance](#) against a control variable (e.g. environmental information) depicted as background coloured bars.

Usage

```
chkp(obj, ...)
```

S4 method for signature 'binClst'

```
chkp(obj, ctrlLbls = NULL, ctrlClrs = NULL, ctrlLgnd = NULL, lims = NULL)
```


Arguments

<code>obj</code>	A binClst_instance .
<code>...</code>	Parameters <code>ctrlLbIs</code> , <code>ctrlClrs</code> , <code>ctrlLgnd</code> and <code>lims</code> are optional.
<code>ctrlLbIs</code>	A numeric vector with the control labels or a string specifying one of 'height', 'azimuth' or 'both' solar covariates. By default, for a binClstPath_instance it is set to the solar height covariate, regardless it has been used or not for the clustering.
<code>ctrlClrs</code>	A vector of colors to depict the control labeling. At least one colour should be specified for each different control label. By default white/grey colours are used for the default control labels.
<code>ctrlLgnd</code>	A vector of strings identifying the labels for the legend of the plot. They are automatically generated for the solar covariates.
<code>lims</code>	A numeric vector with lower and upper bounds to limit the plot.

Examples

```
# -- apply EMbC to \code{expth} --
mybcp <- stbc(expth)
# -- plot the labeling profile against 'both' solar covariates --
chkp(mybcp,ctrlLbIs='both',ctrlClrs=RColorBrewer::brewer.pal(8,'Oranges')[1:4])
```

cnfm

*Confusion matrix***Description**

cnfm computes the confusion matrix of the clustering with respect to an expert/reference labeling of the data. Also, it can be used to compare the labelings of two different clusterings of the same trajectory, (see details).

Usage

```
cnfm(obj, ref, ...)
```

S4 method for signature 'binClst,numeric'

```
cnfm(obj, ref, ret = FALSE, ...)
```

S4 method for signature 'binClstPath,missing'

```
cnfm(obj, ref, ret = FALSE, ...)
```

S4 method for signature 'binClstStck,missing'

```
cnfm(obj, ref, ret = FALSE, ...)
```

S4 method for signature 'binClst,binClst'

```
cnfm(obj, ref, ret = FALSE, ...)
```

Arguments

<code>obj</code>	A binClst_instance or <code>bnClstStck</code> instance.
<code>ref</code>	A numeric vector with an expert/reference labeling of the data. A second binClst_instance (see details).
<code>...</code>	Parameters <code>ref</code> and <code>ret</code> are optional.
<code>ret</code>	A boolean value (defaults to <code>FALSE</code>). If <code>ret=TRUE</code> the confusion matrix is returned as a matrix object.

Details

The confusion matrix yields marginal counts and Recall for each row, and marginal counts, Precision and class F-measure for each column. The 3x2 subset of cells at the bottom right show (in this order): the overall Accuracy, the average Recall, the average Precision, NaN, NaN, and the overall Macro-F-Measure. The number of classes (expert/reference labeling) should match or, at least not be greater than the number of clusters. The overall value of the Macro-F-Measure is an average of the class F-measure values, hence it is underestimated if the number of classes is lower than the number of clusters.

If `obj` is a [binClstPath_instance](#) and there is a column "lbl" in the `obj@pth` slot with an expert labeling, this labeling will be used by default.

If `obj` is a `binClstStck` instance and, for all paths in the stack, there is a column "lbl" in the `obj@pth` slot of each, this labeling will be used to compute the confusion matrix for the whole stack.

If `obj` and `ref` are both a [binClst_instance](#) (e.g. smoothed versus non-smoothed), the confusion matrix compares both labelings.

Value

If `ret=TRUE` returns a matrix with the confusion matrix values.

Examples

```
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- compute the confusion matrix --
cnfm(mybcp,expth$lbl)
# -- as we have expth$lbl the following also works --
cnfm(mybcp,mybcp@pth$lbl)
# -- or simply --
cnfm(mybcp)
# -- numerical differences with respect to the smoothed clustering --
cnfm(mybcp,smth(mybcp))
```

embc

*General pourpose multivariate binary Clustering (EMbC)***Description**

embc implements the core function of the Expectation-Maximization multivariate binary clustering.

