Lecture 29. Multidimensional scaling.

Principal coordinates analysis

In the last lecture we discussed PCA, and then thought about whether it is useful for ordination of community composition. It seems like using euclidean distance as a measure of community dissimilarity has some issues, as does quantifying species' relationships with linear correlation. Then I introduced indices of similarity/dissimilarity that seem more appropriate for community ecology, particularly because they account for the 'double zero' problem.

Now we that have a way to quantify how different two communities are, what do we do with that? We're currently focused on ordination, i.e. we want to map out variation in community composition. And to do that we want to find axes of variation along which communities are very different from each other. If we use an index like the Jaccard or Bray-Curtis coefficient, and calculate the coefficient for each pair of samples, what we get is a *dissimilarity matrix*:

	Site1	Site2	Site3	Site4
Site1	0	0.2	0.6	0.3
Site2		0	0.5	0.1
Site3			0	0.8
Site4				0

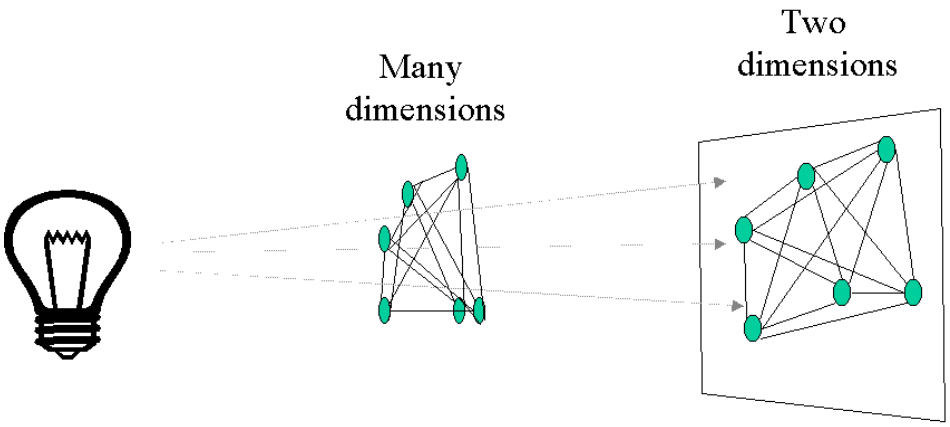
Here I'm imagining we sampled four sites, measured a bunch of species abundances at those sites, transformed those abundances in some way, and then calculated a dissimilarity coefficient between each pair of sites. I only show the upper triangle of the matrix, because the lower triangle would be redundant (the dissimilarity between site 2 and site 3 is the same as the dissimilarity between site 3 and site 2). Also I could have removed the diagonal, because a site has dissimilarity of 0 with itself.

The dissimilarity matrix is now our 'data' for further analysis. Note that the original species abundances are no longer in the picture; they were used to get the dissimilarities, but they will not be involved in the next step of the analysis. In order to use the dissimilarities for ordination, we're going to convert these *dissimilarities* into spatial *distances*, by finding axes of variation that show the greatest total dissimilarity. This is directly analogous to how PCA works: in that case, we find the orthogonal axes that account for the most variation in the data, and that variation is quantified as the euclidean distance. So now instead of using euclidean distance,

we're going to use some other measure of dissimilarity, and again find orthogonal axes that account for as much variation as possible.

This method is called *principal coordinates analysis (PCoA)*, and is also known as *(metric) multidimensional scaling (MDS)*. It is an eigenanalysis method just like PCA. Once we have our dissimilarity matrix, we find the eigenvectors and eigenvalues of that matrix, and use those to calculate the ordination.

PCoA is harder to visualize than PCA, because we can't think of it as a rotation of axes in multi-dimensional 'species space'. The species data have been compressed into the dissimilarity matrix, and that is what the analysis uses. Here is the best attempt at visualization I found on the web, from this site (http://ordination.okstate.edu/overview.htm):



Each green point represents a sample (e.g. a site), and the lines beween the samples represent the dissimilarity between those samples. With many samples, fully representing all the distances would require many dimensions, but the PCoA projects these distances onto a few dimensions (usually 2), in such a way that the distances in 2D space approximate the distances between the points as well as possible.

Let's do a PCoA in R, using the English Channel phytoplankton again.

```
species.data.use = wisconsin(sqrt(species.data.use))

pcoa = capscale(species.data.use ~ 1, dist = "bray")
pcoa

## Call: capscale(formula = species.data.use ~ 1, distance = "bray")
##
##               Inertia Rank
## Total            5.17
## Real Total       6.21
## Unconstrained    6.21   19
## Imaginary       -1.04   20
## Inertia is squared Bray distance
##
```

```
## Eigenvalues for unconstrained axes:
##  MDS1  MDS2  MDS3  MDS4  MDS5  MDS6  MDS7  MDS8
## 2.635 0.907 0.804 0.500 0.459 0.196 0.157 0.140
## (Showed only 8 of all 19 unconstrained eigenvalues)
```

I've used the function capscale() from vegan. This function can also do 'constrained' ordination, which we will discuss later; for now just note that I put a formula with ~ 1 to do PCoA. I gave the function a matrix of transformed species abundance (the rows are samples over time). The PCoA actually uses a dissimilarity matrix, and capscale() will turn the data into a dissimilarity matrix automatically, which is why I said distance = 'bray'. Alternatively I could make the distance matrix myself with vegdist(), and give that to capscale().

