Lecture 28. Testing whether predictors explain variation in composition

PERMANOVA

Constrained ordination is great for visualizing how complex multivariate data are related to one or more predictors. As we saw in the previous lecture, we can do hypothesis tests on the results, though the method isn't really designed for hypothesis tests per se. Here I'll discuss a method that is probably the best current option for testing the effect of predictors on a multivariate response.

The classic approach for modeling a multivariate response is called MANOVA, or multivariate analysis of variance. The concept is a fairly straightforward extension of ANOVA and linear regression. With anova and regression, we ask what proportion of the variation in the normally distributed response variable is explained by predictor(s). When the response is multivariate, such as a matrix of abundances for many species, we can again ask what proportion of the variation in multivariate space is explained by the predictor(s). Here's a nice schematic off the web (http://geog.uoregon.edu/bartlein/courses/geog495/lectures/lec17.html):

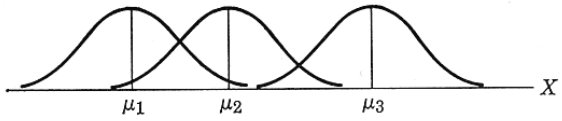


Figure 8.1. The simple anova situation, when the differences among the populations are "real."

source: Cooley & Lohnes ((1971)

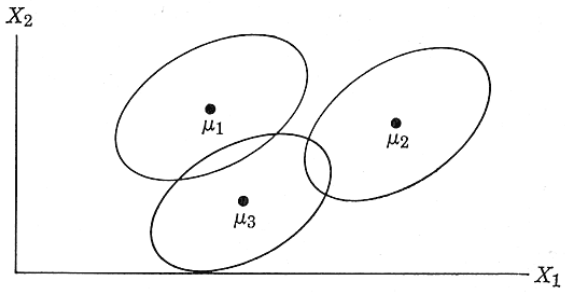


Figure 8.2. The simple manova situation, when the differences among the populations are "real."

With an anova, we are essentially quantifying the difference between the means of the three groups, relative to the difference within each group. The plot below shows a two-dimensional multivariate response. Now the mean for each group is the centroid of the data for that group, which is just the point defined by the mean along each axis. A MANOVA will ask how much variation there is (euclidean distance) among the centroids, relative to the amount of variation within the groups.

This all seems pretty straightforward, but the problem is that a manova assumes the data come from a *multivariate normal distribution*. It turns out that satisfying multivariate normality is harder than satisfying univariate normality, and the results of the manova are less robust to violations of this assumption (compared to anova). For example, multivariate normality requires that each individual dependent variable is normally distributed, that all pairs of dependent variables are linearly related, and that the covariance between dependent variables is the same within each group defined by the independent variables. This is a lot to satisfy.

Because of the stringent requirements of MANOVA, other methods have been devised to fill this niche. It is particularly important for community ecology experiments, where something is manipulated (e.g. the presence of predators), and we want to ask whether there is a significant change in community composition.

A nice method invented by Marti Anderson is called PERMANOVA. The ingredients of the method will be familiar to you from what we've already covered. First, a dissimilarity matrix is required. This could be a true distance matrix, e.g. based on euclidean distances between samples in species space. Or it could be a dissimilarity matrix using Bray-Curtis or some other dissimilarity matrix. With one or more predictors, some clever math can be used to partition the variation in the matrix among the predictors. This tells us how much variation in community composition (how much dissimilarity) is explained by each predictor. Finally, we need some way to test the significance of the explained variation, and so permutation tests are used.

