

# Package ‘activityGCMM’

October 12, 2022

**Title** Circular Mixed Effect Mixture Models of Animal Activity Patterns

**Version** 1.1.1

**Description** Bayesian parametric generalized circular mixed effect mixture models (GCMMs) for estimating animal activity patterns from camera trap data and other nested data structures using 'JAGS', including automatic Bayesian k-cluster selection and random circular intercepts for nested data. The GCMM function automatically selects the number of components for the mixture model (supporting up to 4 mixture components) based on a Bayesian linear finite normal mixture model and fits a Bayesian parametric circular mixed effect mixture model with one or two random effects as random circular intercepts with a von Mises or wrapped Cauchy distribution. Provides graphs of the combined mixture model or separate mixture components. Functionality is provided to allow quantitative comparisons between model parameters. See Campbell et al. (in press) It's time to expand our analyses of animal activity; Campbell et al. (in press) Temporal and microspatial niche partitioning; Campbell et al. (in press) A novel approach to comparing animal activity patterns. News, updates, and tutorials will be available on [www.atlasgoldenwolf.org/stats](http://www.atlasgoldenwolf.org/stats) and [www.github.com/LizADCampbell](http://www.github.com/LizADCampbell).

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 3.00)

**Imports** mclust, runjags, circular, overlap, stats, graphics,  
grDevices, utils,

**NeedsCompilation** no

**Author** Liz AD Campbell [aut, cre] (<<https://orcid.org/0000-0002-8302-7430>>)

**Maintainer** Liz AD Campbell <liz.campbell@zoo.ox.ac.uk>

**Repository** CRAN

**Date/Publication** 2021-06-14 19:20:02 UTC

## R topics documented:

activityHPD . . . . . 3

activityHPDmean . . . . .	3
APDatPeak . . . . .	4
APDpointplot . . . . .	5
calcAPD . . . . .	5
calcprop . . . . .	6
circaxis . . . . .	7
circplotHPD . . . . .	7
circplotmeans . . . . .	8
circplotREs . . . . .	9
combineMCMC . . . . .	9
comboplot . . . . .	10
compareGCMM . . . . .	11
compareGCMMfit . . . . .	12
componentsplot . . . . .	13
convertRad . . . . .	14
exampleGCMM . . . . .	14
extractparam . . . . .	15
GCMM . . . . .	15
GCMMpdens . . . . .	18
GCMMpeaksplot . . . . .	18
GCMMppc . . . . .	19
GCMMprob . . . . .	19
GCMMsims . . . . .	20
GCMMsimsparams . . . . .	20
HDI . . . . .	21
HPDoverlap . . . . .	21
humanssample . . . . .	22
mixtureplot . . . . .	23
mode . . . . .	24
multiplot . . . . .	24
PDS . . . . .	25
peaksPDplot . . . . .	26
plotactivityHPD . . . . .	26
plotGCMMsamples . . . . .	27
plotREs . . . . .	28
posteriorhistplot . . . . .	29
progress . . . . .	29
redfoxsample . . . . .	30
samplerows . . . . .	30
sumCircResids . . . . .	31
updateGCMM . . . . .	32
xaxis . . . . .	33
yMax . . . . .	34

---

activityHPD	<i>Activity highest posterior density interval estimates (activityHPD)</i>
-------------	--

---

**Description**

Calculate activity highest posterior density interval (activityHPD), HPD duration, number of activity peaks, peak activity times and maximum activity probability density for a given probability density mass

**Usage**

```
activityHPD(model, sample = 1000, prob = 0.5, scale = "2pi")
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
sample	Number of posterior samples for which to calculate 95% HDIs
prob	Value to use for probability density mass; default=0.50
scale	Scale of the data for plotting, either "2pi" for 0,2pi or "pi" for -pi,pi; default="2pi"

**Value**

Returns object of class GCMMestimate with list including estimated peak activity times, maximum activity probability density, HPD interval, HPD duration, and number of activity peaks

---

activityHPDmean	<i>Activity highest posterior density interval from mean activity curve</i>
-----------------	---

---

**Description**

Estimates activity highest posterior density interval (HPD), HPD duration, number of activity peaks, peak activity times and maximum activity probability density for a given probability density mass from the GCMM activity curve predicted by the GCMM parameter posterior distribution means

**Usage**

```
activityHPDmean(
  model,
  prob = 0.5,
  scale = "2pi",
  silent = FALSE,
  plot = FALSE,
  col = "cyan4",
  ymax = "NULL"
)
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
prob	Value to use for probability density mass; default=0.50
scale	Scale of the data, either "2pi" for 0,2pi or "pi" for -pi,pi; default="2pi"
silent	Logical vector for whether to print output and plot; default=FALSE
plot	Logical argument for whether to plot activity curve with HPD; default=FALSE
col	Colour for HPD on plot
ymax	Value to use as upper limit of y axis on activity curve plot

**Value**

Returns object of class `GCMMestimate` with list with estimated peak activity times, maximum activity probability density, HPD interval, HPD duration, and number of activity peaks.

---

APDatPeak

*Activity Probability Density at Peak Time of Another*


---

**Description**

Calculates activity probability density from one GCMM model at the peak activity time of a second GCMM model

**Usage**

```
APDatPeak(model1, model2, sample = 1000, HDIprob = 0.95)
```

**Arguments**

model1	Object of class GCMM with output from <a href="#">GCMM</a> function
model2	Object of class GCMM with output from <a href="#">GCMM</a> function
sample	Number of posterior samples from which to build the HDIs
HDIprob	Value for probability mass to use for HDI; default=95%

**Value**

Returns matrix with the mean and HDI of activity probability density estimated from both GCMM models at the peak activity time of the other. Posterior distributions of activity probability density and peak activity times for both GCMM models are also saved.