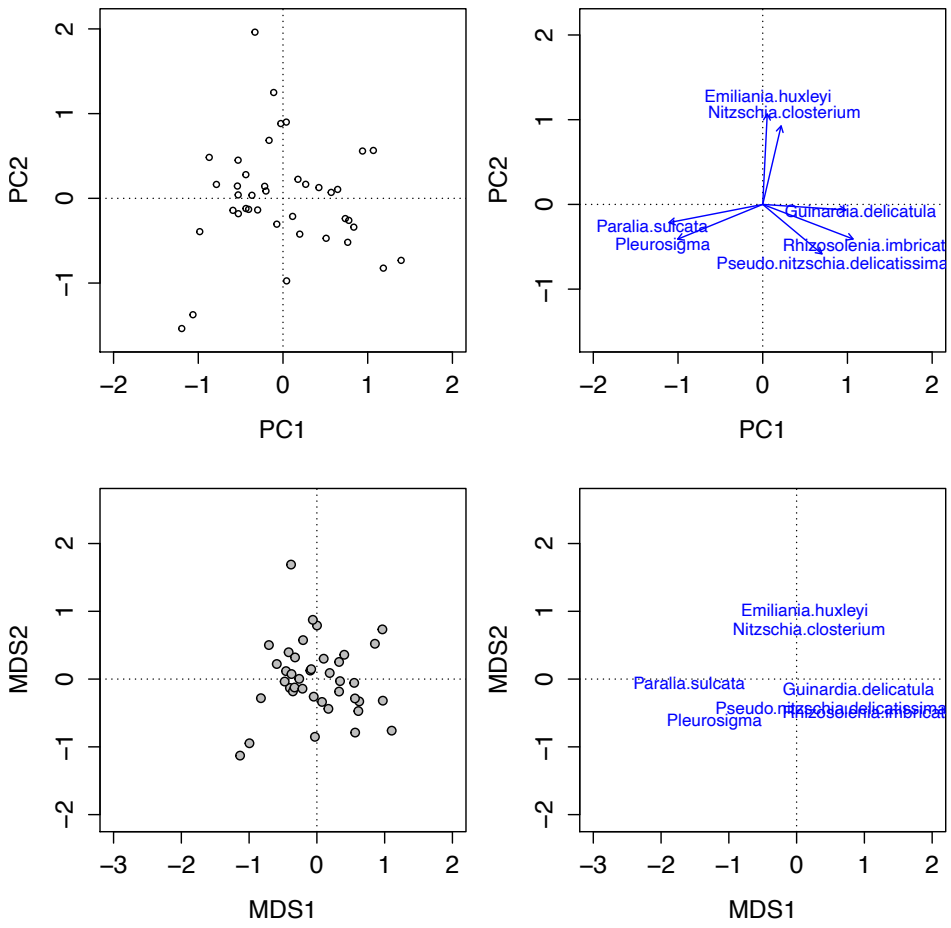
The function returns an Inertia, with real and imaginary components. The total inertia is the summed squared Bray distance, which is the total variation in the data. The eigenvalues sum to this total as well, and the proportion variance explained by an axis is the eigenvalue relative to the sum of the eigenvalues (like PCA). PCoA has a weird feature where some eigenvalues can be negative (which would seem to indicate negative variance explained). This happens because a coefficient like Bray dissimilarity is not a true distance, in mathematical terms (it violates the triangle inequality); there are various corrections for this issue, which are handled automatically by capscale, but which you can also specify yourself if you dig into the help file. We can get more info with summary():

```
summary(pcoa)
##
## Eigenvalues, and their contribution to the squared Bray distance
##
## Importance of components:
##                          MDS1  MDS2  MDS3   MDS4   MDS5   MDS6   MDS7   MDS8
## Eigenvalue            2.635 0.907 0.804 0.4999 0.4588 0.1958 0.1575 0.1396
## Proportion Explained  0.424 0.146 0.130 0.0805 0.0739 0.0315 0.0254 0.0225
## Cumulative Proportion 0.424 0.571 0.700 0.7806 0.8545 0.8861 0.9114 0.9339
```

I have abbreviated this to the variance explained. We have one moderately large axis, which explains 42% of the variance, and a long tail of additional axes. Summary also returns scores for the sites and for the species, but I'll plot those instead:

```
par(mfrow = c(2,2))
biplot(pca, type = 'text', col = c('blue'), display = "species", xlim = c(-
2,2), cex = 0.5)
biplot(pca, type = c('points'), col = c('black'), display = "sites", xlim =
c(-2,2))
```

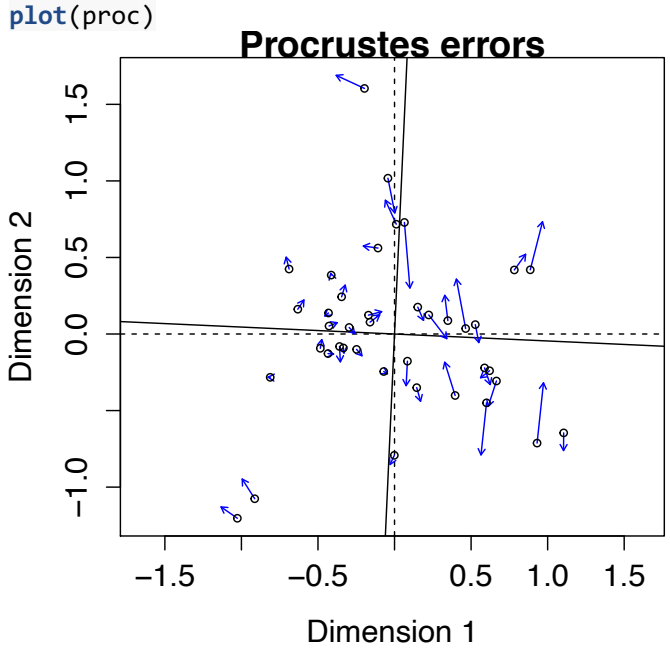
```
plot(pcoa, type = 'n', xlim = c(-3, 2))
text(pcoa, "species", col = 'blue', cex = 0.7)
plot(pcoa, type = 'n', xlim = c(-3, 2))
points(pcoa, col = 'black', bg = 'grey', pch = 21)
```



On the top row I've plotted a PCA on the same data for comparison. On the bottom row are sample scores and species scores for the PCoA. Note that the species scores for the PCoA *are not vectors*. PCA returns vectors for the species because the eigenvectors are linearly related to the original axes, which were species' abundances. PCoA returns points, not vectors, because in this method the eigenvectors, scaled by the eigenvalues, are axes that approximate the distances between observations. The species are then ordinated on the same plot with weighted averages: by going back to the raw data, finding the average location of the species on the PC axes, and averaging the scores of the samples where the species occurs, weighted by its abundance in those samples. So the interpretation is now that samples near to a species in the biplot are expected to have the highest abundance of that species.

