# Package 'gdm'

November 18, 2021

**Title** Generalized Dissimilarity Modeling

**Version** 1.5.0-1

**Date** 2021-11-16

**Description**
A toolkit with functions to fit, plot, summarize, and apply Generalized Dissimilarity Models.

**License** GPL (>= 3)

**Depends** R (>= 3.5.0)

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Config/testthat/edition** 3

**LazyData** true

**Imports** parallel, methods, raster, Rcpp, reshape2, vegan, doParallel, foreach

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Author** Matt Fitzpatrick [aut, cre] (<https://orcid.org/0000-0003-1911-8407>),
Karel Mokany [aut] (<https://orcid.org/0000-0003-4199-3697>),
Glenn Manion [aut],
Diego Nieto-Lugilde [aut] (<https://orcid.org/0000-0003-4135-2881>),
Simon Ferrier [aut] (<https://orcid.org/0000-0001-7884-2388>),
Matthew Lisk [ctb],
Chris Ware [ctb],
Skip Woolley [ctb],
Tom Harwood [ctb]

**Maintainer** Matt Fitzpatrick <mfitzpatrick@umces.edu>

**Repository** CRAN

**Date/Publication** 2021-11-18 16:10:04 UTC

# R topics documented:



---

`gdm-package` *Overview of the functions in the gdm package*

---

## Description

Generalized Dissimilarity Modeling is a statistical technique for modelling variation in biodiversity between pairs of geographical locations or through time. The **gdm** package provides functions to fit, evaluate, summarize, and plot Generalized Dissimilarity Models and to make predictions (across space and/or through time) and map biological patterns by transforming environmental predictor variables.

## Details

The functions in the **gdm** package provide the tools necessary for fitting GDMs, including functions to prepare biodiversity and environmental data. Major functionality includes:

- Formatting various types of biodiversity and environmental data to **gdm**'s site-pair format used in model fitting
- Fitting GDMs using geographic and environmental distances between sites
- Plotting fitted functions & extracting I-spline values
- Estimating predictor importance using matrix permutation and predictor contributions using deviance paritioning
- Using cross-validation to evaluate models
- Predicting pairwise dissimiliarites between sites or times and transforming environmental predictors to biological importance and mapping these patterns.

To see the preferable citation of the package, type `citation("gdm")`.

### I. Formatting input data

GDM fits biological distances to pairwise site geographical and environmental distances. Most users will need to first format their data to **gdm**'s site-pair table format:

| | |
|---|---|
| `formatsitepair` | To convert biodiversity and environmental data to site-pair format |

### II. Model fitting, evaluation, and summary

| | |
|---|---|
| `gdm` | To fit a GDM |
| `gdm.crossvalidation` | To evaluate a GDM |
| `gdm.partition.deviance` | To asses predictor contributions to deviance explained |
| `gdm.varImp` | To asses model significance and predictor importance |
| `summary` | To summarize a GDM |

### III. Model prediction and transformation of environmental data

| | |
|---|---|
| `predict` | To predict biological dissimilarities between sites in space or between time periods |
| `gdm.transform` | To transform each environmental predictor to biological importance |

### IV. Plotting model output and fitted functions

| | |
|---|---|
| `plot` | To plot model fit and I-splines |
| `isplineExtract` | To extract I-spline values to allow for custom plotting |
| `plotUncertainty` | To estimate and plot model sensitivity using bootstrapping |

### Author(s)

The **gdm** development team is Matt Fitzpatrick and Karel Mokany. The R package is based on code originally developed by Glenn Manion under the direction of Simon Ferrier. Where others have contributed to individual functions, credits are provided in function help pages.

The maintainer of the R version of **gdm** is Matt Fitzpatrick <mfitzpatrick@umces.edu>.

`calculate.gdm.deviance` *Calculate GDM Deviance for Observed & Predicted Dissimilarities*

## Description

Calculate GDM deviance for observed & predicted dissimilarities. Can be used for assessing cross-validation data. Translated from the c++ function CalcGDMDevianceDouble() in the file NNLS_Double.cpp from the GDM R package.

## Usage

```
calculate.gdm.deviance(predDiss, obsDiss)
```

## Arguments

| | |
|---|---|
| `predDiss` | (float) A vector of predicted dissimilarity values, of same length as obsDiss. |
| `obsDiss` | (float) A vector of observed dissimilarity values, of same length as predDiss. |

## Value

A single value (float) being the deviance.

`formatsitepair` *Combines Biological and Environmental Data to Produce a GDM-formatted Site-Pair Table*

## Description

This function takes input biological data and environmental, geographic, and other predictor data and builds a site-pair table required for fitting a Generalized Dissimilarity Model using the `gdm` function. NOTE: x-y coordinates of sites MUST be present in either the biological or the environmental data. Site coordinates ideally should be in a projected coordinate system (i.e., not longitude-latitude) to ensure proper calculation of geographic distances.

The input biological data can be in one of the following four formats. Note that the general term "species" is used, but any classification of biological entities (e.g. functional types, haplotypes, etc) can be used as long as an appropriate distance metric is also supplied (see "dist" argument):

1. site-by-species matrix
2. x, y, species list

3. site-by-site biological distance (dissimilarity) matrix

4. an existing site-pair table (see Details)

Predictor data can be provided in three formats:

1. a site-by-predictor matrix with a column for each predictor variable and a row for each site

2. a raster stack, with one raster for each predictor variable

3. one or more site-by-site distance matrices using the "distPreds" argument (see below).