---

APDpointplot	<i>Activity Probability Density Point Plot</i>
--------------	--

---

**Description**

Plot of GCMM activity with predicted activity probability density at particular time points

**Usage**

```
APDpointplot(
  model,
  time,
  ymax = "NULL",
  scale = "2pi",
  cex = 2.5,
  col = "lightseagreen",
  axisunits = "radians"
)
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
time	Time point for which to plot predicted activity probability density
ymax	Upper limit of y axis
scale	Scale for which to plot the activity curve, either "2pi" for 0,2pi or "pi" for -pi, pi; default="2pi"
cex	Size of plotted point; default=2.5
col	Colour of plotted point
axisunits	Units for xaxis

**Value**

No return value; prints plot of activity curve with activity probability density prediction at the specified time point and returns dataframe of time points and activity probability density

---

calcAPD	<i>Calculate activity probability density</i>
---------	---

---

**Description**

Support function for calculating activity probability density at a specified time from an activity curve

**Usage**

```
calcAPD(x, curve)
```

**Arguments**

x	Value in radians for which to predict probability density
curve	Temporal data to predict density from

**Value**

Returns activity probability density value

---

calcprop	<i>Calculate proportions of circular variable within an interval</i>
----------	--

---

**Description**

Support function that calculates proportion of a vector of circular data within an interval

**Usage**

```
calcprop(x, p1, p2)
```

**Arguments**

x	Vector of data
p1	Number identifying start of interval
p2	Number identifying end of interval

**Value**

Returns proportion of the vector within the interval

---

circaxis	<i>Axis labels for circular temporal data plots</i>
----------	---

---

**Description**

Support function for axis labels for circular plots

**Usage**

```
circaxis(axisunits = c("radians", "time", "sun", "none"))
```

**Arguments**

axisunits	Scale to use for the xaxis, either "radians", "time", "sun", or "none"; default="radians"
-----------	---

**Value**

Prints axis

---

circplotHPD	<i>Circular plot of activity HPD intervals</i>
-------------	--

---

**Description**

Circular plot of activity HPD intervals from GCMM activity curves

**Usage**

```
circplotHPD(
  models,
  prob = 0.5,
  col = c("cyan3", "orchid", "deeppink", "dodgerblue"),
  axisunits = c("radians", "sun", "time")
)
```

**Arguments**

models	List of one or more objects of class GCMM containing output from the <a href="#">GCMM</a> function
prob	Value for activityHPD probability density mass; default=0.5 (i.e. 50% HPD)
col	Vector of colours to use in the plot
axisunits	Units to be used for the axis, either "radians", "sun", "time", or "none"

**Value**

No return value; prints plot

**Examples**

```
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians,
  RE1=redfoxsample$SamplingPeriod, family="vonmises", autorun=FALSE)
HumanActivityGCMM<-GCMM(data=humanssample$Radians, RE1=humanssample$SamplingPeriod,
  family="vonmises", autorun=FALSE)
circlplotHPD(models=list(FoxActivityGCMM,HumanActivityGCMM))
```

---

 circlplotmeans

*Circular plot of GCMM means*


---

**Description**

Circular plot of GCMM means (circular intercepts)

**Usage**

```
circlplotmeans(
  models,
  col = c("cyan3", "orchid", "deeppink", "dodgerblue"),
  axisunits = c("radians", "sun", "time")
)
```

**Arguments**

models	List of one or more objects of class GCMM containing output from the <a href="#">GCMM</a> function
col	Vector of colours to use in the plot
axisunits	Units to be used for the axis, either "radians", "sun", or "time"

**Value**

Prints plot

**Examples**

```
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians,
  RE1=redfoxsample$SamplingPeriod, family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1, n.chains=2)
HumanActivityGCMM<-GCMM(data=humanssample$Radians, RE1=humanssample$SamplingPeriod,
  family="vonmises", autorun=FALSE, adapt=0, sample=300, burnin=300, thin=1, n.chains=2)
circlplotmeans(models=list(FoxActivityGCMM,HumanActivityGCMM))
```



---

 circplotREs

*Random Effects Circular Plot*


---

**Description**

Circular plot of GCMM random intercepts and 95% HDI

**Usage**

```
circplotREs(
  model,
  RE1 = TRUE,
  RE2 = FALSE,
  axisunits = c("radians", "sun", "time", "solar", "none")
)
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
RE1	Logical vector for whether to plot GCMM activity curve with random intercepts from RE1; default=TRUE
RE2	Logical vector for whether to plot GCMM activity curve with random intercepts from RE2; default=FALSE
axisunits	Units for x axis, either "radians", "time", "solar", "sun", or "none"; default="radians"

**Value**

No return value; prints circle plot of GCMM random intercepts and 95% HDI

**Examples**

```
FoxGCMMREs<-GCMM(data=redfoxsample$Radians, RE1=redfoxsample$SamplingPeriod,
  saveREs=TRUE, scale=c("2pi"), family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1)
circplotREs(FoxGCMMREs, axisunits="sun")
```

---

 combineMCMC

*Combine MCMC chains for posterior simulations*


---

**Description**

Support function that extracts MCMC chains for creating posterior simulations of activity curves