From the plots of the species scores, we can see that even though they do not have the same exact meaning for PCA and PCoA, they nonetheless show similar patterns in terms of the relative placement of different species. This is reassuring (and is not always the case). Comparing the ordinations of the samples scores is more difficult, as they are not labeled here, but we can compare the ordinations using a procrustes rotation:

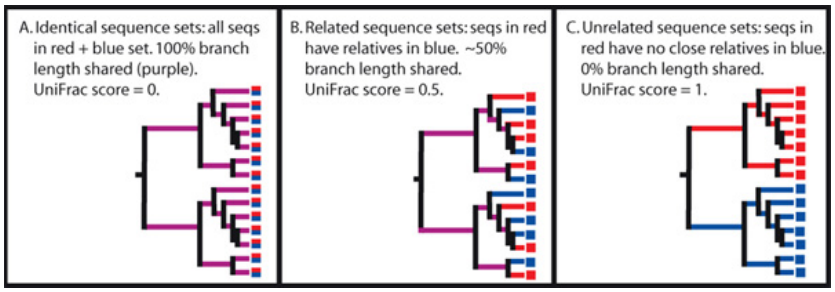
```
proc = procrustes(pcoa, pca)
plot(proc)
```



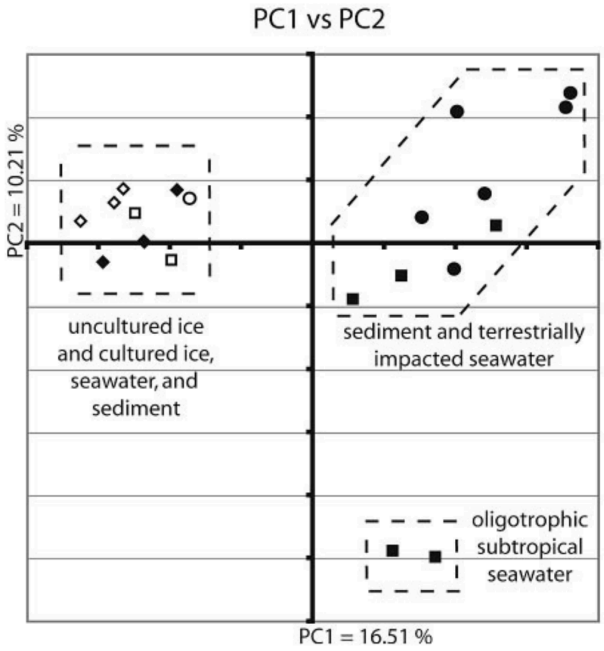
The procrustes function takes an ordination (here the PCA ordination), rotates it to align it with another ordination (here, PCoA), and then shows how much the points need to move in order to be in the same spot. So the black axes are the PCA axes, rotated to align the PCA scores as well as possible with the PCoA scores; and the arrows show where the PCA scores have to move in order to align with the PCoA scores. In other words, it's showing how different the two ordinations are. In this case they look pretty similar, which if nothing else suggests the results are robust.

There are a number of other things we could do with the PCoA ordination: see if the axes are correlated with environmental variation, or see if observations from different times of year form clusters, or see if related species have similar scores. But I will instead do that with a related but generally preferred method, NMDS.

Before I introduce NMDS, I'll show another example of how PCoA is used. I've used transformed species abundance data, and calculated Bray-Curtis dissimilarities, to ordinate community samples. The nice thing about PCoA is that you can use any distance metric. Microbial ecologists have started doing a lot of ordination, now that sequence data can be collected for many microbes simultaneously in a single sample. That sequence data can be used to calculate a dissimilarity between two community samples that is based on the phylogenetic distance between the samples. E.g., the UniFrac method creates a single phylogenetic tree that encompasses all the sequences from both samples, and calculates similarity as the proportion of total branch length that is shared between the two communities.



The authors of this method showed with sequence data from cultured and uncultured isolates that bacteria in culture were similar to each other (regardless of which environment they came from), and similar to uncultured bacteria from sea ice, but distinct from uncultured isolates from seawater and sediment.

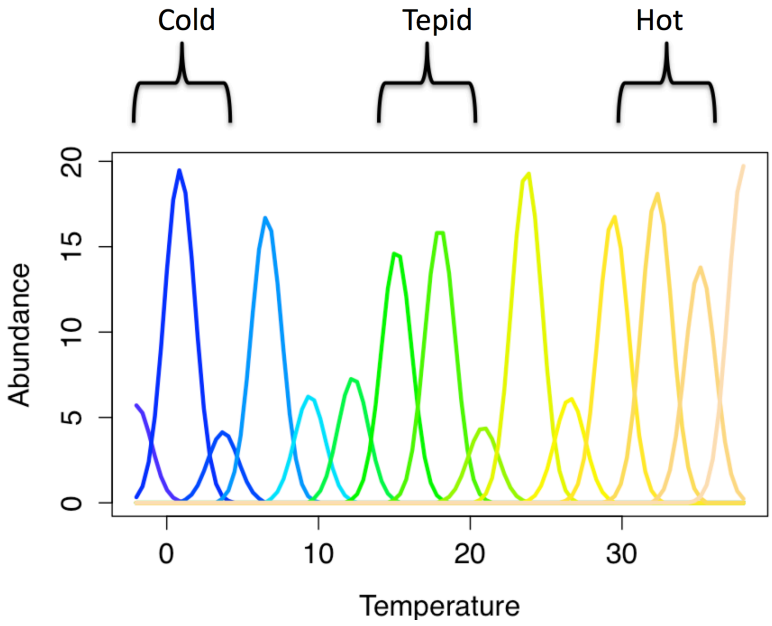


PCoA is not just created for community ordination. E.g. if you want to map out how similar organisms are in trait space, and the traits include continuous variables (e.g. size) as well as categorical variables (e.g. growth form), then you could summarize the trait distance between organisms using Gower's dissimilarity coefficient, and then use PCoA to ordinate.

Non-metric multidimensional scaling

PCoA is somewhat more advantageous for analyses of community composition, relative to PCA, because it can use distance coefficients that are suitable for comparing communities. A limitation of this method is that it is still a linear method in some ways. If we are comparing communites along a 'long' gradient, i.e. one with high species turnover, then we expect a nonlinear relationship between

community similarity and the distance between communities along the gradient. E.g. imagine that were are comparing communities along a temperature gradient. If we compare 'hot, 'tepid', and 'cold' communities, then the cold community may not share any species with either the tepid or the hot community. This means the pairwise similarities will be zero.