## Usage

```
formatsitepair(bioData, bioFormat, dist="bray", abundance=FALSE, siteColumn=NULL,
XColumn, YColumn, sppColumn=NULL, abundColumn=NULL, sppFilter=0, predData,
distPreds=NULL, weightType="equal", custWeights=NULL, sampleSites=1)
```

## Arguments

| | |
|---|---|
| bioData | The input biological (the response variable) data table, in one of the four formats defined above (see Details). |
| bioFormat | An integer code specifying the format of bioData. Acceptable values are 1, 2, 3, or 4 (see Details). |
| dist | Default = "bray". A character code indicating the metric to quantify pairwise site distances / dissimilarities. Calls the vegdist function from the **vegan** package to calculate dissimilarity and therefore accepts any method available from vegdist. |
| abundance | Default = FALSE. Indicates whether the biological data are abundance data (TRUE) or presence-absence (0, 1) data (FALSE). |
| siteColumn | The name of the column in either the biological or environmental data table containing a unique site identifier. If a site column is provided in both the biological and environmental data, the site column name must be the same in both tables. |
| XColumn | The name of the column containing x-coordinates of sites. X-coordinates can be provided in either the biological or environmental data tables, but MUST be in at least one of them. If an x-coordinate column is provided in both the biological and environmental data, the column name must be identical. Site coordinates ideally should be in a projected coordinate system (i.e., not longitude-latitude) to ensure proper calculation of geographic distances. Note that if you are using rasters, they must be in the same coordinate system as the site coordinates. |
| YColumn | The name of the column containing y-coordinates of sample sites. Y-coordinates can be provided in either the biological or environmental data tables, but MUST be in at least one of them. If a y-coordinate column is provided in both the |

biological and environmental data, the column name must be identical. Site coordinates ideally should be in a projected coordinate system (i.e., not longitude-latitude) to ensure proper calculation of geographic distances. Note that if you are using rasters, they must be in the same coordinate system as the site coordinates.

`sppColumn` Only used if bioFormat = 2 (x, y, species list). The name of the column containing unique name / identifier for each species.

`abundColumn` If abundance = TRUE, this parameter identifies the column containing the measure of abundance at each site. Only used if bioFormat = 2 (i.e., x, y, species list), though in the case of abundance data, the table would have four columns: x, y, species, abundance.

`sppFilter` Default = 0. To account for limited sampling effort at some sites, sppFilter removes all sites at which the number of recorded species (i.e., observed species richness) is less than the specified value. For example, if sppFilter = 5, all sites with fewer than 5 recorded species will be removed.

`predData` The environmental predictor data. Accepts either a site-by-predictor table or a raster stack.

`distPreds` An optional list of distance matrices to be used as predictors in combination with predData. For example, a site-by-site dissimilarity matrix for one biological group (e.g., trees) can be used as a predictor for another group (e.g., ferns). Each distance matrix must have as the first column the names of the sites (therefore the matrix will not be square). The name of the column containing the site names should have the same name as that provided for the siteColumn argument. Site IDs are required here to ensure correct ordering of sites in the construction of the site-pair table. Note that the formatsitepair function will not accept only distances matrices as predictors (i.e., at least one predictor variable is required). If you wish to fit GDM using only distance matrices, provide one fake predictor (e.g., with all sites have the same value), plus site and coordinate columns if needed. The s1 and s2 columns for this variable can then be removed by hand before fitting the GDM.

`weightType` Default = "equal". Defines the weighting for sites. Can be either: (1) "equal" (weights for all sites set = 1), (2) "richness" (each site weighted according to number of species recorded), or (3) "custom" (user defined). If weightType="custom", the user must provide a vector of site weights equal to the number of rows in the full site-pair table (i.e., before species filtering (sppFilter argument) or sub-sampling is taken into account (sampleSites argument)).

`custWeights` A two column matrix or data frame of user-defined site weights. The first column should be the site name and should be named the same as that provided for the siteColumn argument. The second column should be numeric weight values and should be named "weights". The weight values represent the importance of each site in model fitting, and the values in the output site-pair table is an average of the two sites in each site-pair. Required when weightType = "custom". Ignored otherwise.

`sampleSites` Default = 1. A number between 0-1 indicating the fraction of sites to be used to construct the site-pair table. This argument can be used to reduce the number of sites to overcome possible memory limitations when fitting models with very large numbers of sites.

## Details

bioData and bioFormat: The function accepts biological data in the following formats:

bioData = site-by-species matrix; bioFormat = 1: assumes that the response data are provided with a site ID column (specified by siteCol) and, optionally, two columns for the x & y coordinates of the sites. All remaining columns contain the biological data, with a column for each biological entity (most commonly species). In the case that a raster stack is provided for the environmental data (predData), x-y coordinates MUST be provided in bioData to allow extraction of the environmental data at site locations. The x-y coordinates will be intersected with the raster stack and, if the number of unique cells intersected by the points is less than the number of unique site IDs (i.e. multiple sites fall within a single cell), the function will use the raster cell as the site ID and aggregate sites accordingly. Therefore, model fitting will be sensitive to raster cell size. If the environmental data are in tabular format, they should have the same number of sites (i.e., same number of rows) as bioData. The x-y coordinate and site ID columns must have the same names in bioData and predData.

bioData = x, y, species list (optionally a fourth column with abundance can be provided); bioFormat = 2: assumes a table of 3 or 4 columns, the first two being the x & y coordinates of species records, the third (sppCol) being the name / identifier of the species observed at that location, and optionally a fourth column indicating a measure of abundance. If an abundance column is not provided, presence-only data are assumed. In the case that a raster stack is provided for the environmental data (predData), the x-y coordinates will be intersected with the raster stack and, if the number of unique cells intersected by the points is less than the number of unique site IDs (i.e. multiple sites fall within a single cell), the function will use the raster cell as the site ID and aggregate sites accordingly. Therefore, model fitting will be sensitive to raster cell size.

bioData = site-pair table; bioFormat = 4: with an already created site-pair table, this option allows the user to add one or more distance matrices (see distPreds above) to the existing site-pair table and/or sub-sample the site-pair table (see sample above). If the site-pair table was not created using the formatsitepair function, the user will need to ensure the order of the sites matches that in other tables being provided to the function.