**Usage**

```
combineMCMC(model)
```

**Arguments**

model                    Object of class GCMM containing output from GCMM function

**Value**

Returns a list of MCMC chains

---

comboplot	<i>GCMM Combined Plot</i>
-----------	---------------------------

---

**Description**

Combined plot of estimated activity curve from mixture model and separate mixture components

**Usage**

```
comboplot(
  model,
  rug = FALSE,
  ruglwd = 2,
  ltyc = 2,
  ltym = 1,
  lwdc = 3,
  lwdm = 3,
  colc = c("grey40", "grey55", "grey70", "grey85"),
  colm = "black",
  scale = "NULL",
  ymax = "NULL",
  axisunits = c("radians", "sun", "solar", "time", "none"),
  xlines = TRUE
)
```

**Arguments**

model                    Model output from [GCMM](#) function, object of class GCMM

rug                        Logical argument for whether to plot a rug of the raw values. Plotting the rug for the separate components requires that saveclustIDs=TRUE when running GCMM. default=FALSE

ruglwd                    Line width for rug plot

ltyc                      Line type for activity curves for components

ltym                      Line type for activity curves for mixture

lwdc                      Line width for activity curve lines for components

lwdm                      Line width for activity curve lines from mixture

colc                      Character vector for colours for the activity curve lines and rug plot for components; must be of equal length to the number of components

colm	Character vector for colour of activity curve line from mixture model
scale	Scale for the plot, either "pi" (-pi, pi) or "2pi" (0, 2pi); default is that recommended by the GCMM function
ymax	Value for upper limit of y-axis
axisunits	Scale to use for the xaxis, either "radians", "time", "solar", "sun", or "none"; default="radians"
xlines	Whether to include lines on the graph for the x axis labels; default=TRUE

**Value**

Prints combined plot of estimated activity curve from mixture model and separate mixture components

**Examples**

```
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians,
  RE1=redfoxsample$SamplingPeriod, family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1,n.chains=2)
comboplot(FoxActivityGCMM)
```

---

 compareGCMM

---

*Compare GCMM parameters or estimates*


---

**Description**

Compare two posterior distributions of GCMM parameters or estimates

**Usage**

```
compareGCMM(model1, p1, model2 = "NULL", p2, sample = 1000, plot = TRUE)
```

**Arguments**

model1	Object either of class GCMM with output from <a href="#">GCMM</a> function or of class GCMMestimate with output from other activityGCMM functions
p1	Name of first parameter to compare, from model1
model2	Object of either class GCMM, GCMMestimate, or a vector or single value for which to compare with model 1; if left blank, it is assumed that both parameters from arguments p1 and p2 are from model1
p2	Second parameter to compare, either name of a parameter, estimate, or a vector of values
sample	Number of posterior samples from which to build HDI; default=1000
plot	Logical argument for whether to plot histograms of p1, p2, and the difference between them; default=TRUE

**Value**

Returns object of class `GCMMestimate` containing a list, including PD containing the posterior distributions of  $p_1$ ,  $p_2$  and the difference between them, and summary information (HDIs and PDS)

---

compareGCMMfit	<i>Compare fit of GCMM models based on circular residuals</i>
----------------	---

---

**Description**

Compare fit of GCMM models by comparing posterior distributions of the summed circular residuals

**Usage**

```
compareGCMMfit(model1, model2, sample = 10000)
```

**Arguments**

model1	Object of class <code>GCMM</code> with output from <code>GCMM</code> function to compare with model2; residuals must be saved when running the <code>GCMM</code> function using <code>saveResids=TRUE</code>
model2	Object of class <code>GCMM</code> with output from <code>GCMM</code> function to compare with model1; residuals must be saved when running the <code>GCMM</code> function using <code>saveResids=TRUE</code>
sample	Number of posterior samples; default=10000

**Value**

Returns object of class `GCMMestimate` with list of output

**Examples**

```
FoxVMGCMM<-GCMM(data=redfoxsample$Radians, RE1=redfoxsample$CameraTrapID, family="vonmises",
  saveResids=TRUE, scale=c("2pi"), autorun=FALSE, adapt=0, sample=1000, burnin=500, thin=1)
FoxWCGCMM<-GCMM(data=redfoxsample$Radians, RE1=redfoxsample$CameraTrapID, family="wrappedcauchy",
  saveResids=TRUE, scale=c("2pi"), autorun=FALSE, adapt=0, sample=1000, burnin=500, thin=1)
FoxModelCompare<-compareGCMMfit(FoxVMGCMM, FoxWCGCMM)
```

---

componentsplot      *GCMM Components Plot*

---

### Description

Plot of activity curves for the separate components in the circular mixture model

### Usage

```
componentsplot(
  model,
  rug = FALSE,
  ruglwd = 2,
  lwd = 3,
  col = c("black", "grey40", "grey60", "grey80"),
  scale = "NULL",
  ymax = "NULL",
  lty = 1,
  axisunits = c("radians", "sun", "solar", "time", "none"),
  xlines = TRUE
)
```

### Arguments

model	Model output from <a href="#">GCMM</a> function, object of class GCMM
rug	Logical argument for whether to plot a rug of the raw values. Plotting the rug for the separate components requires that saveclustIDs=TRUE when running GCMM. default=FALSE
ruglwd	Line width for rug plot
lwd	Line width for activity curve lines
col	Character vector for colours for the activity curve lines and rug plot; must be of equal length to the number of components
scale	Scale for the plot, either "pi" (-pi, pi) or "2pi" (0, 2pi); default is that recommended by the GCMM function
ymax	Value for upper limit of y-axis
lty	Line type for activity curve
axisunits	Scale to use for the xaxis, either "radians", "time", "solar", "sun", or "none"; default="radians"
xlines	Whether to include lines on the graph for the x axis labels; default=TRUE

