

# Package: clustTMB (via r-universe)

November 14, 2024

**Title** Spatio-Temporal Finite Mixture Model using 'TMB'

**Version** 0.1.0

**Description** Fits a spatio-temporal finite mixture model using 'TMB'. Covariate, spatial and temporal random effects can be incorporated into the gating formula using multinomial logistic regression, the expert formula using a generalized linear mixed model framework, or both.

**License** GPL-3

**URL** <https://github.com/Andrea-Havron/clustTMB>,  
<https://andrea-havron.github.io/clustTMB/>

**BugReports** <https://github.com/Andrea-Havron/clustTMB/issues>

**Depends** R (>= 4.0.0)

**Imports** cluster, clustMixType, fmesher, lme4, Matrix, mclust, methods, MoEClust, reformulas, sf, stats, TMB (>= 1.9.0)

**Suggests** bookdown, covr, cowplot, dplyr, FMsmnReg, GGally, ggplot2, ggspatial, giscoR, inlabru, kableExtra, knitr, magrittr, MixSim, mvnfast, mvtnorm, palmerpenguins, rmarkdown, sdmTMB, sp, spData, splancs, testthat, tidy, tweedie, wesanderson

**LinkingTo** RcppEigen, TMB

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** yes

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**SystemRequirements** GNU make

**Config/pak/sysreqs** cmake libgdal-dev gdal-bin libgeos-dev make  
libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

**Repository** <https://andrea-havron.r-universe.dev>

**RemoteUrl** <https://github.com/andrea-havron/clusttmb>

**RemoteRef** HEAD

**RemoteSha** 3ed9476b73568222fd922805250f3f8833752dae

## Contents

clustTMB . . . . .	2
coef.clustTMB . . . . .	4
extractAIC.clustTMB . . . . .	5
fixStruct.names . . . . .	5
init.options . . . . .	6
logLik.clustTMB . . . . .	7
lognormal . . . . .	7
mkInitClass . . . . .	8
parm.lookup . . . . .	8
print.clustTMB . . . . .	9
run.options . . . . .	9
skewness . . . . .	10
splitForm . . . . .	10
summary.clustTMB . . . . .	11
tweedie . . . . .	12

**Index** **13**

---

clustTMB *Fit a finite mixture model using TMB*

---

## Description

Fit a finite mixture model using TMB

## Usage

```
clustTMB(
  response = NULL,
  expertformula = ~1,
  gatingformula = ~1,
  expertdata = NULL,
  gatingdata = NULL,
  family = gaussian(link = "identity"),
  Offset = NULL,
  G = 2,
  rr = list(spatial = NULL, temporal = NULL, random = NULL),
  covariance.structure = NULL,
  Start = list(),
  Map = list(),
```

```

initialization.args = list(control = init.options()),
spatial.list = list(loc = NULL, mesh = NULL, init.range = list(gating.range = NULL,
  expert.range = NULL)),
projection.dat = NULL,
control = run.options()
)

```

## Arguments

response	A numeric vector, matrix, or data frame of observations. When data are multivariate, rows correspond to observations and columns correspond to the multivariate response.
expertformula	Formula defining expert model. This formula corresponds to the covariates included in the response densities. Defaults to intercept only (~1) when no covariates are used.
gatingformula	Formula defining gating model. This formula corresponds to the covariates included in the mixing proportions (logistic regression). Defaults to intercept only (~1) when no covariates are used. When a random effects term is included in the gating network, this formula will be updated so that the intercept term is removed.
expertdata	Data frame containing expert model covariates.
gatingdata	Data frame containing gating model covariates.
family	Statistical distribution and link function of observations.
Offset	Constant in expertformula only used to offset density expectation.
G	Integer specifying the number of clusters.
rr	List specifying dimension of rank reduction in spatial, temporal, and/or random effects. Dimension must be smaller than the total dimension of the response. Rank reduction is applied only to the expertformula random effects. The rank reduction reduces the dimensionality of a correlated multivariate response to a smaller dimension independent response. When used, the covariance structure of the response is switched to 'Diagonal.' Defaults to NULL, no rank reduction. If rank reduction is used in conjunction with a random effect, that random effect must also be specified in the expert formula. Currently, rank reduction on temporal random effects is disabled.
covariance.structure	A character string specifying the covariance structure of the response using mclust naming scheme. See description of modelNames under ?Mclust for details.
Start	Set initial values for random effects parameters (fixed and random terms)
Map	Vector indicating parameter maps, see ?TMB::MakeADFun() for details. Defaults in clustTMB control this map argument and user input is limited
initialization.args	A list consisting of initialization settings used to generate initial values. control Calls <code>init.options()</code> to generate settings for initial values. Arguments of <code>init.options()</code> can be specified by the user.