NOTES: (1) The function assumes that the x-y coordinates and the raster stack (if used) are in the same coordinate system. No checking is performed to confirm this is the case. (2) The function assumes that the association between the provided site and x-y coordinate columns are singular and unique. Therefore, the function will fail should a given site has more than one sets of coordinates associated with it, as well as multiple sites being given the exact same coordinates.

## Value

A formatted site-pair table containing the response (biological distance or dissimilarity), predictors, and weights as required for fitting Generalized Dissimilarity Models.

## Examples

```
## tabular data
# start with the southwest data table
 head(southwest)
 sppData <- southwest[c(1,2,13,14)]
 envTab <- southwest[c(2:ncol(southwest))]

#########table type 1
```

```
## site-species table without coordinates
testData1a <- reshape2::dcast(sppData, site~species)
##site-species table with coordinates
coords <- unique(sppData[, 2:ncol(sppData)])
testData1b <- merge(testData1a, coords, by="site")
## site-species, table-table
exFormat1a <- formatsitepair(testData1a, 1, siteColumn="site", XColumn="Long",
YColumn="Lat", predData=envTab)

#' # next, let's try environmental raster data
## not run
# rastFile <- system.file("./extdata/swBioclims.grd", package="gdm")
# envRast <- stack(rastFile)

## site-species, table-raster
## not run
# exFormat1b <- formatsitepair(testData1b, 1, siteColumn="site", XColumn="Long",
# YColumn="Lat", predData=envRast)

#########table type 2
## site xy spp list, table-table
exFormat2a <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat",
sppColumn="species", siteColumn="site", predData=envTab)

## site xy spp list, table-raster
## not run
# exFormat2b <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat",
# sppColumn="species", siteColumn="site", predData=envRast)

#########table type 3
## It is possible to format a site-pair table by starting
# with a pre-calculated matrix of biological distances
dim(gdmDissim) #square pairwise distance matrix
gdmDissim[1:5, 1:5]
# need to add a site ID column
site <- unique(sppData$site)
gdmDissim <- cbind(site, gdmDissim)
# now we can format the table:
exFormat3 <- formatsitepair(gdmDissim, 3, XColumn="Long", YColumn="Lat",
predData=envTab, siteColumn="site")

#########table type 4
## adds a predictor matrix to an existing site-pair table, in this case,
## predData needs to be provided, but is not actually used
exFormat4 <- formatsitepair(exFormat2a, 4, predData=envTab, siteColumn="site",
                            distPreds=list(as.matrix(gdmDissim)))
```

---

gdm *Fit a Generalized Dissimilarity Model to Tabular Site-Pair Data*

---

## Description

The gdm function is used to fit a generalized dissimilarity model to tabular site-pair data formatted as follows using the formatsitepair function: distance, weights, s1.xCoord, s1.yCoord, s2.xCoord, s2.yCoord, s1.Pred1, s1.Pred2, ...,s1.PredN, s2.Pred1, s2.Pred2, ..., s2.PredN. The distance column contains the response variable must be any ratio-based dissimilarity (distance) measure between Site 1 and Site 2. The weights column defines any weighting to be applied during fitting of the model. If equal weighting is required, then all entries in this column should be set to 1.0 (default). The third and fourth columns, s1.xCoord and s1.yCoord, represent the spatial coordinates of the first site in the site pair (s1). The fifth and sixth columns, s2.xCoord and s2.yCoord, represent the coordinates of the second site (s2). Note that the first six columns are REQUIRED, even if you do not intend to use geographic distance as a predictor (in which case these columns can be loaded with dummy data if the actual coordinates are unknown - though that would be weird, no?). The next N*2 columns contain values for N predictors for Site 1, followed by values for the same N predictors for Site 2.

The following is an example of a GDM input table header with three environmental predictors (Temp, Rain, Bedrock):

distance, weights, s1.xCoord, s1.yCoord, s2.xCoord, s2.yCoord, s1.Temp, s1.Rain, s1.Bedrock, s2.Temp, s2.Rain, s2.Bedrock

## Usage

```
gdm(data, geo=FALSE, splines=NULL, knots=NULL)
```

## Arguments

| | |
|---|---|
| data | A data frame containing the site pairs to be used to fit the GDM (obtained using the formatsitepair function). The observed response data must be located in the first column. The weights to be applied to each site pair must be located in the second column. If geo is TRUE, then the s1.xCoord, s1.yCoord and s2.xCoord, s2.yCoord columns will be used to calculate the geographic distance between site pairs for inclusion as the geographic predictor term in the model. Site coordinates ideally should be in a projected coordinate system (i.e., not longitude-latitude) to ensure proper calculation of geographic distances. If geo is FALSE (default), then the s1.xCoord, s1.yCoord, s2.xCoord and s2.yCoord data columns must still be included, but are ignored in fitting the model. Columns containing the predictor data for Site 1, and the predictor data for Site 2, follow. |
| geo | Set to TRUE if geographic distance between sites is to be included as a model term. Set to FALSE if geographic distance is to be omitted from the model. Default is FALSE. |
| splines | An optional vector of the number of I-spline basis functions to be used for each predictor in fitting the model. If supplied, it must have the same length as the number of predictors (including geographic distance if geo is TRUE). If this vector is not provided (splines=NULL), then a default of 3 basis functions is used for all predictors. |

knots An optional vector of knots in *units of the predictor variables* to be used in the fitting process. If knots are supplied and splines=NULL, then the knots argument must have the same length as the number of predictors * n, where n is the number of knots (default=3). If both knots and the number of splines are supplied, then the length of the knots argument must be the same as the sum of the values in the splines vector. Note that the default values for knots when the default three I-spline basis functions are 0 (minimum), 50 (median), and 100 (maximum) quantiles.

## Value

gdm returns a gdm model object. The function summary.gdm can be used to obtain or print a synopsis of the results. A gdm model object is a list containing at least the following components:

**dataname** The name of the table used as the data argument to the model.

**geo** Whether geographic distance was used as a predictor in the model.

**gdmdeviance** The deviance of the fitted GDM model.

**nulldeviance** The deviance of the null model.

**explained** The percentage of null deviance explained by the fitted GDM model.

**intercept** The fitted value for the intercept term in the model.

**predictors** A list of the names of the predictors that were used to fit the model, in order of the amount of turnover associated with each predictor (based on the sum of the I-spline coefficients).