Usage

```
embc(X, U = NULL, stdv = NULL, maxItr = 200, info = 0)
```

Arguments

X	The input data set. A multivariate matrix where each row is a data point and each column is an input feature (a variable).
U	A multivariate matrix with same dimension as X with the values of certainty associated to each corresponding value in X. Certainties assign reliability to the data points so that the less reliable is a data point the less its leverage in the clustering. By default certainties are set to one (no uncertainty in any value in X).
stdv	a vector with bounds for the maximum precision of clusters, given as minimum standard deviation for each variable, (by default is set to <code>rep(sqrt(.Machine\$double.eps),ncol(X))</code>)
maxItr	A limit to the number of iterations in case of slow convergence (defaults to 200).
info	Level of information shown at each step: info=0 (default) shows step likelihood, number of clusters, and number of changing labels; info=1, include clustering statistics; info=2, include delimiters information; info<0, suppress any step information.

Value

Returns a binClst object.

Examples

```
# -- apply EMbC to the example set of data points x2d ---
mybc <- embc(x2d@d)
```

expth	<i>Synthetic path used in the examples</i>
-------	--

Description

A data.frame with a synthetically generated trajectory with column values (timeStamps, longitudes, latitudes, labels) and column headers ('dTm','lon','lat','lbl'). The order of the columns is important. Column headers can be whatever but are expected to be there. The only exception is the header for the labels column: if headed as 'lbl' it will be used automatically by any methods that can make use of it.

Format

See parameter pth of the [stbc](#) constructor.

lblp	<i>labeling profile plot</i>
------	------------------------------

Description

lblp plots the labeling profile of a [binClst_instance](#).

Usage

```
lblp(obj, ref, ...)
```

```
## S4 method for signature 'binClst,missing'
```

```
lblp(obj, ref, lims = NULL, ...)
```

```
## S4 method for signature 'binClstStck,missing'
```

```
lblp(obj, ref, lims = NULL, ...)
```

```
## S4 method for signature 'binClst,numeric'
```

```
lblp(obj, ref, lims = NULL, ...)
```

```
## S4 method for signature 'binClst,binClst'
```

```
lblp(obj, ref, lims = NULL, ...)
```

Arguments

obj	A binClst_instance .
ref	A numeric vector with an expert's labeling profile. A second binClst_instance to be compared with the first.
...	Parameters ref and lims are optional.
lims	A numeric vector with lower and upper bounds to limit the plot.

Examples

```
# -- apply EMbC to the example path --
mybcp <- stbc(expth)
# -- plot the labeling profile comparing with expert labeling --
lblp(mybcp,expth$lbl)
# -- compare original and smoothed labeling profiles --
lblp(mybcp,smth(mybcp))
```

lkhp

*Likelihood profile plots***Description**

lkhp likelihood optimization plot.

Usage

```
lkhp(obj, offSet = 1)

## S4 method for signature 'binClst'
lkhp(obj, offSet = 1)

## S4 method for signature 'list'
lkhp(obj, offSet = 1)
```

Arguments

obj	A BinClst_instance or a list of them.
offSet	A numeric value indicating an offset to avoid the initial iterations. This is useful to see the likelihood evolution in the last iterations where the changes in likelihood are of different order of magnitude than those at the starting iterations.

Examples

```
# -- apply EMbC to the example path --
mybcp <- stbc(expth)
# -- inspect the likelihood evolution --
lkhp(mybcp)
# -- avoid the initial values --
lkhp(mybcp,10)
```

pkml

Generate a pointwise .kml file of a [binClstPath_instance](#)

Description

pkml generates a pointwise KML file of a [binClstPath_instance](#), which can be viewed using Google Earth or other GIS software.

Usage

```
pkml(obj, folder = "embcDocs", markerRadius = 15, display = FALSE, ...)
```

```
## S4 method for signature 'binClstPath'
```

```
pkml(obj, folder, markerRadius, display, showClst = numeric(), ...)
```

Arguments

obj	A binClstPath_instance .
folder	A character string indicating the name of the folder in which the .kml file will be saved. If the folder does not exist it is automatically created, (defaults to '~/embcDocs').
markerRadius	A numeric value indicating the radius of the markers to be plotted, (defaults to 5 pixels).
display	A boolean value (defaults to FALSE) to automatically launch Google-Earth from within R to visualize the generated .kml document. (Google Earth must already be installed on the system. In Windows, it must be associated with the .kml file type.)
...	Parameters folder, markerRadius, display and showClst are optional.
showClst	A numeric vector indicating a subset of clusters to be shown.