1. `init.method` - Single character string indicating initial clustering method. Methods include: `hc`, `quantile`, `random`, `mclust`, `kmeans`, `mixed`, `user`. Defaults to `'hc'`. In the case where data are univariate and there are no covariates in the gating/expert formula, this defaults to `'quantile'`
  2. `hc.options` - Named list of two character strings specifying `hc` `modelName` and `hcUse` when `init.method = 'hc'`. The default `modelName` is `'VVV'` and the default use is `'SVD'` unless gating/expert covariates specified, in which case default in `VARS`. See `?mclust::mclust.options` for complete list of options.
  3. `mix.method` - String stating initialization method for mixed-type data (`init.method = 'mixed'`). Current default when Tweedie family specified. Options include: Gower `kmeans` (default), Gower `hclust`, and `kproto`.
  4. `user` - Numeric or character vector defining user specified initial classification. `init.method` must be set to `'user'` when using this option.
- `spatial.list` List of data objects needed when fitting a spatial GMRF model
- `projection.dat` Spatial Points class of projection coordinates or Spatial Points Dataframe containing projection coordinates and projection covariates
- `control` List controlling whether models are run and whether standard errors are calculated.

**Value**

list of objects from fitted model

**Examples**

```
data("faithful")
m1 <- clustTMB(response = faithful, covariance.structure = "VVV")
plot(faithful$eruptions, faithful$waiting, pch = 16, col = m1$report$classification + 1)
```

---

<code>coef.clustTMB</code>	<i>Get fixed-effect coefficients</i>
----------------------------	--------------------------------------

---

**Description**

Get fixed-effect coefficients

**Usage**

```
## S3 method for class 'clustTMB'
coef(object, complete = FALSE, ...)
```

**Arguments**

<code>object</code>	The fitted <code>clustTMB</code> model
<code>complete</code>	Currently ignored
<code>...</code>	Currently ignored

**Value**

names numeric vector

---

extractAIC.clustTMB     *Extract the AIC of a clustTMB model*

---

**Description**

Extract the AIC of a clustTMB model

**Usage**

```
## S3 method for class 'clustTMB'
extractAIC(fit, scale, k = 2, ...)
```

**Arguments**

fit	The fitted clustTMB model
scale	The scale, currently ignored
k	Penalization parameter, defaults to 2
...	Currently ignored

**Value**

numeric value

---

fixStruct.names     *Fixed Covariance Structure names*

---

**Description**

Fixed Covariance Structure names

**Usage**

```
fixStruct.names()
```

**Value**

character vector naming available fixed Covariance Structures

**Examples**

```
fixStruct.names()
```

---

init.options	<i>Initialization options with S3 classes</i>
--------------	---

---

**Description**

Initialization options with S3 classes

**Usage**

```
init.options(
  init.method = "hc",
  hc.options = list(modelName = "VVV", use = "SVD"),
  exp.init = list(mahala = TRUE),
  mix.method = "Gower kmeans",
  user.class = integer()
)
```

**Arguments**

init.method	Name of method used to set initial values. If init.method = 'user', must define 'user.class' with a classification vector.
hc.options	Model names and use when init.method is 'hc' following conventions of mclust::mclust.options()
exp.init	Turn on mahala initialization when expert network
mix.method	Initialization methods when data are mixed. Default method when data are Tweedie distributed.
user.class	Vector of classification vector set by user and required when init.method = 'user'

**Value**

list of initialization specifications

**Examples**

```
init.options()
init.options(init.method = "hc")
init.options(init.method = "mixed")
init.options(init.method = "user", user.class = c(1, 1, 2, 1, 3, 3, 1, 2))
```

---

logLik.clustTMB	<i>Extract the log likelihood of a clustTMB model</i>
-----------------	---

---

**Description**

Extract the log likelihood of a clustTMB model

**Usage**

```
## S3 method for class 'clustTMB'  
logLik(object, ...)
```

**Arguments**

object	The fitted clustTMB model
...	Currently ignored

**Value**

object of class logLik with attributes

---

lognormal	<i>Lognormal family and link specification</i>
-----------	--

---

**Description**

Lognormal family and link specification

**Usage**

```
lognormal(link = "identity")
```

**Arguments**

link	link function association with family
------	---------------------------------------

**Value**

An object of class "family"

**Examples**

```
fam <- lognormal()  
fam$family  
fam$link
```

---

mkInitClass	<i>Apply classification method dependent on init.method</i>
-------------	---

---

**Description**

Apply classification method dependent on init.method

**Usage**

```
mkInitClass(n.g, n.i, n.j, control, y)
```

**Arguments**

n.g	Number of clusters
n.i	Number of observations
n.j	Number of columns
control	Classification settings from <a href="#">init.options()</a>
y	Observations

**Value**

classification vector

**Examples**

```
data("faithful")
mkInitClass(2, nrow(faithful), ncol(faithful), init.options(), faithful)
```

---

parm.lookup	<i>Parameter Information</i>
-------------	------------------------------