### Value

Plot of the separate components of the circular mixture model

**Examples**

```
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians,
  RE1=redfoxsample$SamplingPeriod, family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1,n.chains=2)
componentsplot(FoxActivityGCMM)
```

---

 convertRad

*Convert Radians Scale*


---

**Description**

Converts scale of observations in radians to (0,2pi) or (-pi,pi) or hours

**Usage**

```
convertRad(x, to)
```

**Arguments**

x	Vector of observations in radians
to	Character vector to specify conversion, either "2pi" for (0, 2pi), "pi" for (-pi, pi), or "hours" for hours

**Value**

Returns vector of observations on desired scale

**Examples**

```
Rad2<-convertRad(redfoxsample$Radians, to="2pi")
```

---

 exampleGCMM

*Executable example of GCMM function*


---

**Description**

Example of applying generalized circular mixed effect mixture model with activityGCMM using data included in the package

**Usage**

```
exampleGCMM()
```

**Value**

Prints message with example of GCMM function using data included in the package

**Examples**

```
{ exampleGCMM() }
```

---

 extractparam

*Extract parameters for posterior simulations*


---

**Description**

Support function that extracts parameter estimates for creating posterior simulations of activity curves

**Usage**

```
extractparam(model, x)
```

**Arguments**

model	Object of class GCMM containing output from GCMM function
x	Name of parameter to be extracted

**Value**

Returns posterior samples of the parameter

---

 GCMM

*Generalized circular mixed effect mixture (GCMM) model*


---

**Description**

Bayesian parametric generalized circular mixed effect mixture models (GCMM) for estimating animal activity curves from camera traps and other nested data structures using JAGS. Data distributions currently supported include von Mises and wrapped Cauchy, with one or two random effects fit as random circular intercepts. The GCMM function automatically selects the number of components for the mixture model (supporting up to 4 mixture components) and runs the model in 'JAGS' through R. The number of clusters can also be manually selected. The function returns the model summary and the activity curve estimated from the circular mixture model, with additional information from the analysis provided in the output as a list of class GCMM.

Package: activityGCMM Version: 1.0.1 Date: 2021-06-06 Author: Liz AD Campbell

**Usage**

```

GCMM(
  data,
  RE1,
  RE2 = NULL,
  scale = "2pi",
  kmax = 15,
  family = c("vonmises", "wrappedcauchy"),
  autorun = TRUE,
  minESS = 5000,
  maxrep = 5,
  thin = 2,
  burnin = 5000,
  sample = 5000,
  adapt = 1000,
  n.chains = 3,
  saveREs = FALSE,
  saveResids = FALSE,
  saveclustIDs = FALSE,
  saveYExp = FALSE,
  saveJAGS = TRUE,
  Nclust = "NULL",
  clustmeans = NULL
)

```

**Arguments**

data	Vector of observations in radians (0 to 2pi)
RE1	Vector identifying random effect for observations (e.g. camera trap ID)
RE2	Vector identifying second random effect for observations (e.g. study site, year, season, sampling period)
scale	Scale of observations, either 0 to 2pi ("2pi") or -pi to pi ("pi")
kmax	Maximum number to test for vonmises kappa parameter; default=15
family	Probability distribution, either "vonmises" or "wrappedcauchy"
autorun	Logical argument for whether to automatically extend the analyses to achieve MCMC chain convergence and a specified minimum effective sample size (ESS) for all parameters; default=TRUE
minESS	Minimum effective sample size (ESS) from the posterior distribution desired for all parameters; default=5000, though a minimum ESS of 10000 is recommended
maxrep	Maximum number of extensions of the analysis if autorun=TRUE; default=5
thin	Thinning rate for MCMC chains, i.e. how many samples are saved. For longer models, thin can be increased to reduce computer memory requirements
burnin	If autojags=FALSE, the burnin for the MCMC chains which are not saved; default=5000



sample	If autojags=FALSE, the number of MCMC samples per chain (which is multiplied by thin); default=10000
adapt	adaptation to use for MCMC chains; default=1000
n.chains	number of MCMC chains; default=3
saveREs	Whether random intercepts are saved in output; recommended to save only one of saveREs, saveResids or saveYExp at one time due to memory limitations
saveResids	Whether model residuals are saved in output; recommended to save only one of saveREs, saveResids or saveYExp at one time due to memory limitations
saveclustIDs	Whether to save component cluster identification for the data points; default=FALSE
saveYExp	Whether expected Y values based on model are saved in output; recommended to save only one of saveREs, saveResids or saveYExp at one time due to memory limitations
saveJAGS	Logical argument of whether to save runjags output; default=FALSE
Nclust	Number of components for mixture models; if not provided, the function will estimate the number of clusters; if provided, values must be provided for clustmeans
clustmeans	A vector equal in length to Nclust of the potential means for each component in the mixture models

### Details

The number of clusters is automatically selected based on a Bayesian linear finite normal mixture model via the `mclust` package. The Bayesian parametric GCMM is fit using 'JAGS' through R using the `runjags` package.

