Reference manual for the HMM geolocation toolbox for MATLAB

Version 0.6

Compilation date December 20, 2007.

Title HMM geolocation toolbox.

Authors Martin Wæver Pedersen, Uffe Høgsbro Thygesen and Henrik Madsen.

Maintainer Martin Wæver Pedersen

Dependencies Matlab version 7.4.0.287 (the toolbox may work with earlier versions of Matlab but has only been tested on this version).
Some plot functions require the M_Map toolbox which can be downloaded for free at www.eos.ubc.ca/ rich/map.html.
The toolbox has only been tested on a tidal database of the North Sea variations and is in the current version therefore unlikely to work in other domains.

Description This toolbox geolocates Data Storage Tags based on depth measurements and tidal data. If you are interested in geolocating fish from light, temperature, salinity or other types of data please send me an email (mwp@imm.dtu.dk) and I will be happy to help you with this.

Contents

1 Introduction **3**
1.1 Installation . . . 3
1.2 Example . . . 3

2 Data formats **3**
2.1 Tidal database . . . 3
2.2 Depth measurements . . . 6
2.3 Release and recapture position . . . 6

3 Preprocessing **7**
3.1 `datastrip` . . . 7
3.2 `tidebehavextr` . . . 8
3.3 `datalikelihood` . . . 9
3.4 `datalikparam` . . . 10

4 Filtering **11**
4.1 `hmmgeolocate` . . . 11
4.2 `hmmfilter` . . . 12

4.3 likelihood . . . 12
4.4 smoothing . . . 12

5 Postprocessing **13**
5.1 avimaker . . . 13
5.2 mptrack . . . 16
5.3 samptrack . . . 17
5.4 proboftrack . . . 18
5.5 paddtrack . . . 19
5.6 plottidebehav . . . 20
5.7 plottrack . . . 21
5.8 plotud . . . 22
5.9 pplotdistr . . . 23
5.10 pplotlik . . . 24
5.11 pplottrack . . . 25
5.12 probofvisit . . . 26
5.13 viewdistr . . . 27

6 Background functions **28**
6.1 fminsearchbnd . . . 28
6.2 gausskern . . . 28
6.3 gausspdf . . . 29
6.4 mvgausspdf . . . 30
6.5 plotting . . . 31
6.6 plottingfancy . . . 31
6.7 plottingbw . . . 31
6.8 lssinfit . . . 31
6.9 makeplotstandard . . . 31
6.10 mapmatrix . . . 31
6.11 maptopix . . . 31
6.12 normalise . . . 32
6.13 normcdf . . . 32
6.14 pixtomap . . . 32
6.15 deglong . . . 32
6.16 m_surf . . . 32
6.17 flipdb . . . 32

1 Introduction

This MATLAB toolbox can be used for data analysis of depth data from archival tags mounted on fish, in order to obtain knowledge of their whereabouts while at liberty, i.e. their geographical position on a daily basis. The method relies purely on depth data and applies advanced statistical methods to estimate not only the most probable track but also the probability distribution for the position. In figure 1, 2 and 3 is shown examples of data and results obtained from the model.

The geolocation method used in this toolbox was developed in the master thesis (Pedersen, 2007) at the Technical University of Denmark, Institute of Informatics and Mathematical Modelling by Martin Wæver Pedersen, supervised by Prof. Henrik Madsen and Dr. Uffe H. Thygesen. The thesis which documents the method is found at the website www.imm.dtu.dk/∼mwp along with a download mirror for the toolbox and a link to this reference manual.

1.1 Installation

Unzip `hmmgeoloc.zip` file into a directory of your choice. You should be aware that working with the toolbox might create large files (>100 MB for long tracks or large databases) so be sure to have plenty of available disk space. Once the file is unzipped start MATLAB and go to the file menu and pick "Set path...". Choose "Add with subfolders.." and find the root directory of the geolocation toolbox (the one you unzipped the files into). This makes MATLAB able to "see" the toolbox even if you are working in another directory. Now click "Save" and "Close" in the "Set path..." window and you should be ready to start geolocating some tags.

1.2 Example

The toolbox comes with an example script file which shows its basic functionality. The example script file is named `codexample.m` and is stored in the `tbworkdir` directory. The example is executed in MATLAB by typing `codexample` in the prompt. The example reads the database files included in the toolbox and the raw tag data file which is also included. It performs the steps that are always required for geolocating a tag. The script file can be viewed, edited and copied from by typing `edit codexample` in the prompt. Remember that there is many more functions than the ones used in this basic example.

This example might be the easiest approach to the toolbox, both for setting up a script file but also for getting comfortable with the format of the database and tag files, which are viewable as well, see section 2.

2 Data formats

This section describes the format of the input data to the toolbox.

2.1 Tidal database

This section describes the requirements to the text files containing the tidal and bathymetry database to be used with the geolocation toolbox. The MATLAB functions used to convert the text files into data files and compute necessary variables are

`readdb.m` - Reads the text files.
`finddbvars.m` - Calculates bathymetry and tidal variance.

Important: The functions (`readdb` and `finddbvars`) should be run in the order they are listed here.

2.1.1 Function `readdb`

The function needs access to the following files

`dbinfo.dat` - Plain text file containing database informations.
`latlongdep.dat` - Plain text file containing lats, longs and depth.
`const*.dat` - Plain text file(s) containing constituent informations (amp and phase).

The function outputs the file `rawtidaldb.mat` which contains the extracted information from the database stored in a struct variable.

Description of `dbinfo.dat` The text file contains informations that are used to properly extract the data from the database. A typical file could look like this

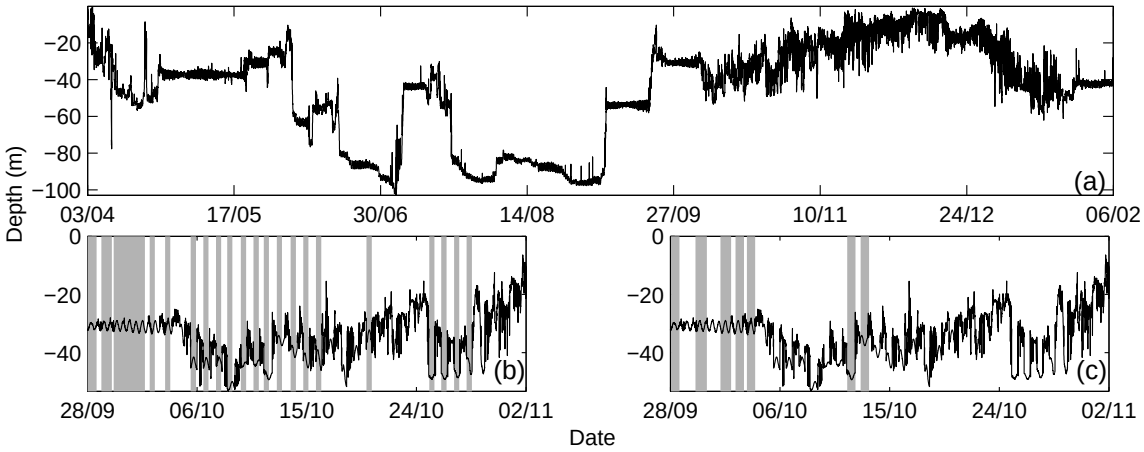

Figure 1: (a) Time series from a tag. (b) Part of the time series classified wrt. tidal information (grey areas). (c) Part of the time series classified wrt. behaviour information (grey areas).

