Using the R package `gdm` to analyze and map biodiversity patterns

Matt Fitzpatrick & Matthew Lisk
March 28, 2016

Contents

1 Introduction 1

2 gdm Basics 2
   2.1 Example Data ................................. 2
   2.2 Preparing site-pair tables ......................... 3
   2.3 Dealing with biases associated with presence-only data ......................... 4

3 gdm analysis 5
   3.1 gdm plots ..................................... 6
   3.2 gdm predictions ................................ 8
   3.3 Predicting biological distances between sites ................................ 8
   3.4 Predicting biological change through time ................................ 9
   3.5 Transforming predictors and visualizing biological patterns .................. 10

References 12

1 Introduction

The R package `gdm` implements Generalized Dissimilarity Modeling (Ferrier S et al. 2007) to analyze and map spatial patterns of biodiversity. GDM models biological variation as a function of environment and geography using distance matrices – specifically by relating dissimilarity in species composition (or any biological distance, notably genetic (Fitzpatrick and Keller 2015), phylogenetic (Rosauer et al. 2014), or function/trait differences (Thomassen et al. 2010)) between sites to how much sites differ in their environmental conditions (environmental distance) and how isolated they are from one another (geographical distance). Geographical distance measures can be simple Euclidean separation or more complex measures of isolation, such as least cost paths or resistance distance. See (Ferrier S et al. 2007) for strategies for including categorical variables. This vignette demonstrates how to implement and interpret generalized dissimilarity models in the context of analyzing and mapping species-level patterns. Examples for modeling genetic variation within species will be included in a future update, though the techniques are largely identical to the species-level case (only the input biological data differ and the interpretations of variation in species vs. genetic composition).
2  gdm Basics

The gdm package is available on CRAN, development versions are available on GitHub. This vignette covers only those functions currently available from the CRAN version.

```r
# installation of the package from CRAN
#install.packages("gdm")

# installation from GitHub
#library(devtools)
#install_github("fitzLab-AL/GDM")

library(gdm)
```

2.1 Example Data

The biological data provided in the gdm package are occurrence data for plants from southwest Australia (Fitzpatrick et al. 2013). Of the original data, a subset of 26 species were selected to be included with the package. The full datasets are available from Dryad. The environmental data include both climatic and soils variables, with the climate data being supplied as both tabular (at sites only) and raster formats (all of southwest Australia).

GDM can use several data formats as input. Most common are site-by-species tables (sites in rows, species across columns) for the response and site-by-environment tables (sites in rows, predictors across columns) as the predictors, though distance matrices and rasters are also accommodated. For example purposes, a biological dissimilarity matrix is provided to showcase the use of a pre-formatted distance matrix (table type 3) as the response variable, though note that distance matrices can also be used as predictors (e.g., to model compositional variation in one group as a function of compositional variation in another group, (Jones et al. 2013)).

The first example uses an x-y species list where there is one row per species record rather than per site - similar to what would be obtained from online databases such as GBIF. Note that the rows and their order must match in the biological and environmental data frames and must not include NAs. In this example both the species and environmental data are provided in the same table, which are then indexed to create two tables, one for the species data and the other for the environmental data.

```r
# reads in example input data
load(system.file("./data/gdm.RData", package="gdm"))
# columns 3-7 are soils variables, remainder are climate
gdmExpData[1:3,]
```

```
##        species site awCA phTotal sandA shcA solumDepth bio5
## 1      spp1 1066  14.4725    71.3250  178.865   875.1725  31.43824  
## 2      spp1 1026  16.2575    105.840  105.840   892.4925  33.14412  
## 3      spp1 1025  23.1375    88.355   88.355   892.2275  32.84000  
##      bio6 bio15 bio18 bio19 Lat   Long
## 1  5.058823  40.38235 132.6471 -32.99425 118.7573  
## 2  4.852941  48.20588 140.2941  118.3495  
## 3  4.817143  53.88571 145.0571  117.8260  
```

```r
# get columns with xy, site ID, and species data
sppTab <- gdmExpData[, c("species", "site", "Lat", "Long")]
```
### # get columns with env. data and xy-coordinates

```r
envTab <- gdmExpData[, c(2:ncol(gdmExpData))]
```