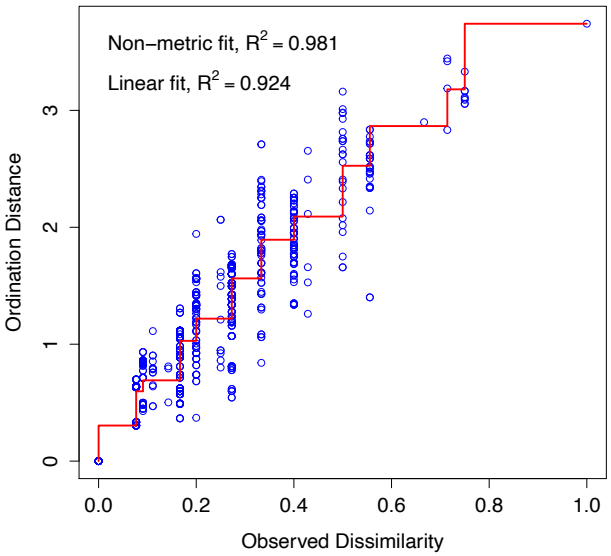


However, in terms of ordinating these communities along a gradient of similarity, the tepid community should be closer to the cold community than the hot community. PCoA is not well-suited to capture this whole gradient, because it basically assumes that similarity declines linearly along the gradient (instead of bottoming out at zero).

The most popular method for handling this issue is called *non-metric multidimensional scaling (NMDS)*. The principle is similar to PCoA: we have a matrix of dissimilarities, and we want a low-dimensional representation of the samples, where the distance between two samples on the plot approximates their dissimilarity. PCoA treats the dissimilarity between two samples as being linearly related to their distance on the ordination plot. In contrast, NMDS only ensures that the *ranking* of dissimilarities is correlated with the *ranking* of distances on the ordination. This allows for nonlinear, but monotonic, relationships between dissimilarity and ordination distance.

Unlike PCA and PCoA, NMDS is not an eigenanalysis method, and it does not have a formula that give you the solution. Instead it uses an iterative procedure to find a good ordination:

1) Pick the *predefined* number of dimensions for the ordination. Only this many dimensions will be used to ordinate the data (it's usually 2-4, to make it plottable, but could be more).
2) Place the observations into ordination space with some initial configuration. Usually the initial configuration is chosen using another ordination method like PCoA.
3) Calculate a non-parametric regression that compares the real dissimilarities between observations to the distance between those observations in ordination space:



Use this regression (the red line) to calculate the *stress*, which is a measure of mismatch between the ordination distances and the dissimilarities they are approximating:

$$Stress = \sqrt{\frac{\sum_{h,i}\left(d_{hi} - \hat{d}_{hi}\right)^2}{\sum_{h,i} d_{hi}^2}}$$

Here d_{hi} is the distance on the ordination plot between observations *h* and *i*, and $\hat{d}_{hi}$ is the distance predicted from the nonparametric regression. Really this is just quantifying how well the ordination distances match up to the observed dissimilarities, and the stress is $\sqrt{1 - R^2}$ for the regression between these two. We use a nonparametric regression because we only care about the relative ranking, i.e. that observations that have greater dissimilarity are further apart in the ordination.

4) Modify the configuration of the observations in ordination space, in such a way that the stress is reduced (fit is improved).

5) Repeat #3 and #4 until the stress doesn't decrease any more, or decreases by a very small value.

Because this is an iterative algorithm working with high-dimensional data, the results may depend on the starting conditions (the initial configuration). Because of this potential sensitivity, the NMDS function 'metaMDS' in the vegan package adds some additional complexity to the algorithm, which you can read in the help file. Most importantly, it runs the algorithm from a series of different starting configurations, and treats the answer as 'convergent' if very similar ordinations are achieved from different starting values. Also:

- If you give it raw data, and ask it to turn that into a distance matrix for analysis, it will transform the data first if the values have a large range (you can turn this off if you want).
- Bray-Curtis is the default dissimilarity index, but you can specify others.
- If two pairs of observations have the same dissimilarity, e.g. both have the maximum dissimilarity of 1, then it allows 'weak' ties, such that these pairs are allowed to have different distances in the ordination. So in my earlier example, the distance between hot and cold would be allowed to be greater than the distance between tepid and cold, even though both pairs have dissimilarity = 1. This makes it much easier to successfully order samples along long gradients.

Here's an NMDS on the same data I used for the PCoA:

```
ord = metaMDS(species.data.use, dist = "bray", trymax = 20)
```

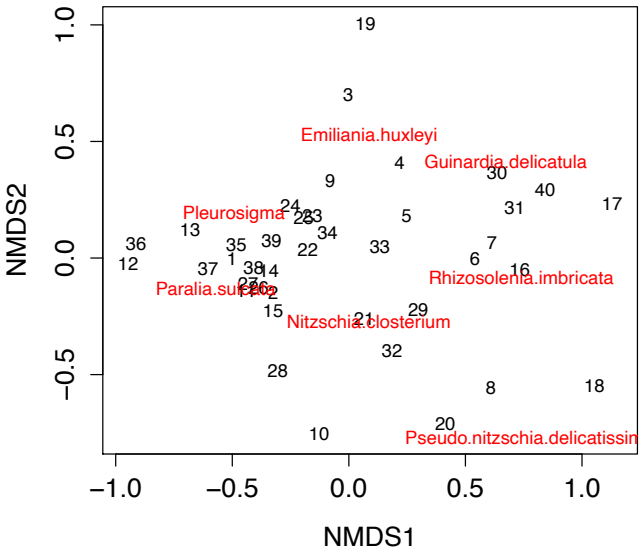
```
ord

##
## Call:
## metaMDS(comm = species.data.use, distance = "bray", trymax = 20)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     species.data.use
## Distance: bray
##
## Dimensions: 2
## Stress:     0.1488
## Stress type 1, weak ties
## Two convergent solutions found after 15 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'species.data.use'
```