### Value

Returns object of class GCMM which is a list containing analysis results and details. A plot of the estimated activity curve from the mixed effect mixture model is printed.

output GCMM model output summary

GCMMmixture Vectors of simulated values from mixture model

GCMMcomponents Vectors of simulated values from each component in the mixture model

runjags GCMM model output from JAGS of class runjags from runjags package; see [run.jags](#)

### Author(s)

Liz AD Campbell

### Examples

```
data(redfoxsample)
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians, RE1=redfoxsample$SamplingPeriod,
  scale=c("2pi"), family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1,n.chains=2 )
```

---

GCMMpdens *Predict activity probability density at a time point*

---

### Description

Calculates predicted activity probability density from GCMM model at a specific time point

### Usage

```
GCMMpdens(model, timepoint, HDI = TRUE, sample = 1000, scale = "2pi")
```

### Arguments

model	Object of class GCMM with output from <a href="#">GCMM</a> function
timepoint	Time point for which to predict activity probability density
HDI	Logical argument for whether to calculate 95% HDI; default=TRUE
sample	Number of posterior samples from which to build HDI; default=1000
scale	Scale of the data, either "2pi" for 0,2pi or "pi" for -pi,pi; default="2pi"

### Value

Returns numeric vector of estimated probability density if HDI=FALSE or posterior probability distribution if HDI=TRUE

---

GCMMpeakplot *Plot estimated time of activity peaks*

---

### Description

Plot mean GCMM activity curve with peak activity time from [activityHPD](#) function

### Usage

```
GCMMpeakplot(x, scale = "2pi", ymax = "NULL")
```

### Arguments

x	Object of class GCMMestimate with output from <a href="#">activityHPD</a> function
scale	Scale for plotting the activity curve, either "2pi" for 0,2pi or "pi" for -pi,pi; default="2pi"
ymax	Value for upper limit of y axis

### Value

No return value; prints plot activity curve and peak activity time

---

GCMMppc *Posterior predictive check of GCMM model*

---

### Description

Conduct posterior predictive check (PPC) by simulating data from fitted GCMM model and plotting against observed data

### Usage

```
GCMMppc(model, YExp = NULL, clustID = NULL)
```

### Arguments

model	Object of class GCMM with output from <a href="#">GCMM</a> function; if YExp and clustIDs are not provided as vectors, the GCMM model must contain this information using the arguments saveYExp=TRUE and saveclustIDs=TRUE
YExp	Vector of YExp values from GCMM function; see also <a href="#">GCMM</a>
clustID	Vector of clustID values from GCMM function; see also <a href="#">GCMM</a>

### Value

Returns vector of simulated values and prints plot of simulated and raw values

### Examples

```
FoxGCMMPPC<-GCMM(data=redfoxsample$Radians, RE1=redfoxsample$CameraTrapID, family="vonmises",
  saveclustIDs=TRUE, saveYExp=TRUE,
  scale=c("2pi"), autorun=FALSE, adapt=0, sample=300, burnin=300, thin=1)
FoxPPC<-GCMMppc(FoxGCMMPPC)
```

---

GCMMprob *Probability of Activity during Time Period*

---

### Description

Calculate activity probability estimates from a GCMM model during a specific period of time

### Usage

```
GCMMprob(model, timestart, timeend, sample = 1000)
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
timestart	Start of time period (in radians)
timeend	End of time period (in radians)
sample	Number of posterior samples from which to calculate the HDI; default=1000

**Value**

Returns object of class `GCMMestimate` containing list with summary of output with mean and 95% HDI and the posterior distribution of predicted activity probability

---

GCMMsims	<i>Create GCMM simulations</i>
----------	--------------------------------

---

**Description**

Support function that creates posterior simulations of GCMM activity curves

**Usage**

```
GCMMsims(PD, s)
```

**Arguments**

PD	Posterior samples of GCMM parameters; output from <code>GCMMsimparams</code> function
s	Index value for running simulations

**Value**

Returns a vector of data simulated from the GCMM mixture

---

GCMMsimparams	<i>Extract GCMM parameters for running GCMM simulations</i>
---------------	---

---

**Description**

Support function that extracts posterior samples of GCMM parameters for posterior simulations of GCMM activity curves

**Usage**

```
GCMMsimparams(model, sample)
```

**Arguments**

model	Object of class GCMM containing output from GCMM function
sample	Number of posterior samples

**Value**

Returns a vector posterior draws

---

HDI	<i>Calculate highest density interval</i>
-----	---

---

**Description**

Calculates the highest density interval

**Usage**

```
HDI(x, prob = 0.95)
```

**Arguments**

x	Vector of data
prob	Value for probability mass of HDI; default=95%

**Value**

Returns matrix of the mean and upper and lower bounds of the HDI

---

HPDoverlap	<i>Activity HPD Overlap</i>
------------	-----------------------------

---

**Description**

Calculate whether there is overlap between two HPDs, the amount of overlap, and the probability of activity during the HPD of the other

**Usage**

```
HPDoverlap(model1, model2, prob = 0.5, sample = 1000, scale = "2pi")
```

**Arguments**

model1	Object of class GCMM with output from <a href="#">GCMM</a> function
model2	Object of class GCMM with output from <a href="#">GCMM</a> function
prob	Value of probability density mass for activityHPD; default=0.50
sample	Number of posterior samples for calculating 95% HDI; default=1000
scale	Scale for plotting the activity curve, either "2pi" for 0,2pi or "pi" for -pi,pi; default="2pi"

**Value**

Returns object of class `GCMMestimate` containing list of posterior distributions and summary information of mean and 95% HDI

---

humanssample	<i>Sample data of camera trap observations of humans</i>
--------------	--

---

**Description**

Example dataset for fitting circular mixed effect mixture models with `activityGCMM` package

**Usage**