**coefficients** A list of the coefficients for each spline for each of the predictors considered in model fitting.

**knots** A vector of the knots derived from the x data (or user defined), for each predictor.

**splines** A vector of the number of I-spline basis functions used for each predictor.

**creationdate** The date and time of model creation.

**observed** The observed response for each site pair (from data column 1).

**predicted** The predicted response for each site pair, from the fitted model (after applying the link function).

**ecological** The linear predictor (ecological distance) for each site pair, from the fitted model (before applying the link function).

## References

Ferrier S, Manion G, Elith J, Richardson, K (2007) Using generalized dissimilarity modelling to analyse and predict patterns of beta diversity in regional biodiversity assessment. *Diversity & Distributions* 13, 252-264.

## See Also

formatsitepair, summary.gdm, plot.gdm, predict.gdm, gdm.transform

## Examples

```
##fit table environmental data
# format site-pair table using the southwest data table
head(southwest)
sppData <- southwest[c(1,2,13,14)]
envTab <- southwest[c(2:ncol(southwest))]

sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat", sppColumn="species",
                              siteColumn="site", predData=envTab)

##fit table GDM
gdmTabMod <- gdm(sitePairTab, geo=TRUE)
summary(gdmTabMod)

##fit raster environmental data
##sets up site-pair table
rastFile <- system.file("./extdata/swBioclims.grd", package="gdm")
envRast <- raster::stack(rastFile)

##environmental raster data
sitePairRast <- formatsitepair(sppData, 2, XColumn="Long",
                               YColumn="Lat", sppColumn="species",
                               siteColumn="site", predData=envRast)
##sometimes raster data returns NA in the site-pair table, these rows will
##have to be removed before fitting gdm
sitePairRast <- na.omit(sitePairRast)

##fit raster GDM
gdmRastMod <- gdm(sitePairRast, geo=TRUE)
summary(gdmRastMod)
```

---

# gdm.crossvalidation *Cross-Validation Assessment of a Fitted GDM*

---

## Description

Undertake a cross-validation assessment of a GDM fit using all the predictors included in the formated GDM input table (spTable). The cross-validation is run using a specified proportion (train.proportion) of the randomly selected sites included in spTable to train the model, with the remaining sites being used to test the performance of the model predictions. The test is repeated a specified number of times (n.crossvalid.tests), with a unique random sample taken each time. Outputs are a number of cross-validation test metrics.

## Usage

```
gdm.crossvalidation(spTable, train.proportion=0.9, n.crossvalid.tests=1,
geo=FALSE, splines=NULL, knots=NULL)
```

### Arguments

| | |
|---|---|
| spTable | (dataframe) A dataframe holding the GDM input table for model fitting. |
| train.proportion | (float) The proportion of sites in 'spTable' to use in training the GDM, with the remaining proportion used to test the model. (default = 0.9) |
| n.crossvalid.tests | (integer) The number of cross-validation sets to use in testing the GDM. (default = 1) |
| geo | (boolean) Geographic distance to be used in model fitting (default = FALSE). |
| splines | (vector) An optional vector of the number of I-spline basis functions to be used for each predictor in fitting the model. |
| knots | (vector) An optional vector of knots in units of the predictor variables to be used in the fitting process. |

### Value

List, providing cross-validation statistics. These are metrics that describe how well the model fit using the sitepair training table predicts the dissimilarities in the sitepair testing table. Metrics provided include: 'Deviance.Explained' (the deviance explained for the training data); 'Test.Deviance.Explained' (the deviance explained for the test data); 'Mean.Error'; 'Mean.Absolute.Error'; 'Root.Mean.Squre.Error'; 'Obs.Pred.Correlation' (Pearson's correlation coefficient between observed and predicted values); 'Equalised.RMSE' (the average root mean square error across bands of observed dissimilarities (0.05 dissimialrity units)); 'Error.by.Observed.Value' (the average root mean square error and number of observations within bands of observed dissimilarities (0.05 dissimialrity units)).

---

## gdm.partition.deviance

*Perform Deviance Partitioning of a Fitted GDM*

---

### Description

Partitions deviance explained from GDM into different user specified components - most typically environment versus space.

### Usage

```
gdm.partition.deviance(sitePairTable, varSets=list(), partSpace=TRUE)
```

### Arguments

| | |
|---|---|
| sitePairTable | A correctly formatted site-pair table from formatsitepair. |
| varSets | A list in which each element is a vector of variable names across which deviance partitioning is to be performed, excluding geographic distance (which is set by the partSpace argument). Variable names must match those used to build the site-pair table. See example. |
| partSpace | Whether or not to perform the partitioning using geographic space. Default=TRUE. |

## Value

A dataframe summarizing deviance partitioning results.

## Author(s)

Matt Fitzpatrick and Karel Mokany

## Examples

```
# set up site-pair table using the southwest data set
sppData <- southwest[c(1,2,13,14)]
envTab <- southwest[c(2:ncol(southwest))]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat",
sppColumn="species", siteColumn="site", predData=envTab)

# EXAMPLE - Partition two groups of variables
# Make list of variable sets for partitioning
varSet <- vector("list", 2)

# now, name the variable groups for partitioning
# note you do not need to add "space" as this is only needed
# for environmental variables

# two groups (soils & climate)
names(varSet) <- c("soil", "climate")

# lastly, add variable names for
varSet$soil <- c("awcA", "phTotal", "sandA", "shcA", "solumDepth")
varSet$climate <- c("bio5", "bio6", "bio15", "bio18", "bio19")
varSet

# run the function to partition soils, climate, and space (partSpace=TRUE)
scgPart <- gdm.partition.deviance(sitePairTab, varSet, partSpace=TRUE)

# EXAMPLE - Partition three groups of variables
# Make list of variable sets for partitioning
varSet <- vector("list", 3)
names(varSet) <- c("soil", "temp", "precip")
varSet$soil <- c("awcA", "phTotal", "sandA", "shcA", "solumDepth")
varSet$temp <- c("bio5", "bio6")
varSet$precip <- c("bio15", "bio18", "bio19")

# partition soils, temperature, and precip
# note we can't also partition space given the function's limit to a
# maximum of three variable sets, so we set partSpace=FALSE
scPart <- gdm.partition.deviance(sitePairTab, varSet, partSpace=FALSE)
```

`gdm.transform` *Transform Environmental Data Using a Fitted Generalized Dissimilarity Model*

## Description

This function transforms geographic and environmental predictors using (1) the fitted functions from a model object returned from `gdm` and (2) a data frame or raster stack containing predictor data for a set of sites.