Let's try this, using the adonis() function in vegan:

```
adonis2(species.data ~ month + nitrate + chl + temp + PAR, data = enviro,
dist = 'bray', by = 'margin')

## Permutation test for adonis under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species.data ~ month + nitrate + chl + temp + PAR, data
= enviro, by = "margin", dist = "bray")
##           Df SumOfSqs      R2      F Pr(>F)
## month     11   1.9061 0.09245 1.3208  0.052 .
## nitrate    1   0.2280 0.01106 1.7377  0.073 .
## chl        1   0.4124 0.02000 3.1437  0.004 **
## temp       1   0.1470 0.00713 1.1205  0.290
## PAR        1   0.1339 0.00650 1.0209  0.376
## Residual  88  11.5452 0.55997
## Total    103  20.6176 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note I have used the function adonis2(), rather than adonis(), because it can do marginal tests on each predict. I.e., it asks whether each predictor can explain significant variation in the dissimilarity matrix after the effects of the other predictors have been accounted for. If you prefer to do sequential tests, where each term is tested in order, based on the order of the terms in the model formula, you can use adonis(). The results of the marginal tests in this case suggest that chl and maybe month are significant predictors of community composition. However, the different predictors in the model are themselves correlated, and so it might be worth investigating whether the effect of a predictor depends on which other predictors are in the model.

Permanova is a great method, because it can use any dissimilarity matrix, and does not require any particular distributional assumptions to do hypothesis tests. However, it has several limitations which you should keep in mind. The first is the reliance on permutation tests.

Digression on permutation tests

Permanova relies on permutation tests, which are a general tool worth considering in a variety of contexts. The logic of permutation tests is that we often address scientific questions using null hypotheses, and null hypotheses are typically stating a *lack of difference* between groups, or a *lack of relationship* between predictors. Therefore, instead of specifying a particular model of our data, if we can randomize the data in a way that simulates the null hypothesis, then we can use the data itself to get a null distribution for our test.

A simple example is given in the vignette for the R package 'permute'. There are 20 measurements of jackal jaw length, 10 from males and 10 from females. To ask whether there is a different in mean jaw length between males and females we could perform a t-test:

```
R> library("permute")
R> data(jackal)
R> jackal

   Length    Sex
1     120   Male
2     107   Male
3     110   Male
4     116   Male
5     114   Male
6     111   Male
7     113   Male
8     117   Male
9     114   Male
10    112   Male
11    110 Female
12    111 Female
13    107 Female
14    108 Female
15    110 Female
16    105 Female
17    107 Female
18    106 Female
19    111 Female
20    111 Female
```

```
R> jack.t <- t.test(Length ~ Sex, data = jackal, var.equal = TRUE,
+                   alternative = "greater")
R> jack.t

        Two Sample t-test

data:  Length by Sex
t = 3.4843, df = 18, p-value = 0.001324
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 2.411156      Inf
sample estimates:
  mean in group Male mean in group Female
               113.4                 108.6
```

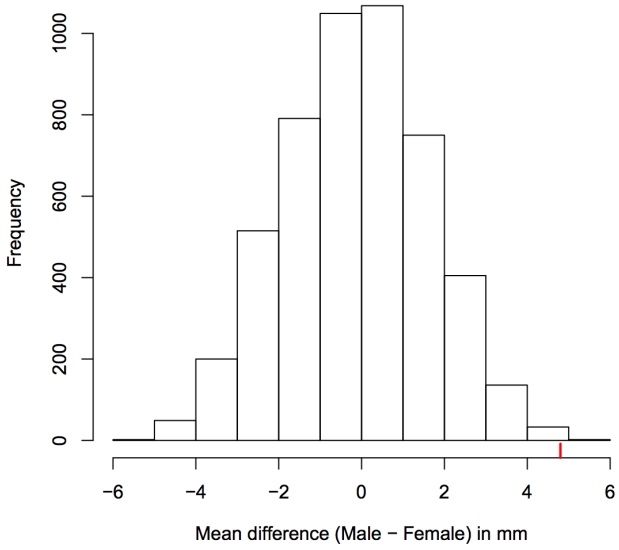
For a t-test we assume the distribution of jaw lengths is gaussian. If instead we use a permutation approach, we don't have to assume any particular distribution for the data (but we do have to assume the observations are independently and identically distributed, as I will explain). The permutation approach relies on the fact that the null distribution is that females and males have the same mean jaw length. If this is true, it means we can randomly reassign male/female status to the observed jaw lengths, to simulate what kinds of differences we might observe under the null hypothesis. The following code does this:

```
R> meanDif <- function(x, grp) {
+   mean(x[grp == "Male"]) - mean(x[grp == "Female"])
+  }
```

```
R> Djackal <- numeric(length = 5000)
R> N <- nrow(jackal)
R> set.seed(42)
R> for(i in seq_len(length(Djackal) - 1)) {
+      perm <- shuffle(N)
+      Djackal[i] <- with(jackal, meanDif(Length, Sex[perm]))
+  }
R> Djackal[5000] <- with(jackal, meanDif(Length, Sex))