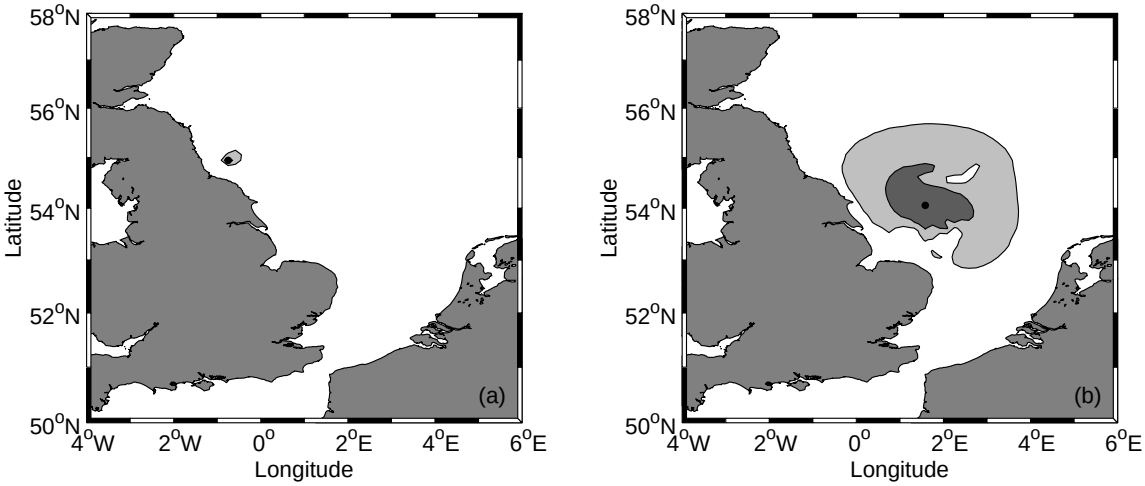

Figure 2: Uncertainty of the geolocation represented as 95% confidence area (light grey) and 50% confidence area (dark grey).

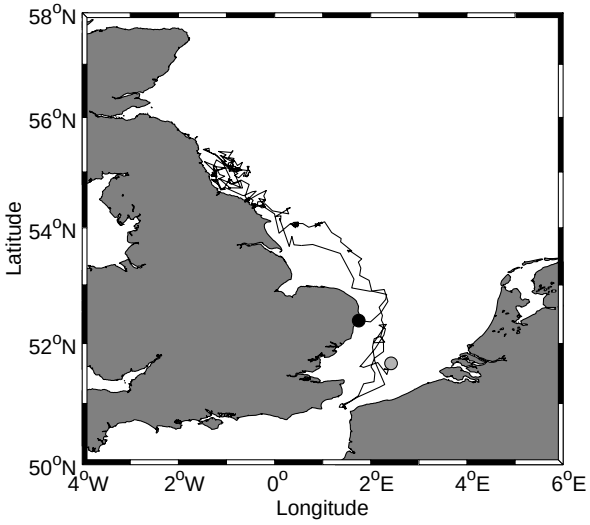

Figure 3: Most probable track, where the black marker is release position, and the grey marker is the recapture position.

```
Land positions indicated by
 -99.0
Nrows    Ncols
  107     119
Nconsts
  7
```

The file is fairly self explanatory. It states how land positions are marked, here by the input `-99.0`, which should be a number that is different from any other occurrences in the bathymetry, lat, long, amp and phase listings. The `Nrows` and `Ncols` states the number of rows and columns in the rectangular database grid. The `Nconsts` state the number of tidal constituents that the program should extract. A corresponding number of constituent files (`const*.dat`) must be available. Blank lines at the beginning and end of the file should be avoided.

Description of `latlongdep.dat` The text file contains the `Nrows*Ncols` latitudinal and longitudinal coordinates of the rectangular grid separated by blank spaces and the depth (bathymetry) in the same format. The first entry of latitude, longitude and depth respectively must correspond to the north west corner of the domain. For the below example `59.944`, `-11.750`, `1203.000` are the values at the north west corner for lat, long and depth respectively. A typical file could look like this (note that `...` denotes the some data is not shown)

```
Lat
59.944   59.944   59.944   59.944   59.944   59.944   59.944   59.944
59.944   59.944   59.944   59.944   59.944   59.944   59.944   59.944
...
48.166   48.166   48.166   48.166   48.166   48.166   48.166   48.166
48.166   48.166   48.166   48.166   48.166
Lon
-11.750  -11.583  -11.417  -11.250  -11.083  -10.917  -10.750  -10.583
-10.417  -10.250  -10.083   -9.917   -9.750   -9.583   -9.417   -9.250
...
 5.917    6.083    6.250    6.417    6.583    6.750    6.917    7.083
 7.250    7.417    7.583    7.750    7.917
Depth
1203.000 1210.000 1217.000 1223.000 1192.000 1161.000 1130.000 1129.000
1128.000 1127.000 1144.000 1139.000 1196.000 1264.000 1311.000 1190.000
...
```

The depth is in metres positive below the sea surface. This is converted to negative below sea surface before storing in the MATLAB datafile. If the user has another format of the text files e.g. depth measured negatively, it might be easier to ignore this at first and just manually change the MATLAB datafile when the data have been read and stored. This can be done by simply loading the datafile (with the `load` command), changing whatever needs to be changed in the loaded variables and overwriting (with the `save` command) the datafile with the changes. It is important that the layout of the text files conforms to the prescription but scales and factors (if needed) may be applied after reading the files. The examples shown here use the default format and units.