Value

The path/name of the saved kml file.

See Also

[bkml](#)

Examples

```
## Not run:
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- generate a pointwise .kml of the output --
pkml(mybcp)
# -- show only stopovers and automatically display the .kml document --
pkml(mybcp,showClst=c(1,2),display=TRUE)
```

```
## End(Not run)
```

pmap

Generate an *HTML* pointwise webmap of a [binClstPath_instance](#).

Description

pmap generates a pointwise .html file-map of a [binClstPath_instance](#) in HTML5, using Google Maps JavaScript API v3 (<https://developers.google.com/maps/documentation/javascript/>). The resulting file can be viewed locally in most browsers (an internet connection is required for displaying the map tiles) or posted online.

Usage

```
pmap(
  obj,
  folder = "embcDocs",
  apiKey = "",
  mapType = "SATELLITE",
  markerRadius = 15,
  display = FALSE
)

## S4 method for signature 'binClstPath'
pmap(
  obj,
  folder = "embcDocs",
  apiKey = "",
  mapType = "SATELLITE",
  markerRadius = 15,
  display = FALSE
)
```

Arguments

obj	A binClstPath_instance .
folder	A character string indicating the name of the folder in which the .html file will be saved. If the folder does not exist it is automatically created, (defaults to '~/embcDocs').
apiKey	A character string specifying the API Key to be passed to the Google Maps server. No Key is needed for using Google Maps JavaScript API v3, but users may wish to specify a key in order to monitor web traffic if the document is being posted online.

mapType	A character string specifying the type of map to be used in the background. This value is passed directly to the Google Maps server, and currently can be set to ROADMAP, SATELLITE, HYBRID, or TERRAIN. (See the Google Maps API documentation for more information.)
markerRadius	A numeric value indicating the radius of the markers to be plotted, (defaults to 5 pixels).
display	A boolean value (defaults to FALSE) to automatically launch the system's default browser from within R to visualize the generated .html document.

Value

The path/name of the saved html file.

Examples

```
## Not run:
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- generate a pointwise HTML of the output --
pmap(mybcp)

## End(Not run)
```

r1bl

Manual relabeling of clusters.

Description

r1bl Manual relabeling of clusters (to merge clusters or relabel merged clusters).

Usage

```
r1bl(obj, old = 0, new = 0, reset = FALSE)
```

```
## S4 method for signature 'binClst'
r1bl(obj, old = 0, new = 0, reset = FALSE)
```

Arguments

obj	A binClst_instance .
old	The number of the cluster to be relabeled.
new	The new number of the cluster.
reset	A boolean value (defaults to FALSE). If reset=TRUE the labeling is reset to the original state.

Details

Whenever two adjacent clusters are merged, the label identifying the splitting variable between them both is meaningless, and the algorithm ends up assigning either a L or H only depending on how it evolved until reaching the merging point. Thus it can happen that the final labeling of the resulting cluster is not the most intuitive one. With this method the labels can be changed as desired. It can also be used to manually force the merging of two clusters.

This method does not return a relabeled copy of the input obj, instead the [binClst_instance](#) itself is relabeled. However, this is intended only for output and visualization purposes (sctr(), lblp(), cnfm(), view()) as the [binClst_instance](#) parameters (GMM parameters and binary delimiters) are not recomputed. Thus the input instance can always be reset to its original state.

Value

This method does not return a relabeled copy of the input obj, instead the [binClst_instance](#) itself is relabeled. It is intended only for visualization purposes, as it does not recompute the GMM parameters nor the binary delimiters of the [binClst_instance](#).

Examples

```
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- manually merge clusters 1 and 2 --
rlbl(mybcp,1,2)
# -- reset to the original state --
rlbl(mybcp,reset=TRUE)
```

sct3

Dynamic 3D-scatterplot

Description

sct3 generates a dynamic 3D-scatterplot of a multivariate [binClst_instance](#), showing clusters in different colors. The scatter plot can be zoomed/rotated with the mouse.

Usage

```
sct3(obj, ...)

## S4 method for signature 'binClst'
sct3(obj, showVars = NULL, showClst = NULL, ...)
```

Arguments

```
obj          A binClst\_instance.
...          Parameters ref, showVars and showClst are optional.
```

showVars	When the number of variables is greater than two, a length 3 numeric vector indicating one splitting variable and two variables to be scattered (given in that order).
showClst	When the number of variables is greater than two, a numeric vector (of variable length) indicating a subset of the clusters that will be shown in the scatter plot. This is useful in case of overlapping clusters.