The argument trymax tells it how many initial configurations to try when looking for a convergent solution. The default is 20, but you may want to increase this with more complex datasets if you don't get a convergent solution. The default #of dimensions for the ordination is 2. The solution has a stress of 0.1488. What does

this mean? It's hard to say for sure, but there are rules of thumb that a stress > 0.3 is quite bad and a stress < 0.1 is quite good. These rules have also been criticized because stress is not perfectly comparable across datasets. Notice that we do not have any eigenvalues/eigenvectors, and we do not have any explained variance measures. This is because there are no such quantities for NMDS, as it is nonparametric and meant to only preserve the rank order of the dissimilarities. The ordination scores for the observations are created as part of the fitting algorithm, and scores for the location of the species on the same plot can be created using a weighted average of the observation scores, as for PCoA. Let's make a biplot:

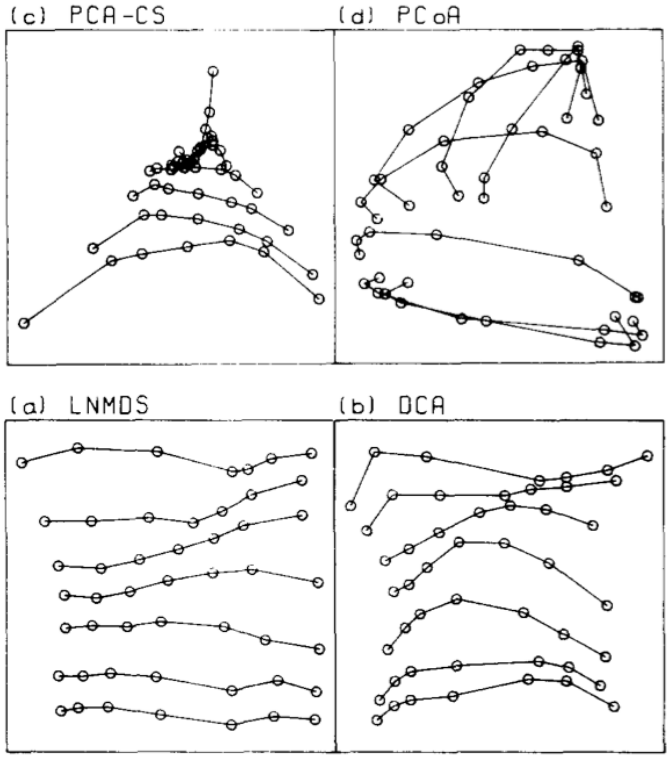
```
plot(ord, type = 'text')
```



The numbers are the observation scores, and the species scores are labeled in red. Another features of NMDS is that the axes are not special, in the sense that *the ordination can be rotated without changing the results*. This is different from PCA and PCoA, and it happens because the iterative algorithm is just moving the points around in the full ordination space, and seeing if that reduces the stress. For display purposes, the metaMDS function rotates the axes so that the most spread is along the first axis, etc, similar to a PCA (but this is an arbitrary decision). It also defines the scale of the axes on a 'half-change' scale, so that a distance of 1 corresponds roughly to a 50% turnover in community composition.

Is this ordination any good? Is it better than what we got from PCoA? The short answer is: we don't know. These aren't really models, they're complex data transformations, and so we can't use something like likelihood and AIC to ask if one ordination method 'fits the data better' than another. In my experience people generally just pick one method and use it. NMDS is the most common these days, I part because of simulation studies that showed it does a better job of reconstructing

gradients of community composition, using data that looks like the hot/tepid/cold plot I made above. Here are some plots from a study by Minchin (1987, *Vegetatio*, An evalutation of the relative robustness of techniques for ecological ordination).



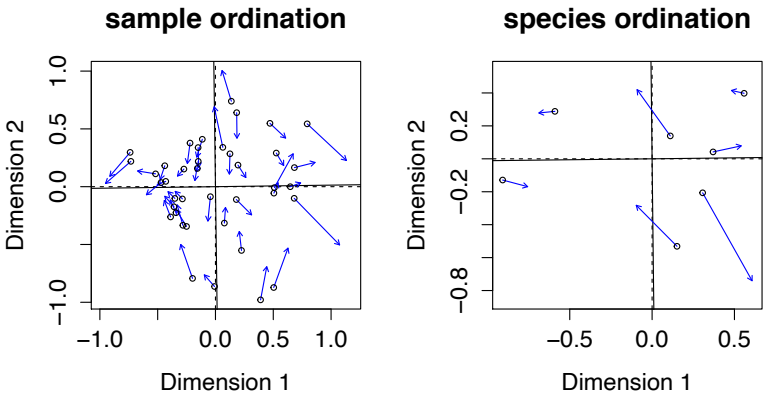
Minchin simulated data where species had unimodal responses to a two-dimensional environmental gradient, like my hot/tepid/cold plot but in two dimensions. E.g., imagine that rainfall increases along one axis and temperature along the other. He then took samples of community composition on a grid in 2D environmental space, and asked whether the different ordination methods could reconstruct this grid. The plot above compares, for one set of simulation parameters: PCA (on normalized data), PCoA (using Bray-Curtis similarity), NMDS (also Bray), and detrended correspondence analysis (another technique that I will not discuss, as NMDS seems to be generally better). As you can see, NMDS does a good job of reconstructing how composition varies along two gradients. PCA and PCoA both capture some of the pattern, but in a warped way.

Studies like the Minchin paper argue that NMDS does a better job of ordinating community data (although I suspect that transforming the abundance data might have helped the other methods look a bit better). The downside of NMDS is that it is primarily trying to maintain the *relative* position of points, rather than represent the magnitude of dissimilarity, and so it is less quantitative, and we don't get a great measure of %variation explained. Personally I might compare NMDS and PCoA, see if the ordinations are similar, and if so use the PCoA as it is more quantitative. But using NMDS as the default is probably a good idea.

If we want to compare ordinations, we can use procrustes as before():

```
par(mfrow = c(1,2))
proc = procrustes(ord, pcoa)
plot(proc, main = 'sample ordination')

proc = procrustes(ord$species, summary(pcoa)$species[,1:2])
plot(proc, main = 'species ordination')
```



Here I extracted both the sample scores (i.e. the observations that are the subject of the ordination), as well as the species scores (which are derived from the sample scores using weighted averaging). And I asked how much the PCoA results had to be rotated/scaled to match the NMDS results. For both plots, it looks like the relative position of the points does not change drastically, though it is certainly not identical between the two methods. If nothing else this suggests the patterns are somewhat robust.