Description of `const*.dat` The text file contains the `Nrows*Ncols` amplitude and phase of the tidal constituent along with its name and frequency. The amplitude should be listed in sequence separated by blank spaces. A typical file could look like this (note that `...` denotes the some data is not shown)

```
Name
M2
Freq, degree/hour
28.98411
Amp (cm).
72.900   73.000   73.100   73.200   73.200   73.300   73.400   73.500
73.500   73.600   73.700   73.700   73.800   73.900   74.000   74.200
...
-99.000  -99.000  -99.000  -99.000  -99.000  -99.000  -99.000  -99.000
-99.000  -99.000  -99.000  -99.000  -99.000
Phase (deg).
175.540  176.050  176.680  177.350  178.030  178.730  179.450  180.180
180.940  181.720  182.520  183.340  184.180  185.040  185.930  186.860
...
```

The name of the constituent is read as a string, in this case `M2`. The frequency of the constituent is stated in degree/hour, in this case `28.98411` and converted to radian/day by the program. As stated the amplitude is measured in centi metres

and phase is measured in degrees. These units are converted by the program to metres and radians respectively.

Each constituent the user wishes to include must have its own file. The files must be numbered incrementally, e.g. as `const1.dat`, `const2.dat`, `const3.dat` if the number of constituents (`Nconsts`) was set to three in the `dbinfo.dat` file.

Contents of `rawtidaldb.mat` When loaded in MATLAB the workspace should include the struct variable `db` which should contain the following fields:

`lat`	`[Nrows x Ncols]`	degrees latitude.
`long`	`[Nrows x Ncols]`	degrees longitude.
`depth`	`[Nrows x Ncols]`	depth, metres (negative below surface).
`land`	`[Nrows x Ncols]`	land = 1, water = 0.
`amp`	`[Nrows x Ncols x Nconsts]`	amplitude, metres.
`phase`	`[Nrows x Ncols x Nconsts]`	phase, radians.
`name`	`{Nconsts}`	names of the constituents.
`freq`	`[Nconsts]`	frequency of the constituents, rad/day.

2.1.2 Function `finddbvars`

The function needs access to the following files

`rawtidaldb.mat` - Data file created by the `readdb` function.

The function outputs the file `tidaldb.mat` which contains the `db` struct which has the same fields as `rawtidaldb.mat` (unchanged) and two new additional fields:

`bathro`	`[Nrows x Ncols]`	bathymetry variance, metres squared.
`tidalro`	`[Nrows x Ncols]`	tidal variance, metres squared.

The user is referred to Pedersen (2007) for further information on the calculation procedure of these variables.

2.2 Depth measurements

The file containing the time and depth data must be arranged according to one of the predefined “Data types” in the `datastrip` function, see section 3.1. An example of the beginning of a data file is shown below:

```
11.11.2004 00:01,-0.21
11.11.2004 00:11,-0.21
11.11.2004 00:21,-0.21
11.11.2004 00:31,-0.19
...
```

This corresponds to the “Data type” 4 in the `datastrip` function, see section 3.1.

2.3 Release and recapture position

The release and recapture position of the fish must be stated in a plain text file according to the format in the example below

```
rel_long      rel_lat      catch_long  catch_lat  catch_unc
0.6953666667, 50.8697,     0.3833,      50.7667,  20
```

The file is loaded in the `datastrip` function, see section 3.1.

3 Preprocessing

3.1 `datastrip`

Takes as input the raw data file downloaded from an archival tag and strips off the irrelevant parts. The stripped data is saved in a `raw*.mat` file for easy access in MATLAB, where `*` is the tag number e.g. `2255`.

3.1.1 Usage

The function is executed by the call:

```
datastrip(tagno,filets,filerelcatch,unit,typ)
```

3.1.2 Dependencies

The function needs access to the following files

A raw data file downloaded from a DST (see section 2.2 for formats).

3.1.3 Input arguments

Variable	Class	Description
`tagno`	char array	Identifier for the tag, e.g. `tagno='2255'`.
`filets`	char array	Name of the text file containing the depth data, e.g. `filets = '2255PRES.csv'`.
`filerelcatch`	char array	Name of the text file containing the release and recapture information, e.g. `filerelcatch = 'relcatch2255.csv'`.
`unit`	char array	Unit of the depth/pressure record: `unit = 'm'`, units in metres. `unit = 'dbar'`, units in decabar, 1 m = 1.0194 dbar. `unit = 'psi'`, units in pounds per square inch, 1 m = 0.7028 psi.

`typ`	char array	Format type of the data. The types are:
		`typ = '1'` e.g. `2001/03/30 00:01:00,9.698552`
		`typ = '2'` e.g. `24/03/99 00:01,-0.110853`
		`typ = '3'` e.g. `06/10/1999 17:21,1.835654`
		`typ = '4'` e.g. `21.11.2003 00:01,-0.146`
		`typ = '5'` e.g. `11.03.2005 00:01,-1.15,6.152`
		`typ = '6'` e.g. `04/12/05 00:01:00 16.326`
		`typ = '7'` e.g. `16.03.2003 16:10:00,57.2263`
		`typ = '8'` e.g. `06/02/2003 13:50 43.57 7.594`

3.1.4 Output

There is no direct variable output from the function but a data file in the format `*.mat` is stored in the current directory. The file is named after the stated `tagno` e.g. `tagno='2255'` results in an output file named `raw2255.mat`.

3.1.5 See Also

Type `help datastrip` in MATLAB.

3.1.6 Example

```
datastrip('2255','2255PRES.csv','2255relcatch.csv','dbar','1')
```

3.2 tidebehavextr

Extraction of tidal information and behavioural information from a raw dataset. See sections 6.1 and 8.1 in Pedersen (2007) for elaboration on the methods.