```
humanssample
```

**Format**

Dataframes with 3 variables Radians Time of observations, in radians (0 to 2pi) CameraTrapID Variable identifying camera traps SamplingPeriod Variable identifying sampling period during which camera traps were recording

**Details**

Sample data of camera trap observations of humans

**Source**

\ Campbell L.A.D. 2017

**Examples**

```
data(humanssample)
## Not run: GCMM(data=humanssample$Radians, RE1=humanssample$SamplingPeriod,
  scale=c("2pi"), family="vonmises", autojags=TRUE, thin=3)
## End(Not run)
```

mixtureplot

*GCMM Mixture Plot***Description**

Plot of estimated activity curve from the circular mixture model

**Usage**

```
mixtureplot(
  model,
  rug = FALSE,
  ruglwd = 2,
  lwd = 3,
  scale = "NULL",
  ymax = "NULL",
  col = "black",
  lty = 1,
  axisunits = c("radians", "time", "sun", "solar", "none"),
  xlines = TRUE
)
```

**Arguments**

model	Model output from <a href="#">GCMM</a> function, object of class GCMM
rug	Logical argument for whether to plot a rug of the raw values. Plotting the rug for the separate components requires that <code>saveclustIDs=TRUE</code> when running <a href="#">GCMM</a> or <a href="#">updateGCMM</a> . default=FALSE
ruglwd	Line width for rug plot
lwd	Line width for activity curve
scale	Scale for the plot, either "pi" (-pi, pi) or "2pi" (0, 2pi); default is that recommended by the <a href="#">GCMM</a> function
ymax	Value to use as y-axis maximum
col	Line colour for plot
lty	Line type for activity curve
axisunits	Scale to use for the xaxis, either "radians", "time", "solar", "sun", or "none"; default="radians"
xlines	Whether to include lines on the graph for the x axis labels; default=TRUE

**Value**

Prints mixture plot of the estimated activity curve from the circular mixture model

**Examples**

```
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians,
  RE1=redfoxsample$SamplingPeriod, family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1, n.chains=2)
mixtureplot(FoxActivityGCMM)
```

---

mode	<i>Mode</i>
------	-------------

---

**Description**

Returns the mode of a vector

**Usage**

```
mode(x)
```

**Arguments**

x                    Vector of data

**Value**

Returns the mode of x

---

multiplot	<i>Plot multiple GCMM activity curves</i>
-----------	---

---

**Description**

Plot of multiple GCMM activity curves

**Usage**

```
multiplot(
  models,
  ymax = "NULL",
  scale = "2pi",
  lwd = 3,
  type = c("mixture", "components"),
  lty = c(1, 2, 3, 4, 5),
  col = c("grey15", "grey40", "grey55", "grey70"),
  axisunits = c("radians", "sun", "solar", "time", "none"),
  xlines = TRUE
)
```



**Arguments**

models	List of one or more objects of class GCMM containing output from the <a href="#">GCMM</a> function
ymax	Value for upper limit of y-axis
scale	Scale for the plot, either "pi" (-pi, pi) or "2pi" (0, 2pi); default is that recommended by the GCMM function
lwd	Value for the width of the lines for the activity curves
type	Type of activity plots, either "mixture" for mixture plots (default) or "components" for components plots
lty	Vector of line types to use for the activity curves
col	Vector of colours to use for the activity curve lines in the plot
axisunits	Scale to use for the xaxis, either "radians", "time", "solar", "sun", or "none"; default="radians"
xlines	Whether to include lines on the graph for the x axis labels; default=TRUE

**Value**

Prints plot

**Examples**

```
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians,
  RE1=redfoxsample$SamplingPeriod, family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1, n.chains=2)
HumanActivityGCMM<-GCMM(data=humanssample$Radians, RE1=humanssample$SamplingPeriod,
  family="vonmises", autorun=FALSE, adapt=0, sample=300, burnin=300, thin=1, n.chains=2)
multiplot(models=list(FoxActivityGCMM,HumanActivityGCMM))
```

---

PDS

*Posterior distribution summaries and support*

---

**Description**

Calculates the proportion  $> 0$ ,  $< 0$ , and posterior distribution support (PDS)

**Usage**

PDS(x)

**Arguments**

x                      Vector of data

**Value**

Returns matrix with data summary

---

peaksPDplot	<i>Plot estimated number of activity peaks</i>
-------------	--

---

**Description**

Plot of posterior samples of estimated number of activity peaks from `activityHPD` function

**Usage**

```
peaksPDplot(x, col = "cyan4")
```

**Arguments**

x	Object of class <code>GCMMEstimate</code> with output from <code>activityHPD</code> function
col	Colour for plot

**Value**

No return value; prints histogram plot of posterior estimates

---

plotactivityHPD	<i>Plot activity curve with activityHPDs</i>
-----------------	--

---

**Description**

Plot GCM activity curve with activityHPDs

**Usage**