2.2 Preparing site-pair tables

The initial step in fitting a generalized dissimilarity model is to combine the biological and environmental data into “site-pair” format. This can be accomplished using the `formatsitepair` function. Each row in the resulting site-pair table contains a biological distance measure in the first column (the default is Bray-Curtis distance though any measure scaled between 0-1 will work). The second column contains the weight to be assigned to each data point in model fitting (defaults to 1, but can be customized by the user or can be scaled to site richness, see below). The remaining columns are the environmental values at a site (s1) and those at a second site (s2) making up a site pair. Subsequent rows repeat this pattern until all possible site pairs are represented and such that pairwise distances between all sites can be calculated and used as predictors. While the site-pair table format can produce extremely large data frames and contain numerous repeat values, it also allows great flexibility. Most notably, individual site pairs easily can be excluded from model fitting.

A properly formatted site-pair table will have at least six columns (distance, weights, s1.xCoord, s1.yCoord, s2.xCoord, s2.yCoord) and possibly more depending upon how many predictor variables are included. See `?formatsitepair` and `?gdm` for more details.

### # x-y species list example

```r
gdmTab <- formatsitepair(sppTab, bioFormat=2, XColumn="Long", YColumn="Lat", 
                         sppColumn="species", siteColumn="site", predData=envTab)
gdmTab[1:3,]
```

```r
## distance weights s1.xCoord s1.yCoord s2.xCoord s2.yCoord s1.awcA
## 132 0.4485981 1 115.057 -29.40472 115.5677 -29.46599 23.0101
## 132.1 0.7575758 1 115.057 -29.40472 116.0789 -29.52556 23.0101
## 132.2 0.8939394 1 115.057 -29.40472 116.5907 -29.58342 23.0101
## s1.phTotal s1.sandA s1.shcA s1.solumDepth s1.bio5 s1.bio6 s1.bio15
## 132 480.3266 83.99326 477.5656 1129.933 34.668 8.908 86.64
## 132.1 480.3266 83.99326 477.5656 1129.933 34.668 8.908 86.64
## 132.2 480.3266 83.99326 477.5656 1129.933 34.668 8.908 86.64
## s1.bio18 s1.bio19 s2.awcA s2.phTotal s2.sandA s2.shcA s2.solumDepth
## 132 35.50571 7.448572 22.3925 494.1225 76.6900 357.7225 1183.9025
## 132.1 36.05000 6.605882 17.0975 415.1275 70.0175 112.4800 985.5300
## 132.2 36.18750 6.131250 17.0300 333.4400 71.5950 165.7250 956.5425
## s2.bio5 s2.bio6 s2.bio15 s2.bio18 s2.bio19
## 132 35.50571 7.448572 22.3925 494.1225 76.6900
## 132.1 36.05000 6.605882 17.0975 415.1275 70.0175
## 132.2 36.18750 6.131250 17.0300 333.4400
```

### # Biological distance matrix example

```r
dim(gdmDissim)
```

```r
## [1] 94 94

gdmDissim[1:5, 1:5]
```

```r
##             V1          V2          V3          V4          V5
## 1 0.00000000 0.8181818 1.00000000 0.50000000 0.7500000
```
## 2.3 Dealing with biases associated with presence-only data

The ideal biological data for fitting a GDM are occurrence records (presence-absence or abundance) from a network of sites where all species (from one or more taxonomic groups) have been intensively sampled such that compositional dissimilarity can be reliably estimated between sites. However most species data are collected as part of ad hoc surveys and are presence-only. Under these circumstances, there is no systematic surveying and no sites per se, but rather grid cells with some number of occurrence records depending on the number of species observed, with many grid cells having none, a few, or even a single species record. When under-sampled sites are used to calculate compositional dissimilarity, erroneously high values will result, which will bias the model.

The `formatsitepair` function provides a few options for dealing with this potential bias, including (i) weighting sites relative to the number of species observed (`weightType="richness"`), (ii) removing sites with few species (e.g., `speciesFilter=10`) or (iii) both. Decisions regarding which approach to use will depend on the nature of the data and study system. See (Ferrier S et al. 2007) for further discussion.

```r
# weight by site richness
gdmTab.rw <- formatsitepair(sppTab, bioFormat=2, XColumn="Long", YColumn="Lat", 
                          sppColumn="species", siteColumn="site", 
                          predData=envTab, weightType="richness")
```

# weights based on richness (number of species records)
gdmTab.rw$weights[1:5]
3  gdm analysis

GDM is a nonlinear extension of permutational matrix regression that uses flexible splines and a GLM to accommodate two types of nonlinearity common in ecological datasets: (1) variation in the rate of compositional turnover (non-stationarity) along environmental gradients, and (2) the curvilinear relationship between biological distance and environmental and geographical distance.

The function gdm fits generalized dissimilarity models and is simple to use once the biological and predictor data have been formatted to a site-pair table. In addition to specifying whether or not the model should be fit with geographical distance as a predictor variable, the user can also specify (i) the number of I-spline basis functions (the default is three, with larger values producing more complex splines) and (ii) the locations of “knots” along the splines (defaults 0 (minimum), 50 (median), and 100 (maximum) quantiles when three I-spline basis functions are used). The effects of altering the number of splines and knot locations has not been systematically explored.

gdm.1 <- gdm(gdmTab, geo=T)

The summary function provides an overview of the model, including deviance explained and the values of the coefficients for the I-spline for each predictor variable. Variables with all coefficients=0 have no relationship with the biological pattern. A shorter summary can be obtained using str.

#summary(gdm.1)
str(gdm.1)
3.1 gdm plots

The fitted splines represent one of the most informative outputs from gdm, which also can be used to transform and map environmental variables such that they best represent biological patterns. The fitted model and I-splines can be viewed using the plot function, which produces a multi-panel plot that includes: (i) the fitted relationship between predicted ecological distance and observed compositional dissimilarity; (ii) predicted versus observed biological distance, and (iii) each I-spline with at least one non-zero coefficient (in the provided example bio18 is not plotted because all three coefficients equaled zero).

The maximum height of each spline indicates the magnitude of total biological change along that gradient and thereby corresponds to the relative importance of that predictor in contributing to biological turnover while holding all other variables constant (i.e., is a partial ecological distance). The spline’s shape indicates how the rate of biological change varies with position along that gradient. Thus, the splines provide insight into the total magnitude of biological change as a function of each gradient and where along each gradient those changes are most pronounced. In this example, compositional turnover is greatest along gradients of bio19 (winter precipitation) and phTotal (soil phosphorus) and most rapid near the low ends of these gradients.

```r
length(gdm.1$predictors) # get idea of number of panels
## [1] 11

plot(gdm.1, plot.layout=c(4,3))

To allow easy customization of I-spline plots, the isplineExtract function will extract the plotted values for each I-spline.

gdm.1.splineDat <- isplineExtract(gdm.1)
str(gdm.1.splineDat)
## List of 2
## $ x: num [1:200, 1:11] 0.452 0.483 0.513 0.544 0.574 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$: NULL
## .. ..$ : chr [1:11] "Geographic" "awcA" "phTotal" "sandA" ...
## $ y: num [1:200, 1:11] 0 0.00045 0.00095 0.0015 0.0021 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$: NULL
## .. ..$ : chr [1:11] "Geographic" "awcA" "phTotal" "sandA" ...

plot(gdm.1.splineDat$x[,"Geographic"], gdm.1.splineDat$y[,"Geographic"], lwd=3, type="l", xlab="Geographic distance", ylab="Partial ecological distance")
```
Figure 1: The fitted model (first two panels) and I-splines (remaining panels).
3.2 gdm predictions

The I-splines provide an indication of how species composition (or other biological measure) changes along each environmental gradient. Beyond these insights, a fitted model also can be used to (i) predict biological dissimilarity between site pairs in space or between times using the `predict` function and (ii) transform the predictor variables from their arbitrary environmental scales to a common biological importance scale using the `transform` function.

The examples show predictions between site pairs and through time, and transformation of both tabular and raster data. For the raster example, the transformed layers are used to map spatial patterns of biodiversity.

3.3 Predicting biological distances between sites

The `predict` function requires a site-pair table in the same format as that used to fit the model. For demonstration purposes, we use the same table as that used to fit the model, though predictions to new sites (or times) can be made as well assuming the same set of environmental/spatial predictors are available at those locations (or times).

```r
gdm.1.pred <- predict(gdm.1, gdmTab)
head(gdm.1.pred)

## [1] 0.4720423 0.7133571 0.8710175 0.8534788 0.9777208 0.3996694
```

```r
plot(gdmTab$distance, gdm.1.pred, xlab="Observed dissimilarity", ylab="Predicted dissimilarity", xlim=c(0,1), ylim=c(0,1), pch=20, col=rgb(0,0,1,0.5))
lines(c(-1,2), c(-1,2))
```
3.4 Predicting biological change through time

The predict function can be used to make predictions across time (Blois et al. 2013), for example, under climate change scenarios to estimate the magnitude of expected change in biological composition in response to environmental change (Fitzpatrick et al. 2011). In this case, rasters must be provided for two time periods of interest.

```r
# fit a new gdm using a table with climate data only (to match rasters)
gdm.rast <- gdm(gdmTab.rast, geo=T)

# make some fake climate change data
futRasts <- envRast
# reduce winter precipitation by 25% & increase temps
futRasts[[3]] <- futRasts[[3]]*0.75
futRasts[[4]] <- futRasts[[4]]+2
futRasts[[5]] <- futRasts[[5]]+3

timePred <- predict(gdm.rast, envRast, time=T, predRasts=futRasts)
plot(timePred)
```
3.5 Transforming predictors and visualizing biological patterns

3.5.1 Transforming from environmental space to biological space

Predictor data to be transformed can be in one of two formats: a data frame with columns in the order of: X, Y, var1, var2, ..., varN, or a raster stack or brick with one layer per predictor. Beyond the x- and y-columns in the data frame, the order of the predictor data (columns for tabular data, layers for rasters) must be the same as that in the site-pair table used in model fitting. If the model was fit without using geographical distance, then the x- and y-columns of the data frame should not be included in the predictor data. If the model was fit with geographical distance and raster data are provided to the transform function, there is no need to provide x- or y-raster layers as these will be generated automatically. However, the character names of the x- and y-coordinates (e.g., “Lat” and “Long”) used to fit the model need to be provided.

```r
# reordering environmental data to create table for transformation
envTrans <- envTab[, c(13,12,2:11)]  # same order as gdmTab
envTrans[1:3,]
```

```
## Long   Lat  awcA phTotal sandA shcA solumDepth bio5 bio6 bio15 bio18 bio19
## 1 118.7573 -32.99425 14.4725  546.1800  71.3250 178.865  875.1725  31.43824
## 2 118.3495 -32.04285 16.2575  470.9950  68.8975 105.840  928.4925  33.14412
## 3 117.8260 -31.99067 23.1375  459.7425  71.4700  88.355  892.2275  32.84000
## 1 5.058823  40.38235  0 132.6471
## 2 4.852941  48.20588  0 140.2941
## 3 4.817143  53.88571  43 145.0571
```
tabTrans <- gdm.transform(gdm.1, envTrans)
# now scaled to biological importance

## Long Lat awcA phTotal sandA shcA
## [1,] 0.2226638 0.1036539 0.000000000 1.1437240 0.09098455 0.0349996799
## [2,] 0.1985913 0.1598227 0.000000000 0.9902375 0.08618790 0.0025608921
## [3,] 0.1676795 0.1629031 0.0005822033 0.9560643 0.09114130 0.0003213945
## solumDepth bio5 bio6 bio15 bio18 bio19
## [1,] 0.5407158 0.3488653 0.10210250 0.01846392 0 0.5102596
## [2,] 0.6269481 0.4892084 0.08073046 0.02528373 0 0.6754013
## [3,] 0.5726173 0.4650542 0.07620206 0.02711470 0 0.7637484

# transform climate rasters & plot pattern
rastTrans <- gdm.transform(gdm.rast, envRast)
#plot(rastTrans)

3.5.2 Visualizing multi-dimensional biological patterns

Pairwise site biological distances are difficult to visualize. However, if the transform function is applied to rasters, the resulting multi-dimensional biological space can be mapped to reveal biological patterns in geographic space. Alternatively, a biplot can be used to depict where sites fall relative to each other in biological space and therefore how sites differ in predicted biological composition. In either case, the multi-dimensional biological space can be most effectively visualized by taking a PCA to reduce dimensionality and assigning the first three components to an RGB color palette.

rastDat <- na.omit(getValues(rastTrans))
#rastDat <- sampleRandom(rastTrans, 50000) # can use if rasters are large
pcaSamp <- prcomp(rastDat)

# note the use of the 'index' argument
pcaRast <- predict(rastTrans, 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

plotRGB(pcaRast, r=1, g=2, b=3)
Figure 5: Predicted spatial variation in plant species composition. Colors represent gradients in species composition derived from transformed environmental predictors. Locations with similar colors are expected to contain similar plant communities.

References