## Usage

```
gdm.transform(model, data)
```

## Arguments

| | |
|---|---|
| `model` | A gdm model object resulting from a call to `gdm`. |
| `data` | Either (i) a data frame containing values for each predictor variable in the model, formatted as follows: X, Y, var1, var2, var3, ..., varN or (ii) a raster stack with one layer per predictor variable used in the model, excluding X and Y (rasters for x- and y-coordinates are built automatically from the input rasters if the model was fit with geo=T). The order of the columns (data frame) or raster layers (raster stack) MUST be the same as the order of the predictors in the site-pair table used in model fitting. There is currently no checking to ensure that the order of the variables to be transformed are the same as those in the site-pair table used in model fitting. If geographic distance was not used as a predictor in model fitting, the x- and y-columns need to be removed from the data to be transformed. Output is provided in the same format as the input data. |

## Value

gdm.transform returns either a data frame with the same number of rows as the input data frame or a raster stack, depending on the format of the input data. If the model uses geographic distance as a predictor the output object will contain columns or layers for the transformed X and Y values for each site. The transformed environmental data will be in the remaining columns or layers.

## References

Ferrier S, Manion G, Elith J, Richardson, K (2007) Using generalized dissimilarity modelling to analyse and predict patterns of beta diversity in regional biodiversity assessment. *Diversity & Distributions* 13, 252-264.

Fitzpatrick MC, Keller SR (2015) Ecological genomics meets community-level modeling of biodiversity: Mapping the genomic landscape of current and future environmental adaptation. *Ecology Letters* 18: 1-16

## Examples

```
# start with the southwest data set
# grab the columns with xy, site ID, and species data
sppTab <- southwest[, c("species", "site", "Lat", "Long")]

##fit gdm using rasters
rastFile <- system.file("./extdata/swBioclims.grd", package="gdm")
envRast <- raster::stack(rastFile)
sitePairRast <- formatsitepair(sppTab, 2, XColumn="Long", YColumn="Lat", sppColumn="species",
                               siteColumn="site", predData=envRast)
##remove NA values
sitePairRast <- na.omit(sitePairRast)

##fit raster GDM
gdmRastMod <- gdm(sitePairRast, geo=TRUE)

##raster input, raster output
transRasts <- gdm.transform(gdmRastMod, envRast)

# map biological patterns
rastDat <- raster::sampleRandom(transRasts, 10000)
pcaSamp <- prcomp(rastDat)

# note the use of the 'index' argument
pcaRast <- raster::predict(transRasts, pcaSamp, index=1:3)

# scale rasters
pcaRast[[1]] <- (pcaRast[[1]]-pcaRast[[1]]@data@min) /
  (pcaRast[[1]]@data@max-pcaRast[[1]]@data@min)*255
pcaRast[[2]] <- (pcaRast[[2]]-pcaRast[[2]]@data@min) /
  (pcaRast[[2]]@data@max-pcaRast[[2]]@data@min)*255
pcaRast[[3]] <- (pcaRast[[3]]-pcaRast[[3]]@data@min) /
  (pcaRast[[3]]@data@max-pcaRast[[3]]@data@min)*255

raster::plotRGB(pcaRast, r=1, g=2, b=3)
```

---

`gdm.varImp` *Quantify Model Significance and Variable Importance in a Fitted Generalized Dissimilarity Model Using Matrix Permutation.*

---

## Description

This function uses matrix permutation to perform model and variable significance testing and to estimate variable importance in a generalized dissimilarity model. The function can be run in parallel on multicore machines to reduce computation time.

## Usage

```
gdm.varImp(spTable, geo, splines = NULL, knots = NULL,
fullModelOnly = FALSE, nPerm = 50, parallel = FALSE, cores = 2,
sampleSites = 1, sampleSitePairs = 1, outFile = NULL)
```

## Arguments

| | |
|---|---|
| spTable | A site-pair table, same as used to fit a gdm. |
| geo | Similar to the gdm geo argument. The only difference is that the geo argument does not have a default in this function. |
| splines | Same as the gdm splines argument. |
| knots | Same as the gdm knots argument. |
| fullModelOnly | Set to TRUE to test only the full variable set. Set to FALSE to estimate model significance and variable importance and significance using matrix permutation and backward elimination. Default is FALSE. |
| nPerm | Number of permutations to use to estimate p-values. Default is 50. |
| parallel | Whether or not to run the matrix permutations and model fitting in parallel. Parallel processing is highly recommended when either (i) the nPerms argument is large (>100) or (ii) a large number of site-pairs (and or variables) are being used in model fitting (note computation demand can be reduced using subsampling - see next arguments). The default is FALSE. |
| cores | When the parallel argument is set to TRUE, the number of cores to be registered for parallel processing. Must be <= the number of cores in the machine running the function. |
| sampleSites | The fraction (0-1, though a value of 0 would be silly, wouldn't it?) of *sites to retain* from the full site-pair table. If less than 1, this argument will completely remove a fraction of sites such that they are not used in the permutation routines. |
| sampleSitePairs | The fraction (0-1) of *site-pairs (i.e., rows) to retain* from the full site-pair table - in other words, all sites will be used in the permutation routines (assuming sampleSites = 1), but not all *site-pair combinations*. In the case where both the sampleSites and the sampleSitePairs argument have values less than 1, sites first will be removed using the sampleSites argument, followed by removal of site-pairs using the sampleSitePairs argument. Note that the number of site-pairs removed is based on the fraction of the resulting site-pair table after sites have been removed, not on the size of the full site-pair table. |
| outFile | An optional character string to write the object returned by the function to disk as an .RData object (".RData"" is not required as part of the file name). The .RData object will contain a single list with the name of "outObject". The default is NULL, meaning that no file will be written. |