```
plotactivityHPD(
  model,
  prob = c(0.75, 0.5, 0.25),
  col = c("lightseagreen", "aquamarine3", "aquamarine"),
  scale = "2pi",
  ymax = "NULL",
  axisunits = "radians"
)
```

**Arguments**

model	Object of class <code>GCM</code> with output from <code>GCM</code> function
prob	Vector of values of probability density mass for activityHPD; default=c(0.50,0.25)
col	Vector of colours for plot
scale	Scale for plotting the activity curve, either "2pi" for 0,2pi or "pi" for -pi,pi; default="2pi"
ymax	Value for upper limit of y axis
axisunits	Units for x axis, either "radians", "time", "solar", "sun", or "none"; default="radians"

**Value**

No return value; prints plot of activity curve with activityHPDs

---

plotGCMMsamples      *Plot GCMM Activity Curve Posterior Samples*

---

**Description**

Plot GCMM activity curve posterior samples for visualizing estimate uncertainty

**Usage**

```
plotGCMMsamples(
  model,
  sample = 100,
  scale = "NULL",
  ymax = "NULL",
  plotmean = TRUE,
  RGB = c(200, 200, 200),
  alpha = 0.05,
  axisunits = "radians",
  lines = TRUE,
  cex.axis = 0.8
)
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
sample	Number of posterior samples to plot; default=100
scale	Scale for which to plot the activity curve, either "pi" for $-\pi$ , $\pi$ or "2pi" for 0, $2\pi$ ; default is that which is recommended by the GCMM function
ymax	Value to use as upper limit for y-axis
plotmean	Logical argument for whether to plot activity curve from posterior distribution mean; default=TRUE
RGB	Vector of RGB values for line colour
alpha	Value for line transparency, between 0 (completely transparent) to 1 (completely opaque); default=0.05
axisunits	Scale to use for the xaxis, either "radians", "time", "solar", "sun", or "non"; default="radians"
lines	Whether to include lines on the graph for the x axis labels; default=TRUE
cex.axis	Font size for axis labels

**Value**

No return value; prints plot of activity curve posterior samples

---

plotREs

*Plot GCMM activity curve with random intercepts*


---

**Description**

Plot GCMM activity curve with random intercepts

**Usage**

```
plotREs(
  model,
  RE1 = TRUE,
  RE2 = FALSE,
  scale = "NULL",
  ymax = "NULL",
  axisunits = "radians"
)
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
RE1	Logical vector for whether to plot GCMM activity curve with random intercepts from RE1; default=TRUE
RE2	Logical vector for whether to plot GCMM activity curve with random intercepts from RE2; default=FALSE
scale	Scale for plotting the activity curve, either "2pi" for 0,2pi or "pi" for -pi,pi; default="2pi"
ymax	Value for upper limit of y axis
axisunits	Units for x axis, either "radians", "time", "solar", "sun", or "none"; default="radians"

**Value**

No return value; prints plot of GCMM activity curve with random intercepts

**Examples**

```
FoxGCMMREs<-GCMM(data=redfoxsample$Radians, RE1=redfoxsample$SamplingPeriod,
  saveREs=TRUE, scale=c("2pi"), family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1)
plotREs(FoxGCMMREs)
```

---

posteriorhistplot	<i>Plot histogram of posterior distribution</i>
-------------------	---

---

**Description**

Plot histogram of posterior samples

**Usage**

```
posteriorhistplot(model, param, col = "cyan4")
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function
param	Parameter for which to plot the posterior samples
col	Colour of histogram

**Value**

Returns matrix with the mean and HDI of activity probability density estimated from both GCMM models at the peak activity time of the other. Posterior distributions of activity probability density and peak activity times for both GCMM models are also saved.

---

progress	<i>Report function progress</i>
----------	---------------------------------

---

**Description**

Support function that prints progress of functions with long computation times at 10% intervals

**Usage**

```
progress(s, sample)
```

**Arguments**

s	Number of iteration in the loop
sample	Total number of iterations in the loop

**Value**

No return value; prints progress of function at 10% intervals

---

redfoxsample

*Sample data of camera trap observations of red fox*

---

**Description**

Example dataset for fitting circular mixed effect mixture models with activityGCMM package

**Usage**

redfoxsample

**Format**

Dataframes with 3 variables Radians Time of observations, in radians (0 to 2pi) CameraTrapID Variable identifying camera traps SamplingPeriod Variable identifying sampling period during which camera traps were recording

**Details**

Sample data of camera trap observations of humans

**Source**

\ Campbell L.A.D. 2017

**Examples**

```
data(redfoxsample)
## Not run: GCMM(data=redfoxsample$Radians, RE1=redfoxsample$SamplingPeriod,
  scale=c("2pi"), family="vonmises", autojags=FALSE,
  adapt=0, sample=300, burnin=300, thin=1, n.chains=2 )
## End(Not run)
```

---

samplerows

*Sample rows of dataframe*

---

**Description**

Support function that samples rows of data from a dataframe

**Usage**

samplerows(df, n)

**Arguments**

df	Dataframe
n	Number of samples

**Value**

Returns sample of dataframe with the number of specified rows

---

sumCircResids	<i>Calculate sum of absolute circular residuals</i>
---------------	---

---

**Description**

Calculate posterior probability distribution of summed absolute circular residuals for assessing GCMM model fit

**Usage**

```
sumCircResids(model)
```

**Arguments**

model	Object of class GCMM with output from <a href="#">GCMM</a> function; residuals must be saved when running the <a href="#">GCMM</a> function using saveResids=TRUE
-------	---

**Value**

Returns list with output summary with mean and 95% HDI and posterior distribution of summed absolute circular residuals

**Examples**

```
FoxGCMMresids<-GCMM(data=redfoxsample$Radians, RE1=redfoxsample$SamplingPeriod, saveResids=TRUE,
  scale=c("2pi"), family="vonmises", autorun=FALSE, adapt=0, sample=300, burnin=300, thin=1)
FoxResids<-sumCircResids(FoxGCMMresids)
```