R> (Dbig <- sum(Djackal >= Djackal[5000]))

[1] 12
```

giving a permutational p-value of

```
R> Dbig / length(Djackal)

[1] 0.0024
```

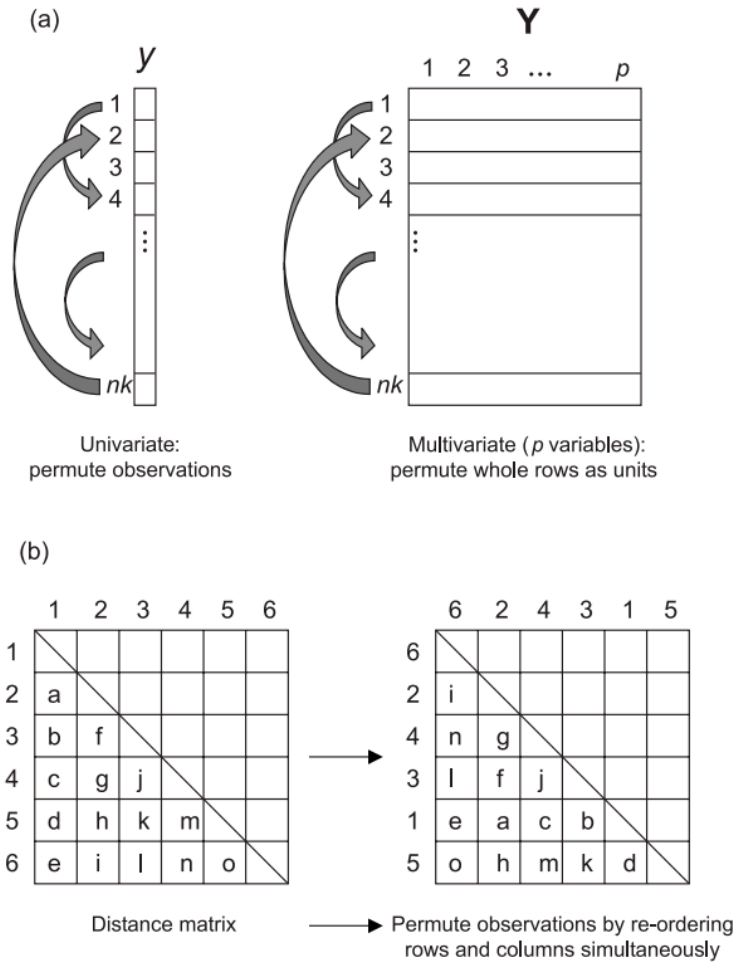


This code uses the function shuffle() to randomly reorder the vector 1:20, and uses that vector to reorder the Sex vector that codes male/female. Then the difference in mean jaw length between males and females is calculate, after shuffling the male/female labels. This is repeated 5000, to get a distribution of what the sex difference in jaw length might look like under the null hypothesis. On average the difference is zero, which makes sense, but at the same time sometimes the difference will deviate from zero by chance. If we compare the observed difference of ~4.8 to the null distribution, a difference this large only occurs 0.24% of the time, or p = 0.0024.

The p-value obtained through permutation is similar to that obtained with a t-test, although this will not always be true. The permutation does not assume the values are normally distributed, but it does assume they are *exchangeable*. I.e., it only makes sense to reshuffle male/female labels if the observations have the same distribution, and if they are not correlated with each other. If males and females

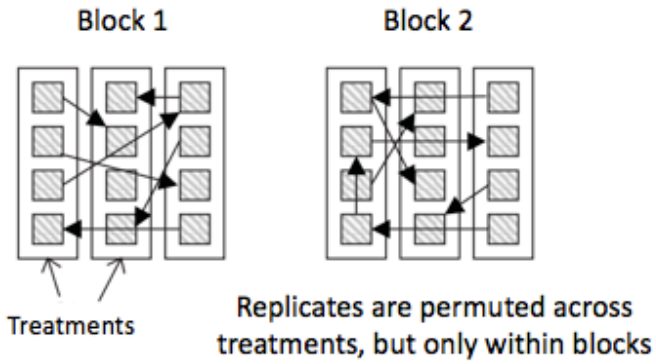
have different variance, they are not exchangeable, and if there is some kind of correlation structure in the data, e.g. due to genetic structure or spatial/temporal structure, then it doesn't make sense to use a null model where these associations are destroyed during the randomization process.

The same logic can be applied to multivariate datasets, where we are trying to associate a matrix (for example, community composition) to some predictor variables. Now, we permute entire rows (where the rows are different samples of community composition), or we directly permute the rows and columns of a dissimilarity matrix calculated from the community matrix:



One challenge of a permutation approach to hypothesis testing is complex experimental or survey designs. For example, treatments are implemented within experimental blocks: how do we account for the effect of block when randomizing samples across treatments? One approach to deal with this limitation is to do *stratified* permutations, where the data are be permuted *within* each block. This would maintain any within-block similarity when constructing the null distribution for the test statistics. Instructions on how to do this can be found in the help file for

the function how() in the package 'permute', which is designed to work with adonis() and other vegan functions.