NMDS with groups

I've spent a lot of time explaining how PCoA and NMDS work, now I'll show the kinds of things they're actually useful for. I'll stick with the phytoplankton data, but now I'll include a larger proportion of the community, and more samples. There are 120 samples (one per month for 10 years), and I've decided to use the 35 most common species in the community, as that is 1) a lot of species, and will let us see what happens with some real complexity, but 2) all of those species are present nearly half of the time, so there aren't any really rare species that don't add much info on compositional gradients.

I started by trying a 2D ordination:

```
ord = metaMDS(species.data.use, dist = "bray", trymax = 100, k = 2)
```

This gave a stress of 0.2, which is not amazing but not awful, although it also said 'no convergent solutions', which means the exact results tend to vary based on starting conditions. Because I really only want to plot this data in 2D, I'm inclined

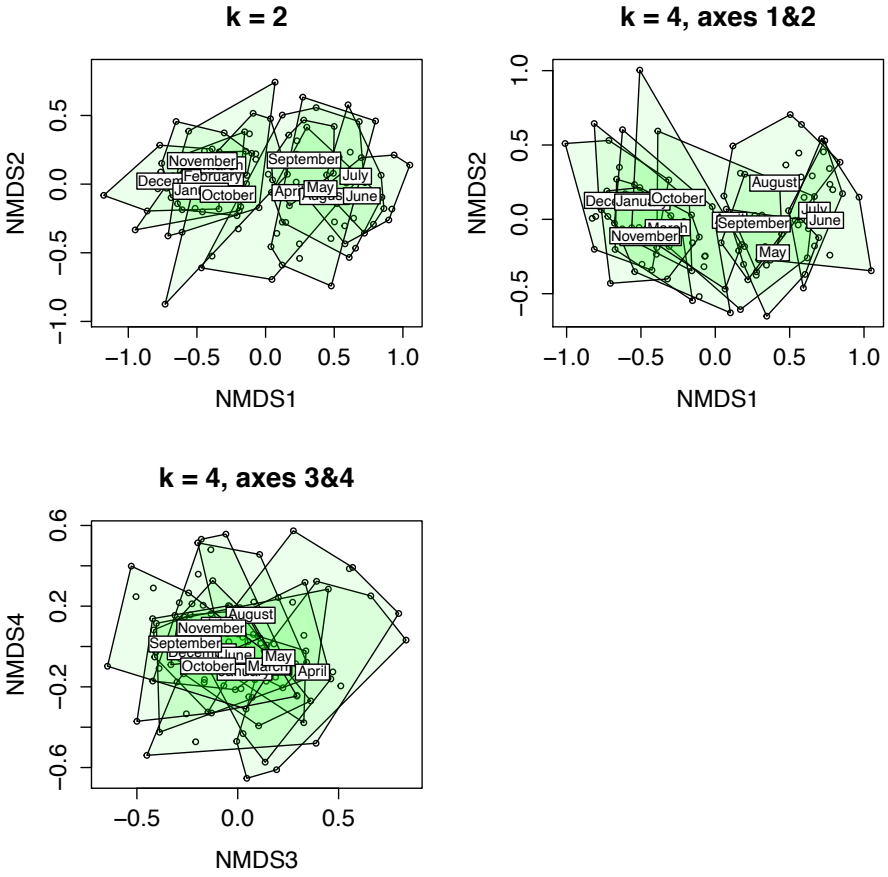
to use this ordination nonetheless, but for pedagogical purposes I'll compare to 3D and 4D ordinations.

```
ord3 = metaMDS(species.data.use, dist = "bray", trymax = 100, k = 3)

ord4 = metaMDS(species.data.use, dist = "bray", trymax = 100, k = 4)
```

Both of these converged, and the 3D version had a stress of 0.15, while the 4D had a stress of 0.12. So clearly the high dimensions allow the ordination to more accurately represent the data, though that will always be the case. How different is the ordination in 2D vs. 4D? This is another tricky question, because *the first two dimensions of the 4D ordination are not the same as the 2D ordination*. This is different from PCA and PCoA. So we don't have an easy way to visually compare the ordinations. Nonetheless, because metaMDS rotates the results using a subequent PCA (yes, this is quite convoluted), in my experience the first two axes are often pretty similar. Instead of comparing them with procrustes, I'll compare them based on something I'm interested in: whether the community composition has a consistent seasonal signal.

```
par(mfrow = c(1,3))
plot(ord, display = "sites", main = 'k = 2')
ordihull(ord, month.use, label = TRUE, col = 'green', border = 'black', alpha
= 20, cex = 0.6, draw = 'polygon')
plot(ord4, display = "sites", main = 'k = 4, axes 1&2')
ordihull(ord4, month.use, label = TRUE, col = 'green', border = 'black', alpha
 = 20, cex = 0.6, draw = 'polygon')
plot(ord4, display = "sites", main = 'k = 4, axes 3&4', choices = c(3,4))
ordihull(ord4, month.use, label = TRUE, col = 'green', border = 'black', alpha
 = 20, cex = 0.6, draw = 'polygon', choices = c(3,4))
```



The first plot shows the convex hulls by month for the 2D ordination. There is not as clear of a pattern as there was for the environmental data (in the previous lecture), but there is definitely a signal of winter (Oct-Mar) vs summer (April-Sep) along the first axis. That is pretty reasonable based on what we already know about these kind of systems, but it is nice to see a clear signal coming out from the ordination. In the next lecture we will see how 'constrained ordination' can be used to more directly visualize compositional variation that is associated with a predictor.

The plot on the top right shows the hulls for the first two axes of the 4D ordination; it looks pretty similar to the first plot, and I bet it would look even more similar if the y-axis was reflected around zero (remember, the orientation is arbitrary). The plot on the bottom left look at the third and fourth axes for the 4D NMDS. Along these axes there doesn't appear to be a strong seasonal signal. So based on these results, which only pertain to this question of seasonality, it doesn't look like we're gaining much from the 4D version, even though it has lower stress.