3.2.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
tidebehavextr(tagno,tideFL,tideLV,behavFL,behavLV)
```

3.2.2 Dependencies

The function needs access to the following files

```
raw*.mat      - created by datastrip.
tidaldb.mat   - created by finddbvars.
lssinfit.m    - included in this toolbox.
```

3.2.3 Input arguments

Variable	Class	Description
tagno	char array	Identifier for the tag, e.g. tagno='2255'.
tideFL	double array	Length of the fitting interval for the tidal extraction in hours. Default: 10.
tideLV	double array	Limit values for the quality of the tidal extraction fits, tideLV=[rmse rsquare amplitude]. rmse, root mean square error of the fit. rsquare, coefficient of determination of the fit. amplitude, minimum accepted amplitude of the fit, default value = 0.6 m. Default: [0.42 0.85 0.6].
behavFL	double array	Length of the fitting interval for the behaviour extraction in hours. Default: 16.
behavLV	double array	Limit values for the quality of the behaviour extraction fits, behavLV=[rmse rsquare amplitude]. rmse, root mean square error of the fit. rsquare, coefficient of determination of the fit. amplitude, minimum accepted amplitude of the fit, default value = 0.6 m. Default: [0.42 0.85 0.6].

3.2.4 Output

There is no direct variable output from the function but a data file in the format *.mat is stored in the current directory. The file is named after the stated tagno e.g. tagno='2255' results in an output file named tagdata2255.mat.

3.2.5 See Also

Type help tidebehavextr in MATLAB.
See section 3.1 on datastrip.
See section 2.1.2 on finddbvars.
See sections 6.1 and 8.1 of Pedersen (2007) for theory.

3.2.6 Examples

```
tidebehavextr('2255',10,[0.42 0.85 0.6],16,[0.42 0.85 0.6])
tidebehavextr('2255',[],[],13)
tidebehavextr('2255')
```

3.3 datalikelihood

Compute the likelihood for the observation in the entire spatial and temporal domain. See sections 6.3-6.6 in Pedersen (2007) for elaboration on the method.

3.3.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
datalikelihood(tagno,type,iter,recap)
```

3.3.2 Dependencies

The function needs access to the following files

```
tagdata*.mat   - created by tidebehavextr.
tidaldb.mat    - created by finddbvars.
```

3.3.3 Input arguments

Variable	Class	Description
tagno	char array	Identifier for the tag, e.g. tagno='2255'.
type	char array	Specifies a faster but less complex model or a slower but more comprehensive model. type='fast', this type uses a constant depth and tidal roughness to increase computation speed. type='full', this type uses all variance parameters. Default: 'fast'.
iter	char array	Specifies whether to display iterations and elapsed time. type='on', print iterations to the screen. type='off', leaves out print of iterations. Default: 'on'.
recap	char array	Specifies whether to use a recapture position or not e.g. if no recapture position was reported or if one wants to investigate the influence of the recapture position on the geolocation. type='yes', calculates the datalikelihood with the recapture information included. type='no', calculates the datalikelihood without the recapture information. Default: 'yes'.

3.3.4 Output

There is no direct variable output from the function but a data file in the format *.mat is stored in the current directory. The file is named after the stated tagno e.g. tagno='2255' results in an output file named datalikelihood2255.mat saved in the current directory.

3.3.5 See Also

Type help datalikelihood in MATLAB.
See section 3.2 on tidebehavextr.
See section 2.1.2 on finddbvars.
See section 6.6 of Pedersen (2007) for theory.

3.3.6 Examples

```
datalikelihood('2255','fast','off')
datalikelihood('2255',[],'on')
datalikelihood('2255')
```

3.4 `datalikparam`

Called by `datalikelihood`

Script that defines the variance parameters used in the datalikelihood function. The values are estimated from the current data set, from the bathymetry and tidal database and from moored tags. The user is referred to (Pedersen, 2007) sections 6.3-6.5 for more information on the specific parameters.

Should not be called manually by the user.

4 Filtering

4.1 hmmgeolocate

Filter the preprocessed data to obtain a geolocation. The theory supporting this function is found in Pedersen (2007). The result can be visualised in the postprocessing functions `mptrack`, `samptrack`, `avimaker` or `viewdistr`.

4.1.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
hmmgeolocate(tagno,mode,viewres,Duser)
```

4.1.2 Dependencies

The function needs access to the following files

`tagdata*.mat`	- created by `tidebehavextr`.
`datalikelihood*.mat`	- created by `datalikelihood`.
`tidaldb.mat`	- created by `finddbvars`.
`hmmfilter.m`	- included in this toolbox.
`smoothing.m`	- included in this toolbox.
`cmap.mat`	- included in this toolbox.
`fminsearchbnd.m`	- open source function for constrained minimisation, included in this toolbox..

4.1.3 Input arguments

Variable	Class	Description
`tagno`	char array	Identifier for the tag, e.g. `tagno='2255'`.
mode	double array	Specify whether to use one or two behaviour modes. `mode=1`, the geolocation model uses one behaviour mode. `mode=2`, the geolocation model uses two behaviour modes. Default: `1`.
viewres	char array	Display the marginal distributions in succession when geolocation is finished using `viewdistr`. `viewres='on'`, plots distributions to the screen. `viewres='off'`, leaves out plot of distributions. Default: `'on'`.
Duser	double array	Input user defined values of the diffusivities, e.g. `Duser = [10 100]`. This option is useful if the data set is of low quality and therefore not expected to yield reasonable values of the diffusivity by estimation. The input values can e.g. be found from a high quality dataset. Default: If no input is given the diffusivities are estimated from data.

4.1.4 Output

There is no direct variable output from the function but a data file in the format `*.mat` is stored in the current directory. The file is named after the stated `tagno` e.g. `tagno='2255'` results in an output file named `result2255.mat`. This file is used as input for the postprocessing functions, see section 5. When the result file is loaded into Matlab the struct `result` is available which contains the relevant results of the geolocation such as the distribution for each day, the diffusivity estimates, the utilisation distribution (UD) etc. It can either be analysed manually or by using the postprocessing functions.

4.1.5 See Also

Type `help hmmgeolocate` in MATLAB.
See section 5 on how to view the results of the geolocation.
See section 3.2 on `tidebehavextr`.
See section 3.3 on `datalikelihood`.
See section 2.1.2 on `finddbvars`.
See section 4.2 on `hmmfilter`.
See section 4.4 on `smoothing`.

See section 6.1 on `fminsearchbnd`.
See sections 2 and 3 of Pedersen (2007) for theory.

4.1.6 Examples

```
hmmgeolocate('2255',2,'on')
hmmgeolocate('2255',[],'on')
hmmgeolocate(1186',[],'on',[17 150])
hmmgeolocate('2255')
```

4.2 `hmmfilter`

Called by `hmmgeolocate`

Called by `likelihood`

Runs the predictive filter in the hidden Markov model. The marginal probability distributions are estimated based on data prior to the current time step. One filtering sweep results in a set of reconstructed probability distributions and a likelihood value related to the set of parameters used for the computation.

Should not be called manually by the user.

4.3 `likelihood`

Called by `hmmgeolocate`

The function evaluates the likelihood of the movement parameters by performing the forward sweep of the filtering. This function is used to find the maximum likelihood estimate of the movement parameters.

Should not be called manually by the user.

4.4 `smoothing`

Called by `hmmgeolocate`

Smoothes the reconstructed probability distributions from `hmmfilter` resulting in marginal distributions conditioned on the entire set of observations. For this reason the smoothed distributions are the recommended result of a geolocation to be used for further analysis in that it is the one with smallest uncertainty.

Should not be called manually by the user.

5 Postprocessing

5.1 `avimaker`

Create an avi-file based on a geolocation result.

5.1.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
avimaker(tagno,options)
```

5.1.2 Dependencies

The function needs access to the following files

`tagdata*.mat`	- created by `tidebehavextr`.
`result*.mat`	- created by `hmmgeolocate`.
`tidaldb.mat`	- created by `finddbvars`.
`mpt*.mat`	- created by `mptrack`.
`plotting.m`	- included in this toolbox.
`cmap.mat`	- included in this toolbox.
`cmapfancy.mat`	- included in this toolbox.