Finally, complex designs can also be accounted for by permuting residuals instead of permuting the raw data. For example, if we want to test how particulate organic matter (POM) affects bivalve growth, while accounting for any effect of temperature, we can first regress growth on temperature, then permute the residual variation (which will include any marginal effect of POM), add the permuted residuals back to the fitted effects, and then test for the effect of POM. This will generate a null distribution for the regression coefficient for POM. A similar algorithm is implemented by adonis2() when you make marginal tests of the predictors.

Location vs. dispersion effects

An important limitation of Permanova is illustrated in this plot from Warton et al. 2012 (Methods in Ecology and Evolution, "Distance-based multivariate analyses confound location and dispersion effects")

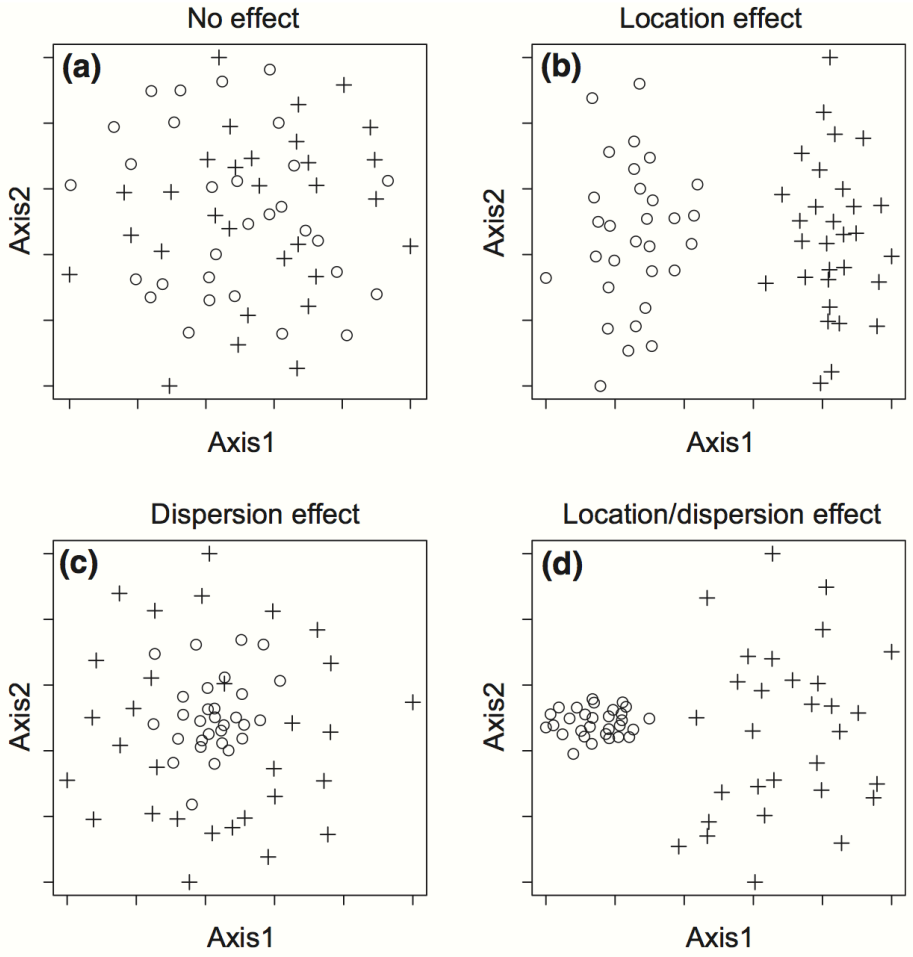


Fig. 2. A schematic diagram as in Anderson *et al.* (2008) illustrating on two axes the types of between-group effects that are often of ecological interest: (a) *no effect*; (b) *location* effect; (c) *dispersion* effect; (d) *location*/*dispersion* effect.

Plot (b) is the kind of thing we want to quantify using permanova: some treatment or other predictor causes the multivariate mean of the data to shift. This is called a *location* effect, because the groups differ in their multivariate location. Plot (c) shows a different kind of effect: the centroids of the two groups are the same, but one has higher variance than the other. This is called a *dispersion* effect.

The problem is that either (b) or (c) can cause a significant difference in a permanova analysis, even though we're usually interpreting the result as meaning (b). There isn't a great solution for this, although the possibility can be assessed using ordination plots as well as the betadisper() function, which will tell you if your groups have significantly different dispersion.

Mixed models and GAMs as options for multivariate data

The multivariate methods I've covered are widely used, and essentially represent the state of the art, particularly for analysis of multivariate community data. However, I personally try to avoid these methods if possible, because all of the data transformation, conversion to dissimilarities, and distance-based analysis of those similarities makes it hard to understand exactly what you're getting at the end. Plus there are the issues with doing valid hypothesis tests that I just mentioned.

If your focus is on seeing how community structure changes as environmental variables change, or seeing if community structure changes do to an experimental manipulation, then mixed models are a viable alternative. To start with, we need a dataframe in 'long' format, so that there is a single column for the abundance of all species, and another column to code what species the data is from:

```
channel[1:5,c("abundance", "species", "nitrate")]

## abundance species                nitrate
## 0.00000   Cerataulina.pelagica   7.980
## 0.00000   Cerataulina.pelagica   7.350
## 0.00000   Cerataulina.pelagica   4.883
## 0.02591   Cerataulina.pelagica   3.115
## 0.15530   Cerataulina.pelagica   0.100
```

I've also created a column 'presence', because I've decided to model species as being just presence or absent, because the data are highly skewed and contain many zeros.

Now we can do the following:

1) Make presence/absence the response for a binomial (binary) model
2) Model presence/absence as a function of whatever predictors we care about (e.g. nitrate, PAR)
3) Allow the different species to respond differentially to these predictors, by including *random slopes*
4) Quantify the change in community composition in terms of how much the slopes of the response differ across species

The upshot here is the *we can actually model the raw data*. That means the results will be more sensitive to any predictors, and less likely to break some hidden assumption we don't understand very well. As a simple example, I made a new variable, 'season', that just divides the year into summer and winter months. This will also show what you would do with an experimental treatment:

```
mod.season = glmer(presence ~ season + (1+season|species), data =
channel, family = binomial)
```

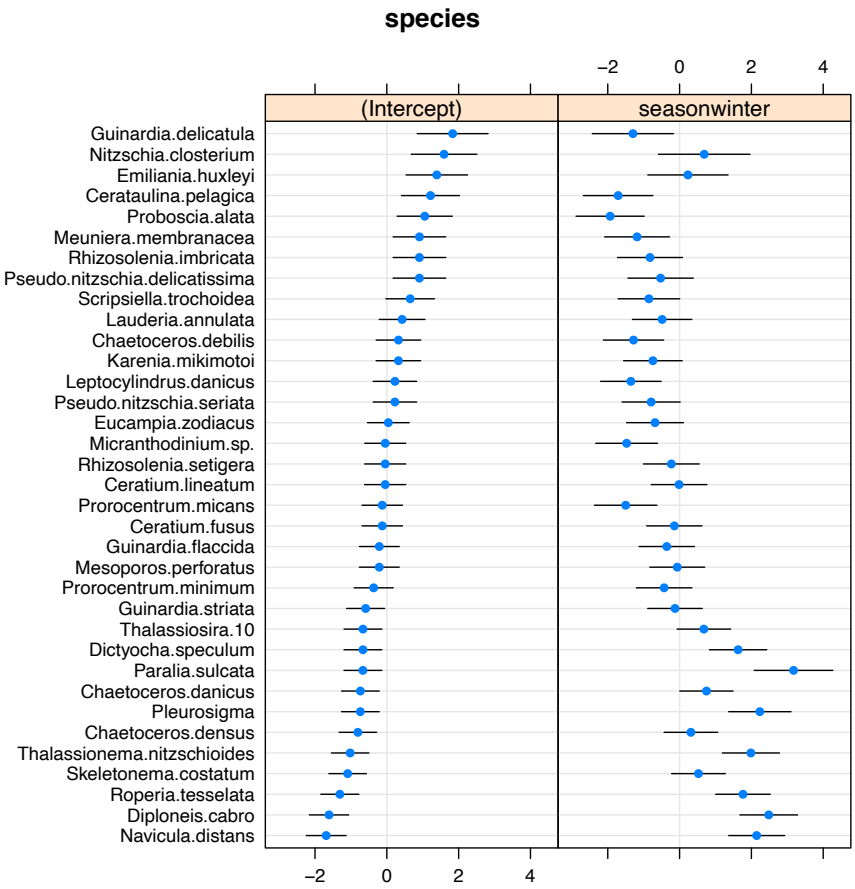
Here I'm saying that the probability of presence differs between seasons, overall abundance differs randomly among species, and the effect of season differs

randomly among species. The key point here is that *we can use the random effect for season to quantify how much community composition changes between seasons*. If this random effect is nonzero, that means that species change in relative abundance between seasons. And the larger the effect is, the more composition changes.