Details

This function needs the package "rgl" to be installed.

Examples

```
## Not run:
# -- apply EMbC to the example path with scv='height' --
mybcp <- stbc(expth,scv='height')
# -- show a dynamic 3D-scatterplot --
sct3(mybcp)
# -- show only a subset of clusters --
sct3(mybcp,showClst=c(2,4,6))

## End(Not run)
```

sctr

Clustering 2D-scatterplot

Description

sctr generates a scatterplot from a [binClst_instance](#), showing clusters in different colors.

Usage

```
sctr(obj, ...)

## S4 method for signature 'binClst'
sctr(obj, ref = NULL, showVars = NULL, showClst = NULL, bg = NULL, ...)

## S4 method for signature 'binClstStck'
sctr(obj, ref = NULL, showVars = NULL, showClst = NULL, ...)
```

Arguments

obj	A binClst_instance .
...	Parameters ref, showVars and showClst are optional.
ref	A numeric vector with expert/reference labeling for visual validation of the clustering. A second binClst_instance to be compared with the former.

showVars	When the number of variables is greater than two, a length 3 numeric vector indicating one splitting variable and two variables to be scattered (given in that order).
showClst	When the number of variables is greater than two, a numeric vector (of variable length) indicating a subset of the clusters that will be shown in the scatter plot. This is useful in case of overlapping clusters.
bg	A valid colour to be used as background colour for multivariate scatterplots. By default a light-grey colour is used to enhance data points visibility.

Examples

```
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- show the scatterplot compared with expert labeling--
sctr(mybcp,expth$lbl)
```

setc

Sets binClst color palette .

Description

setc sets the color palette to a color family from the RColorbrewer package

Usage

```
setc(bC, fam = "RdYlBu")
```

Arguments

bC	A binClst_instance .
fam	The name of a color family from the Rcolorbrewer R-package, (default color palette is 'RdYlBu' which is colorblind safe and print friendly up to 6 colors).

Examples

```
# -- change the color palette of mybc to "PuOr" --
## Not run:
setc(mybc,'PuOr')

## End(Not run)
```

slct	<i>Select a single path from a binClstStck instance.</i>
------	--

Description

slct selects a single path from a binClstStck instance.

Usage

```
slct(stck, pathNmbr)
```

Arguments

stck	A binClstStck instance.
pathNmbr	The number of the single path to be selected.

Value

Returns the single [binClstPath_instance](#) selected.

Examples

```
## Not run:
# -- select path number 3 in mybcystack --
bcp3 <- slct(mybcystack,3)

## End(Not run)
```

smth	<i>Posterior smoothing of single local labels.</i>
------	--

Description

smth Performs a posterior smoothing of single local labels (locations that differ from their neighbouring locations while the later have equal labels).

Usage

```
smth(obj, dlta = 1)

## S4 method for signature 'binClst'
smth(obj, dlta = 1)

## S4 method for signature 'binClstStck'
smth(obj, dlta = 1)
```

Arguments

obj	Either a <code>binClst_instance</code> or a <code>binClstStck_instance</code> .
dlt	A numeric value in the range (0,1) (default is 1) indicating the user's will to accept a change of label. The change of label is done whenever the decrease in likelihood is not greater than <code>dlt</code> .

Value

A smoothed copy of the input instance. In the case of a `binClstStck_instance` smoothing is performed at population level as well as at each individual trajectory in the stack.

Examples

```
# -- cluster the example path with a prior smooth of 1 hour --
mysmoothbcp <- stbc(expth,smth=1,info=-1)
# -- apply a posterior smoothing --
mysmoothbcpsmoothed <- smth(mysmoothbcp,dlt=0.5)
```

stbc	<i>speed/turn bivariate binary Clustering.</i>
------	--

Description

`stbc` is a specific constructor for movement ecology purposes. By default it implements a bivariate (speed/turn) clustering for behavioural annotation of animals' movement trajectories. Alternatively, it can perform a trivariate clustering by including the solar position covariate (i.e. solar height or solar azimuth) as a daytime indicator.