If one wants to use the `'fancy'` type access to the M_Map package and its high resolution coastline is required. The package can be found at the site www.eos.ubc.ca/ rich/map.html.

5.1.3 Input arguments

Variable	Class	Description
tagno	char array	Identifier for the tag, e.g. tagno='2255'.
options	struct array	A struct containing the set of options that one wishes to apply to the avi-file. The possible fields are given below this, e.g. the field options.rate is referred to as .rate. When a field is not defined its default value will apply.
.rate	double array	Integer-value that defines the sample rate of the avi. Default: 1.
.fps	double array	Number of frames per second. Default: 5.
.range	double array	Defines the range of days to be plotted, e.g. range = 30:80. Default: all days.
.mode	double array	Play animation either 'backward' or 'forward'. Default: 'forward'.
.no	double array	Number of times to loop the animation. Default: 1.
.comp	char array	String indicating the compressor to use. See help avifile for further help on a specific compressor. comp='None', no compressor, must be used on UNIX. other valid values are: 'Indeo3','Indeo5','Cinepak','MSVC','RLE'. Default: 'Cinepak'.
.track	char array	Plot the most probable track on top of the animation for comparison. Valid values are: 'on' or 'off'. Default: 'off'.
.movname	char array	Name of the avi-file. For example movname = 'movie' results in the file movie.avi which is stored in the current directory. Default name is according to the tag number, e.g. tagno = '2255' results in the file geolocation2255.avi.
.zoom	double array	Define the latitude and longitude range to show in the avi-file. For example zoom = [53 55 -2 3] shows the domain from 53 to 55 degrees latitude and from -2 to 3 degrees longitude (unless some of these values are outside the result domain then the limiting value will be used instead). Default: Extends of model domain.
.type	char array	Define the display type of the geolocation. Valid values are: 'plain' which displays the result of the geolocation in its raw form which reduces computation time at the cost of appearance. 'fancy' which displays the result in a "fancy" way with a nice coast line and smoothed distribution. 'bw' which displays the 95% and 50% confidence areas of the distribution instead of the distribution it self. This looks similar to a plot created by the pplotdistr function. Default: 'plain'.
.lock	char array	Can be used only when type = 'fancy' to lock the colour scale of the plot for easier comparison of the distribution on different days. However, when the probability distribution becomes too dispersed the colour scale is shifted to give a more detailed view. Default: 'off'.

5.1.4 Output

There is no direct variable output from the function but an avi-file is stored in the current directory. If not stated otherwise the file is named after the stated tagno e.g. tagno='2255' results in an output file named geolocation2255.avi.

5.1.5 See Also

Type help avimaker in MATLAB.
Type help avifile in MATLAB.
See section 4.1 on hmmgeolocate.
The M_Map package home page www.eos.ubc.ca/ rich/map.html.

5.1.6 Examples

`avimaker('1432')`

`avimaker('2255',op)`, where `op` is given by: `op.rate=2`, `op.fps=7`, `op.range=110:230`, `op.mode='backward'`, `op.track='on'`. The undefined optional fields are set to their default values.

5.2 mptrack

Find the most probable track (i.e. the mode) of the joint posterior distribution of a geolocation result. The MPT is found by application of the Viterbi algorithm (Viterbi, 1967) which is also described in section 3.5 of Pedersen (2007).

5.2.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
[mpt] = mptrack(tagno)
```

5.2.2 Dependencies

The function needs access to the following files

```
tagdata*.mat          - created by tidebehavextr.
datalikelihood*.mat   - created by datalikelihood.
result*.mat           - created by hmmgeolocate.
tidaldb.mat           - created by finddbvars.
```

5.2.3 Input arguments

Variable	Class	Description
tagno	char array	Identifier for the tag, e.g. tagno='2255'.

5.2.4 Output

The function stores the result as a *.mat file in the current directory, named after the stated tagno. If for example tagno = '2255' then the file is named mpt2255.mat. Furthermore the track is output in a struct directly from the function as described in the table below.

Variable	Class	Description
mpt	struct array	Data structure containing latitude and longitude of the MPT either as pixel number of the domain, mpt.lat_pix (or without noise mpt.lat_pix_clean) or as actual latitude and longitude, mpt.lat (or without noise mpt.lat_clean). The struct contains additional information about the track related to its uncertainty or likelihood. The field mpt.P contains the probabilities related to the position at each day and mpt.avgP is the average of this. The field mpt.avgL is a normalised likelihood measure i.e. the value associated with the entire track as found in the Viterbi algorithm for determining the most probable track divided by the number of days of the track. The track can be plotted with plottrack.

5.2.5 See Also

Type help mptrack in MATLAB.
See section 4.1 on hmmgeolocate.
See section 3.3 on datalikelihood.
See section 5.7 on plottrack.
See section 3.5 of Pedersen (2007) for theory.

5.2.6 Examples

```
mptrack('2255')
```

5.3 samptrack

Sample random tracks from the joint posterior distribution of a geolocation result.

5.3.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
[samptracks] = samptrack(result,LIK,no)
```

5.3.2 Dependencies

This function has no dependencies.

5.3.3 Input arguments

Variable	Class	Description
result	struct array	Output from hmmgeolocate found in a result*.mat file.
LIK	struct array	Output from datalikelihood found in a datalikelihood*.mat file.
no	double array	Integer value defining the number of random tracks to sample. Default: 1.

5.3.4 Output

Variable	Class	Description
samptracks	struct array	Data structure containing longitude and latitude of the sampled track(s) either as pixel number of the domain, samptracks.lat_pix (or without noise samptracks.lat_pix_clean) or as actual latitude and longitude, samptracks.lat (or without noise samptracks.lat_clean). The struct contains additional information about the track related to its uncertainty or likelihood. The field samptracks.P contains the probabilities related to the position at each day and samptracks.avgP is the average of this. The field samptracks.L is likelihood of moving to the current position given the previous position and samptracks.avgL is the average value of samptracks.L. The track can be plotted with plottrack.