Those plots were using the ordination of the observations to look for gradients in composition. We may also be interested in the species scores, which help us see how species respond differentially along the same axes. Because the species scores

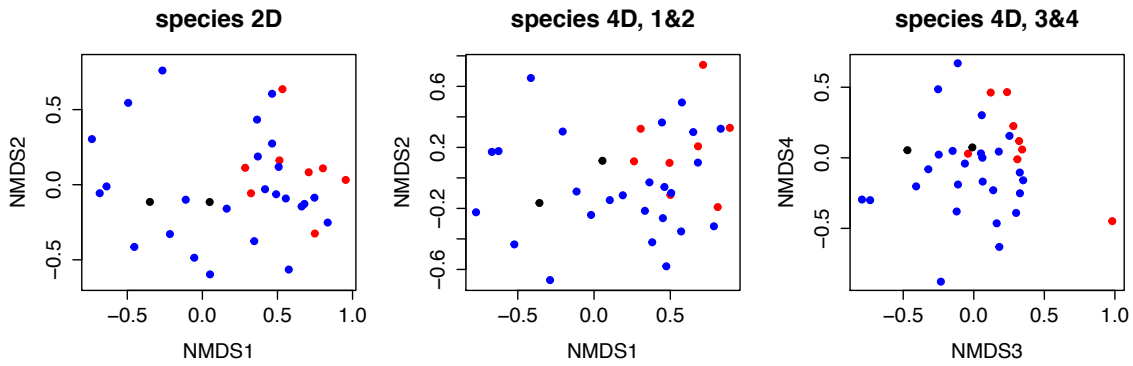
are weighted averages of the sample scores, we can think of them as showing the 'optima' in ordination space, though really there's no requirement that the species show a unimodal pattern, rather it's just the center of mass in terms of a species' abundance.

```
plot(ord, display = "species", type = "n", main = 'species 2D')
points(ord, display = "species", col = c('blue', 'black', 'red')[taxa], pch =
19)

plot(ord4, display = "species", type = "n", main = 'species 4D, 1&2')
points(ord4, display = "species", col = c('blue', 'black', 'red')[taxa], pch =
 19)

plot(ord4, display = "species", type = "n", choices = c(3,4), main = 'species
4D, 3&4')
points(ord4, display = "species", col = c('blue', 'black', 'red')[taxa], pch =
 19, choices = c(3,4))
```

vegan has plotting methods that allow you display species scores with display = 'species'. The sample scores are chosen with display = 'sites', because usually the samples are from different sites, though in this case they are from different times.

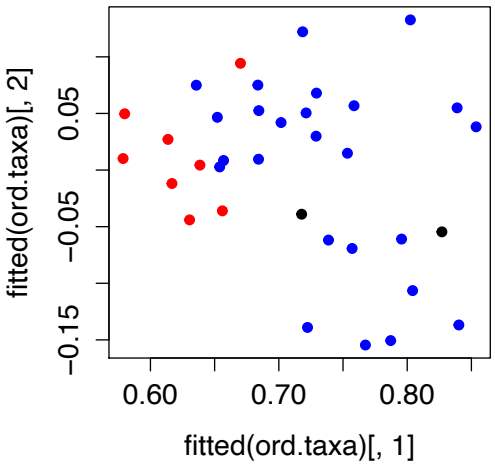


I've colored the species scores based on taxonomy: diatoms (blue), dinoflagellates (red), and other (black). The plot on the left is the 2D ordination. Clearly the dinoflagellates cluster together, but they are not distinct from the diatoms, so much as they overlap with a subset of the diatoms. We see the same patterns in the 4D plot, both along axes 1&2 and along axes 3&4. It's interesting that the dinoflagellates cluster along all four axes in the 4D ordination. Does that mean something? Maybe the additional axes, beyond the first two, help further distinguish communities where dinoflagellates vs. diatoms have been selected by the environmental conditions. Or maybe it's just that the orientation of the axes for an NMDS ordination are arbitrary, because the points would have the same relation to each other regardless of where we draw the axes. We can use procrustes in a different capacity: to rotate the ordination so that is has maximum similarity with an predictor. This could be an environmental predictor, but it could also be a taxonomic predictor, if we restrict it to two levels:

```
ord.taxa = procrustes(as.integer(taxa == 'd'), ord4$species)
```

```
## Warning: X has fewer axes than Y: X adjusted to comform Y

plot(fitted(ord.taxa)[,1], fitted(ord.taxa)[,2], col = c('blue', 'black', 'red
')[taxa], pch = 19)
```



Here I've rotated the species scores ordination so that it aligns as well as possible with a vector that says whether or not the species is a diatom. The function throws a warning because I'm rotating a 4D matrix to be similar to a 1D vector; but that's OK, the function can handle it. Then I use fitted() to get the ordination coordinates on the new (rotated) axes, and plot the first two dimensions of those coordinates. Clearly the first axis of the rotated NMDS now does a great job of distinguishing diatoms from dinoflagellates. This is not too surprising biologically, because dinoflagellates tend to dominate in different conditions (stratified, late summer) than diatoms. And that is consistent with where those groups were positioned on the previous plots (i.e. the summer vs. winter axis). But it is nice to see that we can pull this out of the ordination as well.

NMDS with continuous environmental correlates

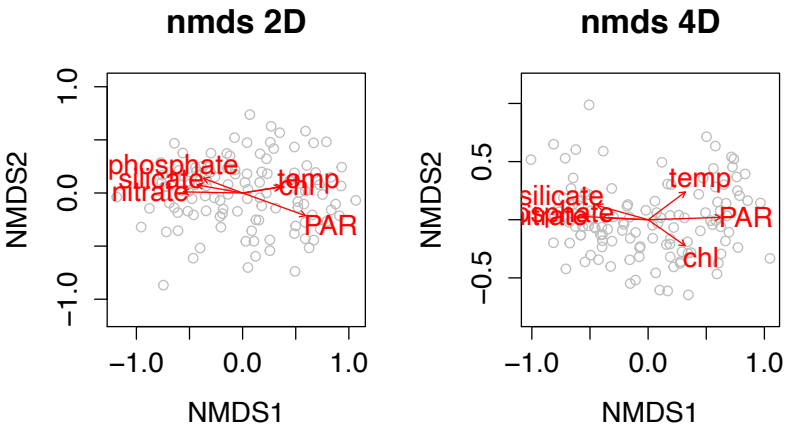
We can also take an ordination plot and see how environmental variables (or other continuous variables) correlate with the ordination axes. This is conveniently done with the envfit() function in vegan:

```
fit = envfit(ord ~ nitrate + phosphate + silicate + PAR + chl + temp, data = e
nviro, na.rm = T)
ordiplot(ord, display = "sites", type = 'n', main = 'nmds 2D')
points(ord, col = 'grey')
plot(fit, col = 'red', arrow.mul = 0.7)

fit = envfit(ord4 ~ nitrate + phosphate + silicate + PAR + chl + temp, data =
enviro, na.rm = T)
ordiplot(ord4, display = "sites", type = 'n', main = 'nmds 4D')
```

```
points(ord4, col = 'grey')
plot(fit, col = 'red', arrow.mul = 0.7)
```