---

 updateGCMM

*Extend GCMM analysis*


---

### Description

Extend GCMM analysis using [extend.jags](#) from package `runjags`

### Usage

```
updateGCMM(
  model,
  burnin = 0,
  sample = 10000,
  saveclustIDs = FALSE,
  saveREs = FALSE,
  saveResids = FALSE,
  autorun = TRUE,
  minESS = 5000,
  maxrep = 5,
  drop.chain = 0
)
```

### Arguments

<code>model</code>	Object of class GCMM that is produced by the <a href="#">GCMM</a> function
<code>burnin</code>	Number of iterations per MCMC chain to be discarded as a burn-in
<code>sample</code>	Number of iterations per MCMC chain
<code>saveclustIDs</code>	Whether to save component cluster identification for the data points; default=FALSE
<code>saveREs</code>	Whether random intercepts are saved in output; recommended to save only one of <code>saveREs</code> , <code>saveResids</code> or <code>saveYExp</code> at one time due to memory limitations; default=FALSE
<code>saveResids</code>	Whether model residuals are saved in output; recommended to save only one of <code>saveREs</code> , <code>saveResids</code> or <code>saveYExp</code> at one time due to memory limitations
<code>autorun</code>	Whether to automatically extend the analysis until MCMC chain convergence and minimum effective sample size (ESS) is achieved; default is TRUE
<code>minESS</code>	Desired minimum effective sample size (MCMC) when automatically extending the analysis using <code>autorun=TRUE</code> ; default is 5000
<code>maxrep</code>	Maximum number of times to automatically extend the analysis if MCMC chains have not converged or the minimum effective sample size is not reached; default=5
<code>drop.chain</code>	A number indicating which MCMC chain to drop from the updated analysis. This may be useful if one chain happens to converge on opposite clusters than the others.



**Value**

Returns an object of class GCMM with a list of analysis details and output; see [GCMM](#). A mixture plot of the estimated activity curve is also printed.

**See Also**

[GCMM extend.jags](#)

**Examples**

```
FoxActivityGCMM<-GCMM(data=redfoxsample$Radians,
  RE1=redfoxsample$SamplingPeriod, family="vonmises", autorun=FALSE,
  adapt=0, sample=300, burnin=300, thin=1, n.chains=2)
updateFoxGCMM<-updateGCMM(FoxActivityGCMM, sample=300, autorun=FALSE)
```

---

xaxis

*Axis labels for temporal activity plots*


---

**Description**

Support function for xaxis labels for graphing temporal activity curves

**Usage**

```
xaxis(
  axisunits = c("radians", "solar", "sun", "time", "none"),
  lines = TRUE,
  cex.axis = 0.8
)
```

**Arguments**

axisunits	Scale to use for the xaxis, either "radians", "time", "solar", "sun", or "none"; default="radians"
lines	Whether to include lines on the graph for the x axis labels; default=TRUE
cex.axis	Font size for axis labels

**Value**

Prints axis

---

yMax

*Calculate y-axis limit for plotting multiple activity curves*

---

**Description**

Identifies maximum probability density for multiple activity curves to select y-axis limit when plotting multiple curves

**Usage**

```
yMax(models, type = "mixture")
```

**Arguments**

models	List of objects of class GCMM containing output from GCMM function
type	Identifier for whether to use maximum probability density from GCMM mixture, using "mixture", or components, using "components"; default is to use the GCMM mixture density

**Value**

Returns a value of the maximum probability density plus buffer space to be used as the y-axis limit in activity curve plots

# Index

- \* **Bayesian**
    - GCMM, 15
  - \* **activity;**
    - GCMM, 15
  - \* **cameratrap;**
    - GCMM, 15
  - \* **circular;**
    - GCMM, 15
  - \* **datasets**
    - humanssample, 22
    - redfoxsample, 30
  - \* **mixture;**
    - GCMM, 15
  - \* **temporal;**
    - GCMM, 15
  - \* **vonmises;**
    - GCMM, 15
  - \* **wrappedcauchy;**
    - GCMM, 15
- activityHPD, 3, 18, 26  
activityHPDmean, 3  
APDatPeak, 4  
APDpointplot, 5
- calcAPD, 5  
calcprop, 6  
circaxis, 7  
cirplotHPD, 7  
cirplotmeans, 8  
cirplotREs, 9  
combineMCMC, 9  
comboplot, 10  
compareGCMM, 11  
compareGCMMfit, 12  
componentsplot, 13  
convertRad, 14
- exampleGCMM, 14  
extend.jags, 32, 33
- extractparam, 15
- GCMM, 3–5, 7–13, 15, 18–20, 22, 23, 25–29, 31–33
- GCMMpdens, 18  
GCMMpeaksplot, 18  
GCMMppc, 19  
GCMMprob, 19  
GCMMsims, 20  
GCMMsimsparams, 20
- HDI, 21  
HPDoverlap, 21  
humanssample, 22
- mixtureplot, 23  
mode, 24  
multiplot, 24
- PDS, 25  
peaksPDplot, 26  
plotactivityHPD, 26  
plotGCMMsamples, 27  
plotREs, 28  
posteriorhistplot, 29  
progress, 29
- redfoxsample, 30  
run.jags, 17
- samplerows, 30  
sumCircResids, 31
- updateGCMM, 23, 32
- xaxis, 33
- yMax, 34