5.3.5 See Also

Type `help samptrack` in MATLAB.
See section 4.1 on hmmgeolocate.
See section 3.3 on datalikelihood.
See section 5.7 on plottrack.
See section 3.4 of Pedersen (2007) for theory.

5.3.6 Examples

```
samptrack(result,LIK,5)
samptrack(result,LIK)
```

5.4 proboftrack

Determine the probability of each position in the track and its average likelihood measure (useful if you have an old track that did not have these values calculated).

5.4.1 Usage

The function is executed by the call:

```
[track] = proboftrack(track,result,LIK)
```

5.4.2 Dependencies

This function has no dependencies.

5.4.3 Input arguments

Variable	Class	Description
track	struct array	The tag struct that needs to be updated.
result	struct array	Output from hmmgeolocate found in a result*.mat file.
LIK	struct array	Output from datalikelihood found in a datalikelihood*.mat file.

5.4.4 Output

Variable	Class	Description
track	struct array	Data structure equal to the track given as input to the function but with the following struct fields added: track.P which contains the probability of each position in the tag. track.avgP which contains the average of track.P. track.L which contains the likelihood of each position in the tag. track.avgL which contains the average of track.L.

5.4.5 See Also

Type help proboftrack in MATLAB.
See section 5.2 on mptrack.
See section 5.3 on samptrack.
See section 3.3 on datalikelihood.
See section 4.1 on hmmgeolocate.

5.4.6 Example

```
proboftrack(track,result,LIK)
```

5.5 paddtrack

Add a track to a plot created by the function pplottrack. Useful if one wants to show more than one track in the same figure.

5.5.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
paddtrack(track,options)
```

5.5.2 Dependencies

The function needs access to the M_Map package and its high resolution coastline. The package can be found at the site www.eos.ubc.ca/ rich/map.html.

5.5.3 Input arguments

Variable	Class	Description
track	struct array	Data structure created e.g. with mptrack or samptrack.
options	N/A	Plotting options defined similarly to the plot function.

5.5.4 Output

The function has no direct output. It plots a track in the current plot.

5.5.5 See Also

Type help paddtrack in MATLAB.
Type help plot in MATLAB.
See section 5.2 on mptrack.
See section 5.3 on samptrack.
See section 5.11 on pplottrack.

5.5.6 Example

```
paddtrack(mpt,'x:b','linecolor','w')
```

5.6 plottidebehav

Plot the tidal and behaviour classification of a tag as computed by the function tidebehavextr.

5.6.1 Usage

The function is executed by the call:

```
plottidebehav(tagno)
```

5.6.2 Dependencies

The function needs access to the following files tagdata*.mat - created by tidebehavextr.

5.6.3 Input arguments

Variable	Class	Description
tagno	char array	Identifier for the tag, e.g. tagno = '1423'.

5.6.4 Output

The function has no direct output. It plots the tidal and behaviour classification in a single plot.

5.6.5 See Also

Type help plottidebehav in MATLAB.
See section 3.2 on tidebehavextr.

5.6.6 Example

```
plottidebehav('2255')
```

5.7 plottrack

Plot a track structure i.e. either a Most Probable Track or one or more random tracks.

5.7.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
[handle] = plottrack(track,linsty,pltyp,opttrack)
```

5.7.2 Dependencies

This function has no dependencies.

5.7.3 Input arguments

Variable	Class	Description
track	struct array	Plot this primary track. Output from either samptrack or mptrack.
linsty	char array	Specifies line style, markers and color analogous to the plot command. Default: '-b'.
pltyp	char array	Specifies the plot type. pltyp='1d', plots longitude and latitude in seperate windows as a function of time. pltyp='2d', plots longitude and latitude in a two dimensional map. Default: '2d'.
opttrack	struct array	Plot an additional track for comparison with the primary track. Output from either samptrack or mptrack.

5.7.4 Output

Variable	Class	Description
handle	double array	Handle for the plot(s).

5.7.5 See Also

Type help plottrack in MATLAB.
See section 5.3 on samptrack.
See section 5.2 on mptrack.
Type help plot in MATLAB.

5.7.6 Examples

```
plottrack(track,'x-','1d',track2)
plottrack(track,[],[],track2)
plottrack(track)
```

5.8 plotud

Plot the utilisation distribution of the track. The utilisation distribution is simply a sum of the daily distributions in the specified time range and shows the areas where the fish has probably resided.

5.8.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
plotud(result,range,type)
```

5.8.2 Dependencies

The function needs access to the following files

`cmap.mat`	- included in this toolbox (needed for showing plain colours).
`cmapfancy.mat`	- included in this toolbox (needed for showing fancy colours).

5.8.3 Input arguments

Variable	Class	Description
`result`	struct array	Output from `hmmgeolocate` found in a `result*.mat` file.
`range`	char array	Sets the range in terms of days to compute the utilisation distribution from, e.g. `range = 10:30` plots from day 25 to day 130. Default: all days are plotted.
`type`	char array	Define the display type. Valid values are: `'plain'` which displays the result of the geolocation with blue water colour. `log'` which displays the log of the utilisation distribution. `'fancy'` which displays the result in the fancy colormap with white water colour.

5.8.4 Output

There is no direct variable output from the function rather a successive plotting of the marginal distributions on a daily basis to the screen.

5.8.5 See Also

Type `help plotud` in MATLAB.
See section 4.1 on `hmmgeolocate`.

5.8.6 Examples

```
plotud(result)
plotud(result,[],'log')
plotud(result,43:213,'fancy')
```

5.9 pplotdistr

Plot a distribution (e.g. likelihood distribution, smoothed posterior distribution) with a “nice” coast line for use in a journal paper.

5.9.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
pplotdistr(result,field,typ,day,lonrange,latrange,pos)
```

5.9.2 Dependencies

The function needs access to the M_Map package and its high resolution coastline. The package can be found at the site www.eos.ubc.ca/ rich/map.html.

5.9.3 Input arguments

Variable	Class	Description
result	struct array	A result struct array as created by the hmmgeoloc function.
field	char array	Specifies which field in the result struct to plot, e.g. field = 'smooth_plot'. To get a list of the available fields type the name of the result struct (typically just result) in the Matlab command prompt. Only three dimensional arrays of the size of the database domain are valid as inputs.
typ	char array	Specifies the detail level of the coast line according to the definitions of M_Map which are: typ='c', crude detail level. typ='l', low detail level. typ='i', intermediate detail level. typ='h', high detail level. typ='f', full detail level.
day	double array	The number of the day to be plotted.
longrange	double array	The longitude range of the plotted area, e.g. longrange = [-12 4].
latrange	double array	The latitude range of the plotted area, e.g. latrange = [50 55].
pos	double array	Plots a position on top of the distribution given by pos = [lat lon], e.g. pos = [56 6].