Here I've given envfit() the nmds ordination object with a formula specifying the environmental variables I want to correlate with the ordination. Note: although the formula is convenient for specification, *this is not a linear model*. Rather, the function uses 1) the ordination scores (the location of the points), and 2) the environmental conditions at those points, to find the direction along which the environmental variable increases most steeply. And it plots a vector in that direction, with the length of the vector proportional to how well it correlates with the ordination.



I've done this for both the 2D and 4D ordinations, for comparison. We can see that the general axis of stratification, which we found when doing a PCA on the environmental variables, is also recoverable from the community ordination. In other words, community composition varies along the axis of stratification, even though we did not use the environmental info in the ordination. The 4D ordination also appears to capture some of the spring vs. fall differentiation, which is associated with higher chl and lower temperature in the spring. The envfit function also returns info on the strength and 'significance' of the environmental correlations:

```
fit

##
## ***VECTORS
##
##             NMDS1  NMDS2   r2 Pr(>r)
## nitrate   -0.999 -0.049 0.66  0.001 ***
## phosphate -0.999 -0.050 0.33  0.001 ***
## silicate  -0.963 -0.271 0.42  0.001 ***
## PAR        1.000 -0.030 0.80  0.001 ***
## chl        0.821  0.570 0.31  0.001 ***
## temp       0.799 -0.601 0.33  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
```

```
## Number of permutations: 999
##
## 21 observations deleted due to missingness
```

For each environmental variable, it shows the strength of the loading on the two ordination axes (similar to PCA), as well as the total R^2 for the amount of variation in the environmental variable associated with those axes. It also returns signficance tests. What are these? In order to ask 'is this environmental variable significantly associated with the ordination axes', the function uses a *permutation test*. The logic is simple:

- There is some correlation between the environmental variable and the NMDS scores
- To get a null distribution for that correlation, we can randomly permute the rows of the environmental variable, to break any association with the NMDS scores.
- If we do that many times (e.g. 1000), we can ask 'how often is the correlation as large as the observed correlation?'.

The permutation is similar to the parametric bootstrap we used previously, but now instead of simulation fake data from a model, we are rearranging the raw data to get a null distribution. In general, this is a fine procedure if we don't take it too seriously. Because it's not an explicit statistical model, it has hidden assumptions. For example, if the data are grouped or temporally correlated, but we are not accounting for that in the permutation procedure, then the p-values may not be valid (we will return to this next lecture).

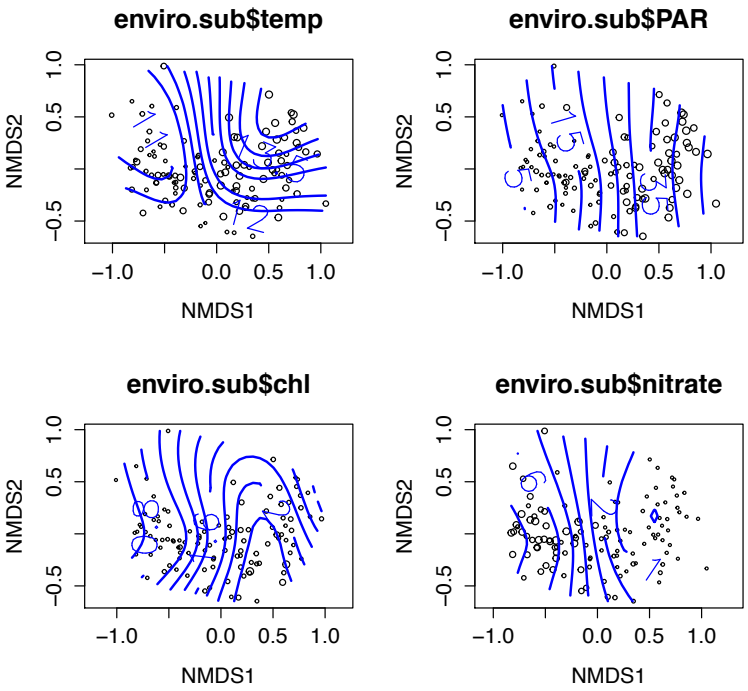
The vectors and correlations returned by envfit() assume that the relationship between the environmental variables and the ordination scores is linear. We can relax this assumption, and fit a GAM where the response is the environmental variable, and the the predictor is a 2D smoother using the ordination scores on the two axes:

```
par(mfrow = c(2,2))
ordisurf(ord4, enviro.sub$temp, bubble = TRUE, labcex = 1.5, col = 'blue', lwd
 = 2)

ordisurf(ord4, enviro.sub$PAR, bubble = TRUE, labcex = 1.5, col = 'blue', lwd
= 2)

ordisurf(ord4, enviro.sub$chl, bubble = TRUE, labcex = 1.5, col = 'blue', lwd
= 2)

ordisurf(ord4, enviro.sub$nitrate, bubble = TRUE, labcex = 1.5, col = 'blue',
lwd = 2)
```



The ordisurf() function uses mgcv behind the scenes to fit a 2D smoother and plot a contour plot of the fitted smoother, with the ordination scores plotted as a bubble plots, with the size of the point proportional to the value of the environmental variable. In the plots above, we can see that the environmental correlations are roughly linear in the way implied by the vectors in the previous plot. Temperature shows a somewhat nonlinear relationship, in that it increase along the top of the plot but less so along the bottom of the plot. These plots for the environmental variables may seem a bit backwards, and indeed they are: we're using the ordination of community composition as the predictor, and the environmental variables as the response, in order to see how the environment covaries with the community ordination. Nonetheless, this can be very helpful for visualizing what may be driving variation in community structure. In the next lecture, we will address this question more directly with methods that explicitly relate a matrix of environmental variation to the matrix of community composition.