```
summary(mod.season)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: presence ~ season + (1 + season | species)
##    Data: channel
##
##      AIC      BIC   logLik deviance df.resid
##     4361     4392    -2176     4351     3635
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.026 -0.838  0.394  0.741  2.103
##
## Random effects:
##  Groups  Name         Variance Std.Dev. Corr
##  species (Intercept)  0.91     0.954
##          seasonwinter 1.96     1.400    -0.69
## Number of obs: 3640, groups:  species, 35
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.953      0.172    5.54    3e-08 ***
## seasonwinter   -0.811      0.250   -3.24   0.0012 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## seasonwintr -0.695
```

```
dotplot(ranef(mod.season, condVar = T))
```

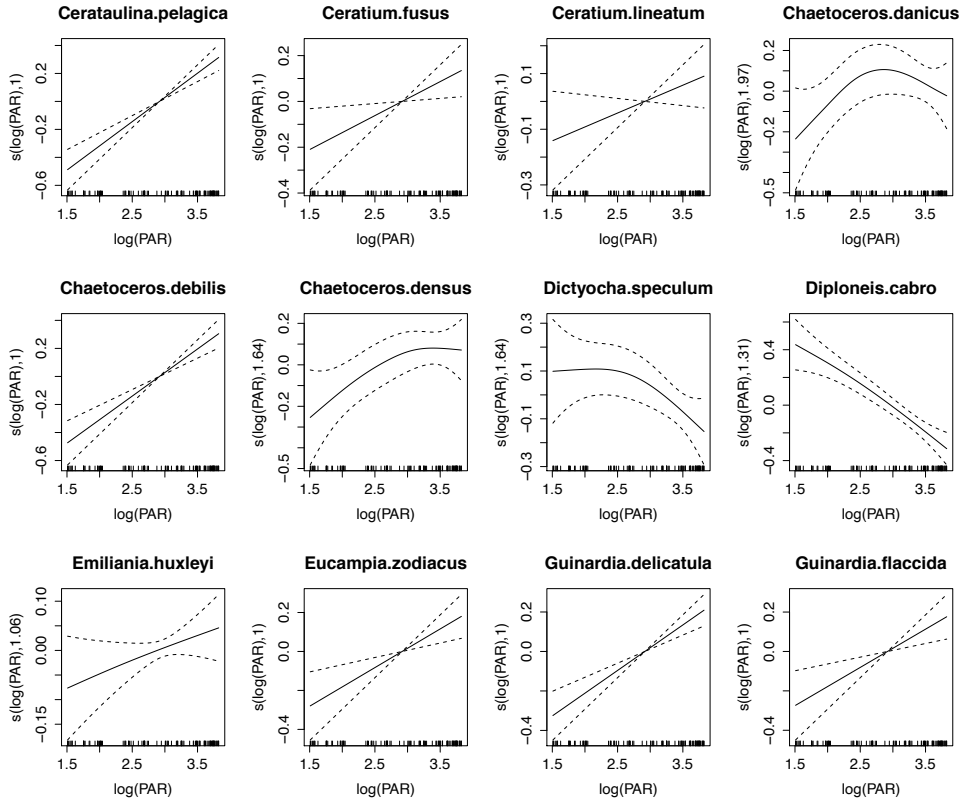


We can see that the variation in the season effect across species is large. The standard deviation is 1.4, which is fairly large. Going from 0 to 1.4 on the logit scale corresponds to going from 0.5 to 0.8, in terms of probability of presence. So the implication is that the composition of the community, in terms of who is present, changes a lot between seasons, because the species have much different season effects. We can test the season effect by dropping the random slope and doing a LRT:

```
mod.no.season = glmer(presence ~ season + (1|species), data = channel, family
= binomial)

anova(mod.season, mod.no.season)

## Data: channel
## Models:
## mod.no.season: presence ~ season + (1 | species)
## mod.season: presence ~ season + (1 + season | species)
##               Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mod.no.season  3 4610 4628  -2302     4604
## mod.season     5 4361 4392  -2176     4351   253      2     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The species-specific responses to season are highly significant, and also cause a large drop in AIC.

The upshot of this approach is that we can see the species-specific responses more clearly, while getting at the same scientific question. The downside is that if we want to use continuous environmental predictors, we need to assume the effects of those predictors are linear. Often this is roughly true anyways. For example, I looked at 12 of the species in the dataset, and fit a binomial (logit) GAM to see how their presence/absence varies with PAR:



In some cases the data are somewhat non-monotonic, i.e. unimodal, such as *Chaetoceros danicus*. But most of these would be pretty well captured by a linear relationship, especially for purposes of seeing how the direction/steepness of the response varies across species.

Next I standardized my three predictors, such that they have a mean of zero and a standard deviation of one:

```
channel$nitrate = with(channel, (nitrate - mean(nitrate))/sd(nitrate))
channel$temp = with(channel, (temp - mean(temp))/sd(temp))
channel$PAR = with(channel, (log(PAR) - mean(log(PAR)))/sd(log(PAR)))
```

This will help with interpreting a random slopes model, because now a slope of *b* has the same meaning for each predictor: if that predictor increases by one standard deviation, then the response increases by *b* units.

Now we can fit the model:

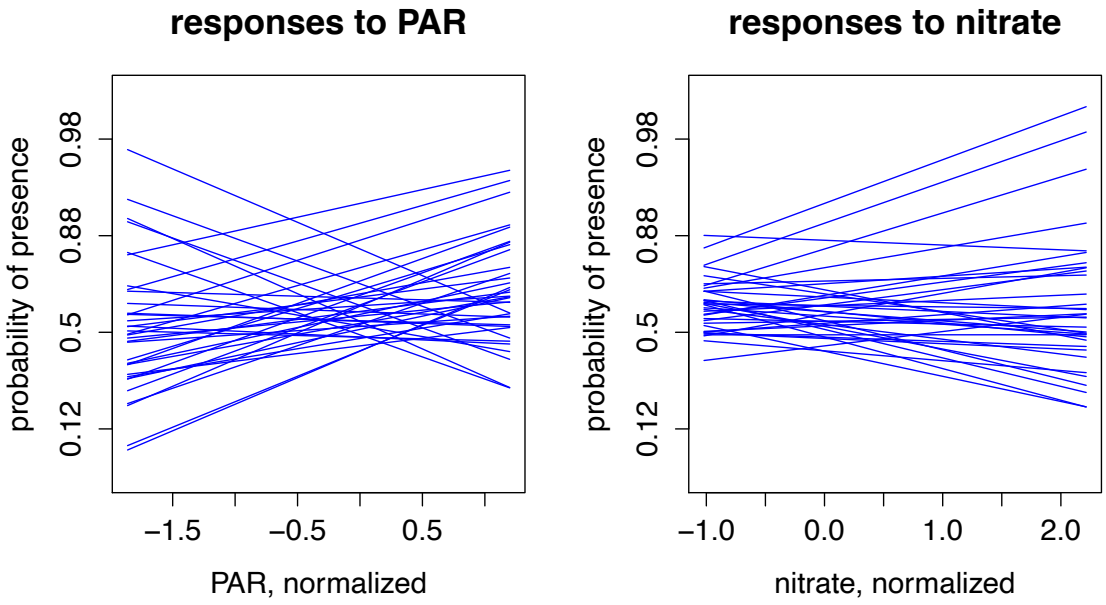
```
mod = glmer(presence ~ nitrate + temp + PAR + (1+nitrate+temp+PAR|species), da
ta = channel, family = binomial)
```