5.9.4 Output

The function outputs a figure to the screen.

5.9.5 See Also

Type help pplotdistr in MATLAB.
See section 4.1 on hmmgeolocate.
See section 5.5 on paddtrack.
See section 5.11 on pplottrack.

5.9.6 Examples

```
pplotdistr(result,'smooth','h',100,[-10 8],[48 60],[53 3]);
```

5.10 pplotlik

Plot a likelihood distribution with a “nice” coast line for use in a journal paper.

5.10.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
pplotlik(LIK,result,typ,day,lonrange,latrange)
```

5.10.2 Dependencies

The function needs access to the M_Map package and its high resolution coastline. The package can be found at the site www.eos.ubc.ca/ rich/map.html.

5.10.3 Input arguments

Variable	Class	Description
LIK	struct array	A datalikelihood struct array as created by the datalikelihood function.
result	struct array	A result struct array as created by the hmmgeoloc function.
typ	char array	Specifies the detail level of the coast line according to the definitions of M_Map which are: typ='c', crude detail level. typ='l', low detail level. typ='i', intermediate detail level. typ='h', high detail level. typ='f', full detail level.
day	double array	The number of the day to be plotted.
longrange	double array	The longitude range of the plotted area, e.g. longrange = [-12 4]
latrange	double array	The latitude range of the plotted area, e.g. latrange = [50 55]

5.10.4 Output

The function outputs a figure to the screen.

5.10.5 See Also

Type help pplotlik in MATLAB.
See section 3.3 on datalikelihood.
See section 5.5 on paddtrack.
See section 5.11 on pplottrack.

5.10.6 Examples

```
pplotlik(LIK,result,'h',100,[-10 8],[48 60]);
```

5.11 pplottrack

Plot a track with a “nice” coast line for use in a journal paper.

5.11.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
pplottrack(track,typ,lonrange,latrange,dayrange,opttrack,db,landcolour)
```

5.11.2 Dependencies

The function needs access to the M_Map package and its high resolution coastline. The package can be found at the site www.eos.ubc.ca/ rich/map.html.

5.11.3 Input arguments

Variable	Class	Description
track	struct array	A track struct array as created by e.g. the mptrack function.
typ	char array	Specifies the detail level of the coast line according to the definitions of M_Map which are: typ='c', crude detail level. typ='l', low detail level. typ='i', intermediate detail level. typ='h', high detail level. typ='f', full detail level. Default: typ='l'.
longrange	double array	The longitude range of the plotted area, e.g. longrange = [-12 4]. Default: longrange = [-10 8].
latrange	double array	The latitude range of the plotted area, e.g. latrange = [50 55]. Default: latrange = [48 60].
dayrange	double array	The range of days to plot, e.g. dayrange = 1:50. Default: All days are plotted.
opttrack	struct array	Add another track to the plot. Useful for comparing two tracks. Default: No optional track is plotted.
db	struct array	Specify the bathymetry database to use. The database must be of the same format as the North Sea database which can be seen in the db struct array when loaded from the file tidaldb.mat. Default: The bathymetry of the North Sea is used.
landcolour	double/char	Specify the colour of the land areas. The colour must be specified according to Matlab standards i.e. either as an rgb code e.g. [0 0.5 0] for dark green or as some of the predefined string specifiers e.g. 'w' for white. Default: [0.5 0.5 0.5].

5.11.4 Output

The function outputs a figure to the screen.

5.11.5 See Also

Type help pplottrack in MATLAB.
See section 5.2 on mptrack.
See section 5.3 on samptrack.
See section 5.5 on paddtrack.
See section 5.9 on pplotdistr.

5.11.6 Example

```
pplotdistr(result,'smooth','h',100,[-10 8],[48 60]);
```

5.12 probofvisit

Estimate the probability that the fish visited some rectangular area(s) during its entire time at liberty. This area could e.g. be a marine protected area where fishing is prohibited.

5.12.1 Usage

The function is executed by the call:

```
[p] = probofvisit(tagno,area,dayrange)
```

5.12.2 Dependencies

The function needs access to the following files

`result*.mat`	- created by `hmmgeolocate`.
`datalikelihood*.mat`	- created by `datalikelihood`.
`samptrack.m`	- Matlab function, included in this toolbox.

5.12.3 Input arguments

Variable	Class	Description
`tagno`	char array	Identifier for the tag, e.g. `tagno='2255'`.
`area`	double array	Specify the rectangular area(s) in longitude and latitude coordinates e.g. as `area = [longmin longmax latmin latmax]`.
`dayrange`	double array	Specify the days to base the calculation on e.g. as `dayrange = 21:54`. Default: All days are used.

5.12.4 Output

The function plots the area along with a coast line.

Variable	Class	Description
`p`	double array	The estimated probability that the fish visited the defined area within the define time period (day range).

5.12.5 See Also

Type `help probofvisit` in MATLAB.
See section 4.1 on `hmmgeolocate`.
See section 3.3 on `datalikelihood`.
See section 5.3 on `samptrack`.

5.12.6 Example

```
probofvisit('2255',[1 2 55 56; 1 3 54.5 55], 12:46)
```

5.13 viewdistr

Function for convenient plotting of a result of a geolocation. The function is used in the function hmmgeolocate to view the newly created result.