## Details

To test model significance, first a "full model" is fit using un-permuted environmental data. Next, the environmental data are permuted nPerm times (by randomizing the order of the rows) and a

GDM is fit to each permuted table. Model significance is determined by comparing the deviance explained by GDM fit to the un-permuted table to the distribution of deviance explained values from GDM fit to the nPerm permuted tables. To assess variable significance, this process is repeated for each predictor individually (i.e., only the data for the variable being tested is permuted rather than the entire environmental table). Variable importance is quantified as the percent change in deviance explained between a model fit with and without that variable (technically speaking, with the variable permuted and un-permuted). If fullModelOnly=FALSE, this process continues by then permutating the site-pair table nPerm times, but removing one variable at a time and reassessing variable importance and significance. At each step, the least important variable is dropped (backward elimination) and the process continues until all non-significant predictors are removed.

## Value

A list of four tables. The first table summarizes full model deviance, percent deviance explained by the full model, the p-value of the full model, and the number of permutations used to calculate the statistics for each fitted model (i.e., the full model and each model with variables removed in succession during the backward elimination procedure if fullModelOnly=F). The remaining three tables summarize (1) variable importance, (2) variable significance, and (3) the number of permutations used to calculate the statistics for that model, which is provided because some GDMs may fail to fit for some permutations / variable combinations and you might want to know how many permutations were used when calculating statistics. Or maybe you don't, you decide.

Variable importance is measured as the percent change in deviance explained by the full model and the deviance explained by a model fit with that variable permuted. Significance is estimated using the bootstrapped p-value when the variable has been permuted. For most cases, the number of permutations will equal the nPerm argument. However, the value may be less should any of the permutations fail to fit.

If fullModelOnly=T, the tables will have values only in the first column and NAs elsewhere.

NOTE: In some cases, GDM may fail to fit if there is a weak relationship between the response and predictors (e.g., when an important variable is removed). Such cases are indicated by -9999 values in the variable importance, variable significance, and number of permutations tables.

## Author(s)

Karel Mokany and Matt Fitzpatrick

## References

Ferrier S, Manion G, Elith J, Richardson, K (2007) Using generalized dissimilarity modelling to analyse and predict patterns of beta diversity in regional biodiversity assessment. *Diversity & Distributions* 13, 252-264.

Fitzpatrick, MC, Sanders NJ, Ferrier S, Longino JT, Weiser MD, and RR Dunn. 2011. Forecasting the Future of Biodiversity: a Test of Single- and Multi-Species Models for Ants in North America. *Ecography* 34: 836-47.

## Examples

```
##fit table environmental data
##set up site-pair table using the southwest data set
```

```
sppData <- southwest[c(1,2,13,14)]
envTab <- southwest[c(2:ncol(southwest))]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat",
sppColumn="species", siteColumn="site", predData=envTab)

## not run
#modTest <- gdm.varImp(sitePairTab, geo=T, nPerm=50, parallel=T, cores=10)
#barplot(sort(modTest[[2]][,1], decreasing=T))
```

---

`gdmDissim` *An example biological dissimilarity matrix*

---

## Description

Pairwise Bray-Curtis dissimilarity calculated using the species occurrence data from the `southwest` data set.

## Usage

```
gdmDissim
```

## Format

A data frame with 94 rows and 94 columns:

---

`isplineExtract` *Extract I-spline Values From a Fitted Generalized Dissimilarity Model.*

---

## Description

Extracts the I-spline values from a gdm object. There is one I-spline for each predictor that has at least one non-zero coefficient in the fitted model.

## Usage

```
isplineExtract(model)
```

## Arguments

| | |
|---|---|
| `model` | A gdm object from `gdm`. |

## Value

A list with two items. The first item contains the x-values (actual values of the predictors) of the I-splines and the second item contains the y-values (partial ecological distances) of the fitted I-splines.

## References

Ferrier S, Manion G, Elith J, Richardson, K (2007) Using generalized dissimilarity modelling to analyse and predict patterns of beta diversity in regional biodiversity assessment. *Diversity & Distributions* 13, 252-264.

Fitzpatrick MC, Sanders NJ, Normand S, Svenning J-C, Ferrier S, Gove AD, Dunn RR (2013). Environmental and historical imprints on beta diversity: insights from variation in rates of species turnover along gradients. Proceedings of the Royal Society: Series B 280, art. 1768

## Examples

```
##set up site-pair table using the southwest data set
sppData <- southwest[, c(1,2,14,13)]
envTab <- southwest[, c(2:ncol(southwest))]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat", sppColumn="species",
                              siteColumn="site", predData=envTab)

##create GDM
gdmMod <- gdm(sitePairTab, geo=TRUE)

##extracts splines
exSplines <- isplineExtract(gdmMod)

##plot spline(s)
#spline for winter precip (bio19)
plot(exSplines[[1]][,"bio19"], exSplines[[2]][,"bio19"], type="l",
     lwd=3, xlab="Winter precipitation (mm)", ylab="Partial Ecological Distance")
```

---

`plot.gdm` *Plot Model Fit and I-splines from a Fitted Generalized Dissimilarity Model.*

---

## Description

plot is used to plot the I-splines and fit of a generalized dissimilarity model created using the gdm function.