I've included a random slope for each predictor. This will let us see whether species differ in their response to those predictors. If so, this means that the community composition changes as those predictors change.

```
summary(mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: presence ~ nitrate + temp + PAR + (1 + nitrate + temp + PAR |
##     species)
##    Data: channel
##
##      AIC      BIC   logLik deviance df.resid
##     4195     4282    -2084     4167     3626
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.700 -0.779  0.347  0.677  3.936
##
## Random effects:
##  Groups  Name        Variance Std.Dev. Corr
##  species (Intercept) 0.592    0.769
##          nitrate     0.259    0.509    0.52
##          temp        0.189    0.435    0.26 0.07
##          PAR         0.536    0.732    0.07 0.09 0.02
## Number of obs: 3640, groups:  species, 35
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 5.76e-01   1.37e-01    4.20  2.6e-05 ***
## nitrate     4.39e-05   1.23e-01    0.00    1.000
## temp        3.86e-01   9.37e-02    4.12  3.8e-05 ***
## PAR         2.54e-01   1.43e-01    1.78    0.075 .
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The fixed effects show that on average, species tend to increase as temperature increases. There are not strong mean trends for nitrate and PAR. The random effects show that all of the predictors have substantial interspecific variation. To better visualize what this means, we can use the fixed and random effects to plot the predicted responses for each species:



Here the y-axis is probability of presence, plotted on the logit scale. Across the range of PAR in the data, there is substantial change in community composition, with many species increasing or decreasing their probability of presence by ~0.5. Nitrate also leads to shifts in composition, although the amount of variation in slopes is somewhat less than for PAR. To test whether there is significant variation in slopes, we can use LRTs:

```
anova(mod, mod.no.nitrate)
```

```
## Data: channel
## Models:
## mod.no.nitrate: presence ~ nitrate + temp + PAR + (1 + temp + PAR | specie
s)
## mod: presence ~ nitrate + temp + PAR + (1 + nitrate + temp + PAR |
## mod:     species)
##                Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mod.no.nitrate 10 4200 4262  -2090     4180
## mod            14 4195 4282  -2084     4167  12.8      4      0.012 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod, mod.no.temp)
```

```
## Data: channel
## Models:
## mod.no.temp: presence ~ nitrate + temp + PAR + (1 + nitrate + PAR | specie
s)
## mod: presence ~ nitrate + temp + PAR + (1 + nitrate + temp + PAR |
## mod:     species)
##             Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mod.no.temp 10 4222 4284  -2101     4202
## mod         14 4195 4282  -2084     4167  35.2      4    4.2e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(mod, mod.no.par)

## Data: channel
## Models:
## mod.no.par: presence ~ nitrate + temp + PAR + (1 + nitrate + temp | specie
s)
## mod: presence ~ nitrate + temp + PAR + (1 + nitrate + temp + PAR |
## mod:     species)
##            Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mod.no.par 10 4262 4324  -2121     4242
## mod        14 4195 4282  -2084     4167  75.3      4    1.7e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In terms of AIC, the important of the factors is PAR > temperature > nitrate, though all appear to affect community composition.

I think this model-based approach to multivariate community data is going to grow in popularity, because it deals with the raw data directly, and lets you see the species-specific responses while also quantifying whole-community effects in terms of varying slopes. It does require linear responses to continuous environmental factors, but in many datasets this is often approximately true. If responses are strongly nonlinear, an alternative approach is to fit a GAM, where the smoother varies by species. E.g.,

```
mod.gam = gam(presence ~ s(nitrate, by = species) + s(temp, by = species) +
s(PAR, by = species), data = channel.common, family = binomial)
```

Then you can visualize the variation in composition in terms of how different the smoothers are across species, and can test for significant compositional effects using an LRT. This approach is probably more feasible for modest numbers of species (~10), as the computation of the model can really slow down for larger numbers.