5.13.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
viewdistr(distr,fps,range,mode,no,type,land)
```

5.13.2 Dependencies

The function needs access to the following files

```
cmap.mat        - included in this toolbox (needed for showing plain colours).
cmapfancy.mat   - included in this toolbox (needed for showing fancy colours).
```

5.13.3 Input arguments

Variable	Class	Description
distr	double array	Subvariable in the result structure found in an output from hmmgeolocate.
fps	double array	Integer value defining the number of frames per second to show, i.e. the speed of the movie, e.g. fps = 1 shows one frame per second. Default: maximum speed allowed by the computer system.
range	double array	Sets the range in terms of days to plot, e.g. range = 10:30 plots from day 10 to day 30. Default: all days are plotted.
mode	char array	Specifies whether to play the animation forward or backward in time. Valid values are: 'forward','backward'. Default: 'forward'.
no	double array	Integer value defining the number of time to play the animation. Default: 1.
type	double array	Define the display type. Valid values are: 'plain' which displays the result of the geolocation in its raw form which reduces computation time at the cost of appearance. 'fancy' which displays the result in the fancy colormap with white water colour. 'fancylock' is similar to 'fancy' but locks the colorbar for easier comparison of distributions on different days. Default: 'plain'
land	double array	Is required if type = 'fancy' or type = 'fancylock' is set. A land array e.g. db.land or result.land which contains the land indicators for the domain. Default: No land. (note that type = 'plain' uses a "built in" land indication).

5.13.4 Output

There is no direct variable output from the function rather a successive plotting of the marginal distributions on a daily basis to the screen.

5.13.5 See Also

Type help viewdistr in MATLAB.
See section 4.1 on hmmgeolocate.

5.13.6 Examples

```
viewdistr(result.phi_plot,10,50:3:200,'backward',2)
viewdistr(result.smooth_plot,[],[],'backward')
viewdistr(result.smooth,[],[],'backward','fancy',result.land)
```

6 Background functions

6.1 fminsearchbnd

Called by hmmgeolocate

Minimising function created by John D'Errico. Uses Matlab's built in function fminsearch to perform a constrained minimisation. This function is used in hmmgeolocate to find the minimum of the likelihood function for the movement parameters constrained by the maximum swimming speed of the fish.

The function can be downloaded at the open source function archive at Math Works.

6.2 gausskern

Create a 2D Gauss kernel as created by the function fspecial in the Image Processing toolbox.

6.2.1 Usage

Arguments in *italic* are optional. When omitted their default value is used. The function is executed by the call:

```
[kern] = gausskern(siz,sigma,muadv)
```

6.2.2 Dependencies

The function needs access to the following files

normalise.m - included in this toolbox.

6.2.3 Input arguments

Variable	Class	Description
siz	double array	Size of the kernel, siz × siz. Must be a positive integer.
sigma	double array	Standard deviation of the kernel. Unit is cell width. Must be a positive number.
muadv	double array	Advection of the kernel. Unit of the input is cell width. Default: 0.

6.2.4 Output

Variable	Class	Description
kern	double array	The Gauss kernel as a 2D matrix of size: siz × siz.

6.2.5 See Also

Type help gausskern in MATLAB.

6.2.6 Examples

```
kern = gausskern(3,0.5)
kern = gausskern(9,2,1.5)
```

6.3 gausspdf

Find the value of the probability density function (pdf) of a (possibly multivariate) Gaussian distribution.

6.3.1 Usage

The function is executed by the call:

```
[pdfval] = gausspdf(x,mu,invcholSigma,const)
```

6.3.2 Dependencies

This function has no dependencies.

6.3.3 Input arguments

Variable	Class	Description
x	double array	Independent variable to find pdf value for.
mu	double array	Mean of the Gaussian distribution.
invcholSigma	double array	The inverse of the cholesky factorised covariance matrix.
const	double array	The constant in pdf of a Gaussian distribution, in the 1D standard case this is equal to $1/\sqrt{2\pi}$.

6.3.4 Output

Variable	Class	Description
pdfval	double array	Value of the pdf related to x.

6.3.5 See Also

Type `help gausspdf` in MATLAB.

6.4 mvgausspdf

Find the value of the probability density function (pdf) of a (possibly multivariate) Gaussian distribution.

6.4.1 Usage

The function is executed by the call:

```
[pdfval] = mvgausspdf(x,mu,sigma)
```

6.4.2 Dependencies

This function has no dependencies.

6.4.3 Input arguments

Variable	Class	Description
x	double array	Independent variable to find pdf value for.
mu	double array	Mean of the Gaussian distribution.
sigma	double array	The covariance matrix.

6.4.4 Output

Variable	Class	Description
pdfval	double array	Value of the pdf related to x.

6.4.5 See Also

Type `help mvgausspdf` in MATLAB.

6.5 plotting

Called by avimaker

The function plots the result of a geolocation. The function is used by the avimaker function to generate each frame in the animation.

Should not be called manually by the user.

6.6 plottingfancy

Called by avimaker

The function plots the result of a geolocation. The function is used by the avimaker function to generate each frame in the animation.

Should not be called manually by the user.

6.7 plottingbw

Called by avimaker

The function plots the result of a geolocation. The function is used by the avimaker function to generate each frame in the animation.

Should not be called manually by the user.

6.8 lssinfit

Called by tidebehavextr

Fits a sine wave function to the input data using a least squares regression. The function returns the root mean square error, the R^2 and the amplitude of the fit. The function has an optional outlier removal feature and, if used, outputs the number of outliers removed in the process.

Should not be called manually by the user.

6.9 makeplotstandard

Called by pplotdistr

Function that creates a two-dimensional cumulated distribution function of e.g. a result of a geolocation. This is convenient when one needs to plot e.g. a 50% confidence area of a distribution.

6.10 mapmatrix

Called by mptrack

Called by samptrack

The function creates a so called mapmatrix which is used for conversion between array indices and lat/long coordinates. It is used for calculating the latitude and longitudes of estimated tracks.

Should not be called manually by the user.

6.11 maptopix

Called by mptrack

Called by samptrack

The function uses a mapmatrix created by the mapmatrix function to convert lat/long coordinates to array indices.

Should not be called manually by the user.

6.12 normalise

Called by too many of the functions in the toolbox to list all here

The function normalises a distribution e.g. the result of a geolocation, to sum to one.

Should not be called manually by the user.

6.13 normcdf

Called by datalikelihood

Compute the value of the cumulated density function of a Gaussian distributed number with mean mu and standard deviation sigma.

Should not be called manually by the user.

6.14 pixtomap

Called by mptrack

Called by samptrack

The function uses a mapmatrix created by the mapmatrix function to convert array indices to lat/long coordinates.

Should not be called manually by the user.

6.15 deglong

Called by readdb

Called by mptrack

Called by samptrack

The function returns the geographical distance in km of one degree of longitude at the latitude taken as input. If the function is given zero as input the it returns the geographical distance in km of one degree of latitude (or one degree of longitude at zero degrees latitude).

Should not be called manually by the user.

6.16 m_surf

Called by plottingfancy

The function works as m_contour which is included in the M_Map package except that it plots a surface instead of the contour lines.

Should not be called manually by the user.

6.17 flipdb

Called by tidebehavextr

Called by datalikelihood

The function is used to flip the database on either the latitudinal or longitudinal axis.

Should not be called manually by the user.

References

Pedersen MW (2007) Hidden markov models for geolocation of fish Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU IMM publication.

Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform.Theory* IT-13:260–269.