Usage

```
stbc(
  obj,
  stdv = c(0.1, 5 * pi/180),
  spdLim = 40,
  smth = 0,
  scv = "None",
  maxItr = 200,
  info = 0
)
```

Arguments

obj	A <code>data.frame</code> object with (timeStamp,lon,lat) values in columns 1:3 respectively. Timestamps must be given as <code>POSIXct()</code> with specific format "%Y-%m-%d %H:%M:%S". Further columns of associated data are allowed and will be included in the binClstPath_instance @pth slot. A Move object from the "move" R-package.
-----	---

	A list of trajectories given either as <code>data.frame</code> or <code>Move</code> objects, to perform a joined clustering of all of them. This is mainly intended to perform analysis at population level.
<code>stdv</code>	a vector with bounds for the maximum precision of clusters, given as minimum standard deviation for each variable, (by default is set to 0.1 m/s for velocities and 5 degrees for turns).
<code>spdLim</code>	A speed limit for automatic detection of outliers. Trajectory locations with associated values of speed above the <code>spdLim</code> are not eliminated but will play no part in the clustering. By default is set to 40 m/s.
<code>smth</code>	A smoothing time interval in hours. This is used to estimate local values of speed and turn computed as an average over a time window centered at each location.
<code>scv</code>	A solar position covariate to be used as a daytime indicator. It can be either 'height' (the solar height in degrees above the horizon) or 'azimuth' (the solar azimuth in degrees from north). If it is used, a trivariate clustering is performed, increasing to a maximum of 8 the number of clusters (behaviours) that can potentially be identified. By default this value is set to <code>None</code> (i.e. perform the standard bivariate speed/turn clustering).
<code>maxIter</code>	A limit to the number of iterations in case of slow convergence (defaults to 200).
<code>info</code>	Level of information shown at each step: <code>info=0</code> (default) shows step likelihood, number of clusters, and number of changing labels; <code>info=1</code> , include clustering statistics; <code>info=2</code> , include delimiters information; <code>info<0</code> , suppress any step information.

Value

Returns a `binClstPath` object.

Examples

```
# -- apply EMbC to the example path --
mybcp <- stbc(expth)
## Not run:
# --- binary clustering of a Move object ---
require(move)
mybcm <- stbc(move(system.file("extdata", "leroy.csv.gz", package="move")))
# --- binary clustering of a stack of trajectories ---
mybcm <- stbc(list(mypth1, mypth2, mypth3))

## End(Not run)
```

stts

Clustering statistics.

Description

stts clustering statistics information.

Usage

```

stts(obj, dec = 2, width = 8)

## S4 method for signature 'binClst'
stts(obj, dec = 2, width = 8)

## S4 method for signature 'binClstStck'
stts(obj, dec = 2, width = 8)

```

Arguments

obj	Either a binClst_instance or a binClstStck instance. In the latter case statistics are given at stack level.
dec	The number of decimals for mean/stdv formatting.
width	The number of digits for mean/stdv formatting.

Details

This method prints a line for each cluster with the following information: the cluster number, the cluster binary label, the cluster mean and variance of each input feature (two columns for each variable), and the size of the cluster in number and proportion of points (the posterior marginal distribution).

Examples

```

# -- apply EMbc to the example path with solar covariate 'height'--
mybcp <- stbc(expth,scv='height',info=-1)
# -- show clustering statistics --
stts(mybcp,width=5,dec=1)
## Not run:
# -- show clustering statistics of mybcpstack at stack level --
stts(mybcpstack)
# -- show individual statistics for path number 3 in mybcpstack --
stts(slct(mybcpstack,3))

## End(Not run)

```

varp

Variables' profile plots

Description

varp easy plot of input, output and intermediate variables of a [binClstPath_instance](#).

Usage

```
varp(obj, ...)

## S4 method for signature 'binClstPath'
varp(obj, lims = NULL, ...)

## S4 method for signature 'matrix'
varp(obj, lims = NULL, ...)
```

Arguments

obj	Either a matrix or a binClstPath_instance .
...	Parameter lims is optional.
lims	A numeric vector with lower and upper bounds to limit the plot.

Details

If obj is a matrix, axes labels are automatically generated from the `colnames()` of the matrix, hence they can be changed as desired.

If obj is a [binClstPath_instance](#) it plots the values of the intermediate computations saved in slots `mybcp@spn` (span times), `mybcp@dst` (distances) and `mybcp@hdg` (local heading directions).