## Usage

```
## S3 method for class 'gdm'
plot(x, plot.layout = c(2, 2), plot.color = "blue",
  plot.linewidth = 2, include.rug = FALSE, rug.sitepair = NULL, ...)
```

## Arguments

| | |
|---|---|
| x | A gdm model object returned from gdm. |

| | |
|---|---|
| plot.layout | This argument specifies the row and column layout for the plots, including: (1) a single page plot of observed response data against the raw linear predictor (ecological distance) from the model, and (2) a single page plot of the observed response against the predicted response from the model, i.e. after applying the link function, 1.0 - exp(-y), to the linear predictor, and (3) the I-splines fitted to the individual predictors. Default is 2 rows by 2 columns. To produce one predictor plot per page set plot.layout to c(1,1). The first two model plots are always produced on a single page each and therefore the layout parameter affects only the layout of the I-spline plots for those predictors that featured in the model fitting process (i.e., predictors with all-zero I-spline coefficients are not plotted). |
| plot.color | Color of the data points that are plotted for the overall plots. |
| plot.linewidth | The line width for the regression line over-plotted in the two overall plots to optimize the display of the line over the data points. |
| include.rug | Whether or not to include a rug plot of the predictor values used to fit the gdm in the I-spline plots. When set to TRUE, a s ite-pair table must be supplied for the rug.sitepair argument. Default is FALSE. |
| rug.sitepair | A site-pair table used to add a rug plot of the predictor values used to fit the gdm in the I-spline plots. This should be the same site-pair table used to fit the gdm model being plotted. The function does not check whether the supplied site-pair table matches that used in model fitting. |
| ... | Ignored. |

## Value

plot returns NULL. Use summary.gdm to obtain a synopsis of the model object.

## References

Ferrier S, Manion G, Elith J, Richardson, K (2007) Using generalized dissimilarity modelling to analyse and predict patterns of beta diversity in regional biodiversity assessment. *Diversity & Distributions* 13:252-264.

## See Also

isplineExtract

## Examples

```
##set up site-pair table using the southwest data set
sppData <- southwest[c(1,2,13,14)]
envTab <- southwest[c(2:ncol(southwest))]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat",
                              sppColumn="species", siteColumn="site",
                              predData=envTab)

##create GDM
gdmMod <- gdm(sitePairTab, geo=TRUE)

##plot GDM
```

```
plot(gdmMod, plot.layout=c(3,3))
```

---

`plotUncertainty` *Plot I-splines With Error Bands Using Bootstrapping.*

---

## Description

This function estimates uncertainty in the fitted I-splines by fitting many GDMs using a subsample of the data. The function can run in parallel on multicore machines to reduce computation time (recommended for large number of iterations). I-spline plots with error bands (+/- one standard deviation) are produced showing (1) the variance of I-spline coefficients and (2) a rug plot indicating how sites used in model fitting are distributed along each gradient. Function result optionally can be saved to disk as a csv for custom plotting, etc. The result output table will have 6 columns per predictor, three each for the x and y values containing the lower bound, full model, and upper bound.

## Usage

```
plotUncertainty(spTable, sampleSites, bsIters, geo=FALSE,
splines=NULL, knots=NULL, splineCol="blue", errCol="grey80",
plot.linewidth=2.0, plot.layout=c(2,2), parallel=FALSE, cores=2, save=FALSE,
fileName="gdm.plotUncertainy.csv")
```

## Arguments

| | |
|---|---|
| `spTable` | A site-pair table, same as used to fit a `gdm`. |
| `sampleSites` | The fraction (0-1) of sites to retain from the full site-pair table when subsampling. |
| `bsIters` | The number of bootstrap iterations to perform. |
| `geo` | Same as the `gdm` geo argument. |
| `splines` | Same as the `gdm` splines argument. |
| `knots` | Same as the `gdm` knots argument. |
| `splineCol` | The color of the plotted mean spline. The default is "blue". |
| `errCol` | The color of shading for the error bands (+/- one standard deviation around the mean line). The default is "grey80". |
| `plot.linewidth` | The line width of the plotted mean spline line. The default is 2. |
| `plot.layout` | Same as the `plot.gdm` plot.layout argument. |
| `parallel` | Perform the uncertainty assessment using multiple cores? Default = FALSE. |
| `cores` | When the parallel argument is set to TRUE, the number of cores to be registered for the foreach loop. Must be <= the number of cores in the machine running the function. |
| `save` | Save the function result (e.g., for custom plotting)? Default=FALSE. |
| `fileName` | Name of the csv file to save the data frame that contains the function result. Default = gdm.plotUncertainy.csv. Ignored if save=FALSE. |

## Value

plotUncertainty returns NULL. Saves a csv to disk if save=TRUE.

## References

Shryock, D. F., C. A. Havrilla, L. A. DeFalco, T. C. Esque, N. A. Custer, and T. E. Wood. 2015. Landscape genomics of *Sphaeralcea ambigua* in the Mojave Desert: a multivariate, spatially-explicit approach to guide ecological restoration. *Conservation Genetics* 16:1303-1317.

## See Also

plot.gdm, formatsitepair, subsample.sitepair

## Examples

```
##set up site-pair table using the southwest data set
sppData <- southwest[c(1,2,13,14)]
envTab <- southwest[c(2:ncol(southwest))]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat",
                              sppColumn="species", siteColumn="site", predData=envTab)

##plot GDM uncertainty using one core
#not run
#plotUncertainty(sitePairTab, sampleSites=0.70, bsIters=5, geo=TRUE, plot.layout=c(3,3))

##plot GDM uncertainty in parallel
#not run
#plotUncertainty(sitePairTab, sampleSites=0.70, bsIters=50, geo=TRUE, plot.layout=c(3,3),
                 #parallel=T, cores=10)
```

---

# predict.gdm *Predict Biological Dissimilarities Between Sites or Times Using a Fitted Generalized Dissimilarity Model*

---

## Description

This function predicts biological distances between sites or times using a model object returned from gdm. Predictions between site pairs require a data frame containing the values of predictors for pairs of locations, formatted as follows: distance, weights, s1.X, s1.Y, s2.X, s2.Y, s1.Pred1, s1.Pred2, ..., s1.PredN, s2.Pred1, s2.Pred2, ..., s2.PredN, ..., Predictions of biological change through time require two raster stacks or bricks for environmental conditions at two time periods, each with a layer for each environmental predictor in the fitted model.