---

**Description**

Parameter Information

**Usage**

```
parm.lookup()
```

**Value**

Description of parameters, including dimension and structure

**Examples**

```
parm.lookup()
```



---

```
print.clustTMB          Print brief model summary
```

---

**Description**

Invoke TMB's print.report function

**Usage**

```
## S3 method for class 'clustTMB'
print(x, ...)
```

**Arguments**

x	The fitted clustTMB model
...	Not used

**Value**

numeric matrix of parameter estimate and standard errors

---

```
run.options          Run Options
```

---

**Description**

Run Options

**Usage**

```
run.options(check.input = NULL, run.model = NULL, do.sdreport = NULL)
```

**Arguments**

check.input	TRUE: Return initial values before running TMB
run.model	FALSE: Return TMB object before optimizing model
do.sdreport	TRUE: Run delta method to obtain standard errors

**Value**

list

**Examples**

```
run.options()
```

---

skewness	<i>Calculates skewness</i>
----------	----------------------------

---

**Description**

Calculates skewness

**Usage**

```
skewness(x)
```

**Arguments**

x                    numeric vector of values for which skewness is calculated

**Value**

skewness value of x

**Examples**

```
skewness(rgamma(100, 1, 1))
```

---

splitForm	<i>Split formula containing special random effect terms</i>
-----------	---

---

**Description**

Parse a formula into fixed formula and random effect terms, treating 'special' terms appropriately

**Usage**

```
splitForm(
  formula,
  defaultTerm = "norm",
  allowFixedOnly = TRUE,
  allowNoSpecials = TRUE,
  debug = FALSE
)
```

**Arguments**

formula	a formula containing special random effect terms
defaultTerm	default type for non-special RE terms
allowFixedOnly	(logical) are formulas with no RE terms OK?
allowNoSpecials	(logical) are formulas with only standard RE terms OK?
debug	(logical) debug?

**Details**

Taken from Steve Walker's lme4ord, ultimately from the flexLambda branch of lme4 <https://github.com/stevencarlislewalker/lme4ord/blob/master/R/formulaParsing.R>. Mostly for internal use.

**Value**

a list containing elements fixedFormula; reTrmFormulas list of x | g formulas for each term; reTrmAddArgs list of function+additional arguments, i.e. list() (non-special), foo() (no additional arguments), foo(addArgs) (additional arguments); reTrmClasses (vector of special functions/classes, as character)

**Author(s)**

Steve Walker

**Examples**

```
splitForm(~x+y)                ## no specials or RE
splitForm(~x+y+(f|g))          ## no specials
splitForm(~x+y+diag(f|g))      ## one special
splitForm(~x+y+(diag(f|g)))    ## 'hidden' special
splitForm(~x+y+(f|g)+cs(1|g))  ## combination
splitForm(~x+y+(1|f/g))        ## 'slash'; term
splitForm(~x+y+(1|f/g/h))      ## 'slash'; term
splitForm(~x+y+(1|(f/g)/h))    ## 'slash'; term
splitForm(~x+y+(f|g)+cs(1|g)+cs(a|b,stuff)) ## complex special
splitForm(~((x+y)))            ## lots of parentheses
splitForm(~1+rr(f|g,n=2))
```

---

summary.clustTMB

*summary tables of model parameters*

---

**Description**

Invoke TMB's summary.sdreport function

**Usage**

```
## S3 method for class 'clustTMB'
summary(
  object,
  select = c("all", "fixed", "random", "report"),
  p.value = FALSE,
  ...
)
```

**Arguments**

object	The fitted clustTMB model
select	Parameter classes to select. Can be any subset of "fixed" ( $\hat{\theta}$ ), "random" ( $\hat{u}$ ) or "report" ( $\phi(\hat{u}, \hat{\theta})$ ) using notation as <code>TMB::sdreport()</code> .
p.value	Add column with approximate p-values
...	Currently ignored

**Value**

numeric matrix of parameter estimate and standard errors

---

tweedie	<i>Tweedie family and link specification</i>
---------	--

---

**Description**

Tweedie family and link specification

**Usage**

```
tweedie(link = "log")
```

**Arguments**

link	link function association with family
------	---------------------------------------

**Value**

An object of class "family"

**Examples**

```
fam <- tweedie()
fam$family
fam$link
```

# Index

`clustTMB`, [2](#)  
`coef.clustTMB`, [4](#)  
`extractAIC.clustTMB`, [5](#)  
`fixStruct.names`, [5](#)  
`init.options`, [6](#)  
`init.options()`, [3](#), [8](#)  
`logLik.clustTMB`, [7](#)  
`lognormal`, [7](#)  
`mkInitClass`, [8](#)  
`parm.lookup`, [8](#)  
`print.clustTMB`, [9](#)  
`run.options`, [9](#)  
`skewness`, [10](#)  
`splitForm`, [10](#)  
`summary.clustTMB`, [11](#)  
`TMB::MakeADFun()`, [3](#)  
`TMB::sdreport()`, [12](#)  
`tweedie`, [12](#)