Examples

```
# -- apply EMbC to the example path --
mybcp <- stbc(expth,info=-1)
# -- plot clustering data points --
varp(mybcp@X)
# -- plot data points' certainties --
varp(mybcp@U)
# -- plot intermediate computations (span-times, distances and headings) in one figure --
varp(mybcp)
## Not run:
# -- plot only span-times between locations a and b --
plot(seq(a,b),mybcp@spn[a:b],col=4,type='l',xlab='loc',ylab='spanTime (s)')

## End(Not run)
```

view

Path fast view

Description

`view` provides a fast plot of a segmented trajectory or specific chunks of it.

Usage

```
view(obj, ...)

## S4 method for signature 'binClstPath'
view(obj, lbl = NULL, lims = NULL, bg = NULL, ...)

## S4 method for signature 'data.frame'
view(obj, lbl = NULL, lims = NULL, bg = NULL, ...)
```

Arguments

obj	A binClstPath_instance or a data.frame with the format described for slot binClstPath@pth.
...	Parameters lbl and lims are optional.
lbl	A numeric vector with location labels. If obj is a binClstPath_instance the clustering labels are used by default.
lims	A numeric vector with lower and upper limit locations to show only a chunk of the trajectory.
bg	A valid colour to be used as background colour. By default a light-grey colour is used to enhance data points visibility.

Examples

```
# -- Fast view of the binClstPath instance included in the package --
view(expth)
# -- the same with reference labels --
view(expth, lbl=TRUE)
```

x2d

Synthetic 2D object used in the examples

Description

An ad-hoc object with a set of bivariate data points synthetically generated by sampling from a four component GMM and their corresponding labels indicating which component of the mixture generated each data point.

Format

See parameter X of the [embc](#) constructor.

Index

* datasets

expth, [12](#)

x2d, [25](#)

* package

EMbC-package, [2](#)

binClst-class, [4](#)

binClst_instance, [6](#), [8–10](#), [12](#), [16–19](#), [23](#)

binClstPath-class, [4](#)

binClstPath_instance, [5](#), [6](#), [7](#), [9](#), [10](#), [14](#), [15](#),
[20](#), [21](#), [23–25](#)

binClstStck-class, [5](#)

bkml, [6](#), [14](#)

bkml, binClstPath-method (bkml), [6](#)

bmap, [7](#)

bmap, binClstPath-method (bmap), [7](#)

chkp, [8](#)

chkp, binClst-method (chkp), [8](#)

cnfm, [9](#)

cnfm, binClst, binClst-method (cnfm), [9](#)

cnfm, binClst, numeric-method (cnfm), [9](#)

cnfm, binClstPath, missing-method (cnfm),
[9](#)

cnfm, binClstStck, missing-method (cnfm),
[9](#)

EMbC (EMbC-package), [2](#)

embc, [11](#), [25](#)

EMbC-package, [2](#)

expth, [12](#)

lblp, [12](#)

lblp, binClst, binClst-method (lblp), [12](#)

lblp, binClst, missing-method (lblp), [12](#)

lblp, binClst, numeric-method (lblp), [12](#)

lblp, binClstStck, missing-method (lblp),
[12](#)

lkhp, [13](#)

lkhp, binClst-method (lkhp), [13](#)

lkhp, list-method (lkhp), [13](#)

pkml, [6](#), [14](#)

pkml, binClstPath-method (pkml), [14](#)

pmap, [15](#)

pmap, binClstPath-method (pmap), [15](#)

rlbl, [16](#)

rlbl, binClst-method (rlbl), [16](#)

sct3, [17](#)

sct3, binClst-method (sct3), [17](#)

sctr, [18](#)

sctr, binClst-method (sctr), [18](#)

sctr, binClstStck-method (sctr), [18](#)

setc, [19](#)

slct, [20](#)

smth, [20](#)

smth, binClst-method (smth), [20](#)

smth, binClstStck-method (smth), [20](#)

stbc, [4](#), [5](#), [12](#), [21](#)

stts, [22](#)

stts, binClst-method (stts), [22](#)

stts, binClstStck-method (stts), [22](#)

varp, [23](#)

varp, binClstPath-method (varp), [23](#)

varp, matrix-method (varp), [23](#)

view, [24](#)

view, binClstPath-method (view), [24](#)

view, data.frame-method (view), [24](#)

x2d, [25](#)