## Usage

```
## S3 method for class 'gdm'
predict(object, data, time=FALSE, predRasts=NULL, ...)
```

## Arguments

| | |
|---|---|
| object | A gdm model object resulting from a call to gdm. |
| data | Either a data frame containing the values of predictors for pairs of sites, in the same format and structure as used to fit the model using gdm or a raster stack if a prediction of biological change through time is needed.<br>For a data frame, the first two columns - distance and weights - are required by the function but are not used in the prediction and can therefore be filled with dummy data (e.g. all zeros). If geo is TRUE, then the s1.X, s1.Y and s2.X, s2.Y columns will be used for calculating the geographical distance between each site for inclusion of the geographic predictor term into the GDM model. If geo is FALSE, then the s1.X, s1.Y, s2.X and s2.Y data columns are ignored. However these columns are still REQUIRED and can be filled with dummy data (e.g. all zeroes). The remaining columns are for N predictors for Site 1 and followed by N predictors for Site 2. The order of the columns must match those in the site-pair table used to fit the model.<br>A raster stack should be provided only when time=T and should contain one layer for each environmental predictor in the same order as the columns in the site-pair table used to fit the model. |
| time | TRUE/FALSE: Is the model prediction for biological change through time? |
| predRasts | A raster stack characterizing environmental conditions for a different time in the past or future, with the same extent, resolution, and layer order as the data object. Required only if time=T. |
| ... | Ignored. |

## Value

predict returns either a response vector with the same length as the number of rows in the input data frame or a raster depicting change through time across the study region.

## See Also

gdm.transform

## Examples

```
##set up site-pair table using the southwest data set
sppData <- southwest[, c(1,2,14,13)]
envTab <- southwest[, c(2:ncol(southwest))]

# remove soils (no rasters for these)
envTab <- envTab[,-c(2:6)]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat", sppColumn="species",
                              siteColumn="site", predData=envTab)

gdmMod <- gdm(sitePairTab, geo=TRUE)

##predict GDM
predDiss <- predict(gdmMod, sitePairTab)
```

```
##time example
rastFile <- system.file("./extdata/swBioclims.grd", package="gdm")
envRast <- raster::stack(rastFile)

##make some fake climate change data
futRasts <- envRast
##reduce winter precipitation by 25%
futRasts[[3]] <- futRasts[[3]]*0.75

timePred <- predict(gdmMod, envRast, time=TRUE, predRasts=futRasts)
raster::plot(timePred)
```

---

`southwest` *Species and Environmental Data from Southwestern Australia.*

---

## Description

A data set containing species occurrence and associated environmental data at 94 sites in southwestern Australia.

## Usage

```
southwest
```

## Format

A data frame with 29364 rows and 14 variables:

**species** species name

**site** site name

**awcA** plant-available water capacity in soil horizon A

**phTotal** soil pH

**sandA** percent sand content in soil horizon A

**shcA** saturated hydraulic conductivity in soil horizon A

**solumDepth** soil depth to unweathered parent material

**bio5** maximum temperature of the coldest month

**bio6** minimum temperature of the coldest month

**bio15** precipitation seasonality

**bio18** precipitation of warmest quarter

**bio19** precipitation of coldest quarter

**Lat** latitude

**Long** longitude

`subsample.sitepair` *Remove Sites at Random from a Site-Pair Table*

## Description

Randomly selects a number of sites from a given site-pair table and removes them from the site-pair table. It will remove all instances of the sites randomly selected to be removed in both s1 and s2 positions.

## Usage

```
subsample.sitepair(spTable, sampleSites)
```

## Arguments

| | |
|---|---|
| `spTable` | A site-pair table, same as used to fit a `gdm`. |
| `sampleSites` | The fraction (0-1, though a value of 0 would be silly, wouldn't it?) of *sites to retain* from the full site-pair table. If less than 1, this argument will completely remove a fraction of sites such that they are not used in the permutation routines. |

## Value

A site-pair table, such as one created by `formatsitepair`, ideally smaller than the one given. In the very rare case where the function determines not to remove any sites, or should the sampleSites argument be 1, then the function will return the full site-pair table.

## Note

This function removes sites, not just site-pairs (rows) from the site-pair table. This function is called from several of the other functions within the gdm package, including the `plotUncertainty` and `gdm.varImp` functions, for the purposes of subsampling the sites in the site-pair table.

## See Also

`formatsitepair`

## Examples

```
##set up site-pair table using the southwest data set
sppData <- southwest[c(1,2,13,14)]
envTab <- southwest[c(2:ncol(southwest))]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat", sppColumn="species",
                              siteColumn="site", predData=envTab)

subsample.sitepair(sitePairTab, sampleSites=0.7)
```

summary.gdm *Summarize a Fitted Generalized Dissimilarity Model*

## Description

This function summarizes the gdm model object returned from gdm.

## Usage

```
## S3 method for class 'gdm'
summary(object, ...)
```

## Arguments

| | |
|---|---|
| object | A gdm model object resulting from a call to gdm. |
| ... | Ignored. |

## Value

summary prints its output to the R Console window and returns no value.

## See Also

gdm

## Examples

```
##set up site-pair table using the southwest data set
sppData <- southwest[, c(1,2,14,13)]
envTab <- southwest[, c(2:ncol(southwest))]
sitePairTab <- formatsitepair(sppData, 2, XColumn="Long", YColumn="Lat", sppColumn="species",
                              siteColumn="site", predData=envTab)

##create GDM
gdmMod <- gdm(sitePairTab, geo=TRUE)

##summary of GDM
summary(gdmMod)
```

# Index