7. