SPECIES RESPONSE CURVES

Steven M. Holland

Department of Geology, University of Georgia, Athens, GA 30602-2501

June 2014

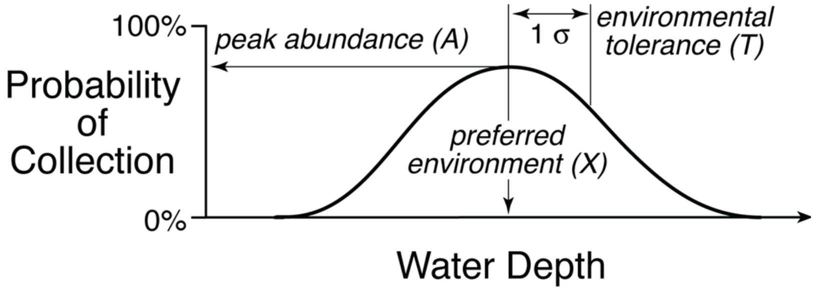
Introduction

Species live on environmental gradients, and we often would like to describe their occurrence along those gradients. The distribution of a species along a gradient is known as a species response curve. In modern environments, many species display a unimodal distribution along an environmental gradient, and this distribution is often symmetrical. Such a distribution can easily be modeled as a normal distribution, defined with three parameters.

The **optimum** measures where a species is most likely to be found, that is the peak of a distribution. For any distribution, the optimum is equivalent to the mode of the distribution. For a symmetrical distribution, the optimum is also equivalent to the mean of the distribution. In our work, we have often called the optimum the **preferred environment (PE)** or, where we were dealing specifically with a water depth gradient, the **preferred depth (PD)**.

The **tolerance** measures the ability of a species to live in non-optimal environments, that is, it describes the spread or width of a distribution. For a symmetrical distribution, this is equivalent to the standard deviation of the distribution. We have often called tolerance by the terms **environmental tolerance (ET)** or **depth tolerance (DT)**.

The **maximum** measures how abundant a species is at its optimum, that is the height of the abundance distribution at the optimum. The maximum could be described in terms of abundance, but it can also be described as a probability, such as the probability of encountering the species at the optimum. We have often used the term **peak abundance (PA)** for the maximum.


100%
peak abundance (A)
1 σ
environmental tolerance (T)
Probability of Collection
preferred environment (X)
0%
Water Depth

Although many species distributions can be described by a normal distribution with these three parameters, some species have different distributions. The most common alternative is an asymmetrical distribution, in which tail is much longer on one side of the optimum than the other. Asymmetrical distributions often arise when a species' optimum lies close to a fixed boundary, such as a marine species whose optimum lies in very shallow water, or a terrestrial species whose optimum lies near sea level. Bimodal or polymodal distributions also occur, but they are much rarer than symmetrical and asymmetrical unimodal distributions. These notes will focus on symmetrical unimodal distributions.

Computation

There are two main approaches to estimating optimum, tolerance, and maximum: weighted averaging and logistic regression. Both methods start with a standard species-abundance matrix, with species in columns, and samples in rows. Both methods also need the position along an environmental gradient for each sample. The gradient may be **direct**, such as water depth, salinity, or temperature, or the gradient may be **indirect**, such as the sample scores from an ordination method, such as detrended correspondence analysis or multidimensional scaling.

Weighted averaging

The weighted averaging method gets its name from the calculation of the optimum: the abundance-weighted average gradient position of every sample bearing the species. By weighting this average by abundance, the importance of a sample containing many more individuals of a species (and therefore, likely close to the optimum) counts proportionally more than a sample containing only a single occurrence of the species (which is likely far from the optimum).

In ordination methods such as detrended correspondence analysis and non-metric multidimensional scaling, species scores are calculated as abundance-weighted averages. If response curves are being calculated from these ordination methods, no additional calculations need to be made to obtain the optima - they are simply the species scores.

Tolerance is calculated by calculating the standard deviation of all gradient positions of the samples that contain the taxon. For ordination-based methods, tolerance is the standard deviation of the sample scores of samples bearing the taxon. If a species occurs over a limited set of gradient positions, this standard deviation will be small, and if a species occurs over a broad range of gradient positions, the standard deviation will reflect a large tolerance.

Maximum is estimated by first calculating the percent occurrence of a species in all samples that lie within one tolerance (i.e., standard deviation) of the optimum. This is multiplied by a a constant (approximately 1.169) that reflects the ratio of the peak height of a normal distribution to the average height of that distribution over the interval of the mean plus or minus one standard deviation.

Logistic regression

Logistic regression is used when the dependent variable (in this case, the probability of occurrence) is constrained to lie between 0 and 1. A Gaussian function can be fitted to the occurrence data using this function:

$$logit(p) = b_0 + b_1x + b_2x^2$$

The right-hand side is an equation for a parabola, where x is the position along a gradient. The left-hand side, known as the transfer function, converts this parabola to a Gaussian distribution. The function logit(p) is the log-odds of the probability of collection, defined as

$$logit(p) = log\left(\frac{p}{1-p}\right)$$

where p is the probability of collection of a taxon.

From this logistic regression and its three parameters, b0, b1, and b2, estimates of the optimum (O), tolerance (T), and maximum (M) can be calculated:

$$O = \frac{-b_1}{2b_2}$$

$$T = \frac{1}{\sqrt{-2b_2}}$$

$$M = e^{b_0+b_1O+b_2O^2}$$

Species response curves in R

To perform these methods, you will need an abundance matrix and a vector of gradient positions.

The abundance matrix should be the same as you would use in any ordination. Each column should contain the abundance of a single species, with each row reflecting a single sample. It will likely be useful to have species names in the first row of the data set and sample labels in the first column.

The vector of gradient positions should have the same length as the number of samples, and these gradient positions must be in the same order as the samples in the abundance matrix. If you use an ordination method to establish the gradients, the vector should be in the correct order.

For this tutorial, we will assume that you will want to use an ordination to establish the gradient. Although this will be illustrated here with detrended correspondence analysis (DCA), you can follow a similar approach with non-metric multidimensional scaling or any other ordination.

First, read in the data. This tutorial uses the frankfort data, which consists of Ordovician marine invertebrates from the Frankfort, Kentucky area (Holland and Patzkowsky 2004).

```
frank <- read.table(file='frankfort.txt', header=TRUE, row.-
names=1, sep=',')
```

Next, load the vegan library, which contains the useful data transformation function **decostand** and the DCA function **decorana**.

```
library(vegan)
```

Perform two data transformations. First standardize by row totals, which converts all the abundance counts to percentages, preventing sample size from dominating the ordination. Second, standardize by the maximum in each column, which sets the minimum value of every column to 0 and the maximum value to 1, in effect giving all species on the same weight.

```
frank.t1 <- decostand(frank, method='total')
frank.t2 <- decostand(frank.t1, method='max')
```

Perform a detrended correspondence analysis on the doubly transformed data.

```
frank.dca <- decorana(frank.t2)
```

It is helpful to extract the sample scores and species scores from the decorana output. We will focus only on axis 1, the most important source of variation in the data.

```
frank.sampscores1 <- scores(frank.dca, display="sites", choic-
es=1)

frank.specscores1 <- scores(frank.dca, display="species",
choices=1)
```

At this point, you have everything you need: the abundance matrix (**frank**), a vector of gradient positions of samples (**frank.sampscores1**), and a vector of gradient positions of species (**frank.specscores1**).

Weighted averaging

Getting a vector of optima (preferred environment, PE) is the easiest - it is just your vector of species scores.

```
PE <- frank.specscores1
```

Next are the values of tolerance (environmental tolerance, ET). This begins by creating a vector **ET** to hold the results. It then loops through the species. For each species, it creates a temporary vector (**occupiedPositions**), which will hold all of the gradient positions at which the species occurred. For that species, the method steps through each sample score and checks to see if the abundance of that species in that sample is greater than zero, meaning the species is present. If it is present in that sample, the gradient position (sample score) is added to `occupiedPositions`. After all samples have been checked, the standard deviation of the occupied gradient positions is calculated, which gives the tolerance for that species. This procedure is repeated for all species.

```
ET <- vector(length=length(PE))
for (species in 1:length(PE) {
  occupiedPositions <- vector(length=0)
  i <- 1
  for (sample in 1:length(frank.sampscores1) {
     if (frank[sample,species]>0) {
       occupiedPositions[i] <- frank.sampscores1[sample]
       i <- i+1
     }
  }
  ET[species] <- sd(occupiedPositions)
}
```

If you discover a way to vectorize this code and remove the loops, I'd appreciate knowing how you did it.

The last step is to calculate the maxima, which follows a similar pattern of looping through all of the species. First, a vector is created to hold all maxima (`PA`) for each species. For each species, variables are created to hold the total number of samples within one tolerance of the optimum for that species (`total`) and the number of those samples that contain that species (`present`). Both of these are initially set to 0. Each sample is then checked for that species. If the gradient position of that sample lies within the optimum plus or minus one tolerance for the species, `total` is incremented by 1, and the abundance is checked. If it is found, `present` is incremented by 1. The final line calculates the maxima as a percentage, with the correction factor.

```
PA <- vector(length=length(PE))
for (species in 1:length(frank.specscores1))
{
  present <- 0
  total <- 0
  for (sample in 1:length(frank.sampscores1)){
     if (abs(frank.sampscores1[sample]-
             frank.specscores1[species]) <= ET[species]) {
       total <- total + 1
       if (frank[sample, species] > 0) {
          present <- present + 1
       }
     }
  }
  PA[species] <- present/total * 100 * 1.168739
}
```

For convenience, all of this can be wrapped in a function that takes the abundance matrix, the sample score vector, and the species score vector as arguments. The function returns a data frame with columns for optimum (PE), tolerance (ET), and maximum (PA).

```
speciesResponseWeightedAveraging <- function(abundance,
 sampscores, specscores) {
   numSamples <- length(sampscores)
   numSpecies <- length(specscores)

   PE <- specscores

   ET <- vector(length=numSpecies)
   for (species in 1:numSpecies) {
     occupiedPositions <- vector(length=0)
     i <- 1
     for (sample in 1:numSamples) {
         if (abundance[sample, species]>0) {
           occupiedPositions[i] <- sampscores[sample]
           i <- i+1
         }
     }
     ET[species] <- sd(occupiedPositions)
   }

   PA <- vector(length=numSpecies)
   for (species in 1:numSpecies) {
     present <- 0
     total <- 0
     for (sample in 1:numSamples) {
         if (abs(sampscores[sample]-specscores[species])
              <= ET[species]) {
           total <- total + 1
           if (abundance[sample, species]>0) {
               present <- present + 1
           }
         }
     }
     PA[species] <- present/total * 100 * 1.168739
   }

   threeparams <- data.frame(PE=PE, ET=ET, PA=PA)
}
```

Call it like this, and assign the results so that they can be viewed.

```
params <- speciesResponseWeightedAveraging(frank, frank.samp-
scores1, frank.specscores1)
```

Logistic regression

Logistic regression is based simply on whether a species occurs at any of the gradient positions. For this method, you will need a vector of gradient positions (**`frank.sampscores1`**), and you will need a vector of whether the species occurs at that gradient position. To illustrate the method, we will use the common brachiopod *Hebertella* in the Frankfort data (column 13) and convert its abundances to presence/absence. This is done by finding every value that is greater than zero and replacing it with a 1, such that all values are now 0 (absent) or 1 (present).

```
species13 <- frank[, 13]
species13[species13 > 0] <- 1
```

For clarity, we will call the sample scores x, as they are the independent variable, and we will call the presence/absence data y, as they are the dependent variable.

```
    x <- frank.sampscores1
    y <- species13
```

The logistic regression is performed with **`glm()`** command and assign the results to an object.

```
    logitReg <- glm(y ~ x + I(x^2), family=binomial)
```

The **`glm()`** command will produce warnings if a regression could not be fit to the data, which happens in some cases.

For clarity, we will relabel the coefficients from the regression object to match the logistic equation given above.

```
    b0 <- logitReg$coefficients[1]
    b1 <- logitReg$coefficients[2]
    b2 <- logitReg$coefficients[3]
```

The optimum (**`opt`**), tolerance (**`tol`**), and maximum (**`pmax`**) are calculated as shown.

```
    opt <- (-b1)/(2*b2)
    tol <- 1 / sqrt(-2*b2)
    pmax <- 1 / (1 + exp(b1^2 / (4 * b2) - b0))
```

For convenience, this can be wrapped in a function to calculate the optimum, tolerance, and maximum for one species. It should be supplied with a data frame, in which column 1 contains the gradient positions, and column 2 contains the presence/absence data. The results are bound together with **`cbind()`** and returned as a matrix.

```
tBLparamsForOneSpecies <- function(theData) {
    x <- theData[ ,1]
    y <- theData[ ,2]

    logitReg <- glm(y ~ x + I(x^2), family=binomial)
```

```
        b0 <- logitReg$coefficients[1]
        b1 <- logitReg$coefficients[2]
        b2 <- logitReg$coefficients[3]

        opt <- (-b1)/(2*b2)
        tol <- 1 / sqrt(-2*b2)
        pmax <- 1 / (1 + exp(b1^2 / (4 * b2) - b0))

        theParams <- cbind(opt, tol, pmax)
        theParams
    }
```

More generally, you will likely want to calculate these parameters on an entire data set, not just one species. The following function does that. The first parameter is a vector of sample scores, and the second is the matrix of species abundances. In cases where values cannot be computed, two lines near the end of the function replace the missing values with **NaN** (not a number). If this replacement is necessary, you will see a warning message after issuing the command.

```
    tBLparamsForSpeciesMatrix <- function(sampleValues,
      abundanceMatrix) {
        numTaxa <- ncol(abundanceMatrix)

        opt <- vector(mode="numeric", length=numTaxa)
        tol <- vector(mode="numeric", length=numTaxa)
        pmax <- vector(mode="numeric", length=numTaxa)

        for (t in 1:numTaxa) {
              x <- sampleValues
              y <- abundanceMatrix[ ,t]
              y[y>0] <- 1

              logitReg <- glm(y ~ x + I(x^2), family=binomial)

              b0 <- logitReg$coefficients[1]
              b1 <- logitReg$coefficients[2]
              b2 <- logitReg$coefficients[3]

              opt[t] <- (-b1)/(2*b2)
              tol[t] <- 1 / sqrt(-2*b2)
              pmax[t] <- 1 / (1 + exp(b1^2 / (4 * b2) - b0))
        }

        tBLParams <- data.frame(opt, tol, pmax)
        tBLParams$opt[is.na(tBLParams$tol)] <- NaN
        tBLParams$pmax[is.na(tBLParams$tol)] <- NaN

        rownames(tBLParams) <- colnames(abundanceMatrix)
        tBLParams
    }
```

When calling this function, assign the results to an object so that you can use them. Each row corresponds to a species, in the same order as the columns of your abundance matrix. The columns correspond to optimum (opt), tolerance (tol), and maximum (pmax).

```
    params <- tBLparamsForSpeciesMatrix(frank.sampscores1, frank)
```

References

Coudun, C., and J. C. Gegout, 2006. The derivation of species response curves with Gaussian logistic regression is sensitive to sampling intensity and curve characteristics. Ecological Modeling 199:164–175.

Holland, S.M., 1995. The stratigraphic distribution of fossils. Paleobiology 21: 92–109.

Holland, S. M., and M.E. Patzkowsky, 2004. Ecosystem structure and stability: Middle Upper Ordovician of central Kentucky, USA. Palaios 19:316–331.

Holland, S. M., and A. Zaffos, 2011. Niche conservatism along an onshore-offshore gradient. Paleobiology 37: 270–286.

Jongman, R.H.G., C.J.F. ter Braak, and O.F.R. Van Tongeren, 1995. Data Analysis in Community and Landscape Ecology. Cambridge University Press: Cambridge, 299 p.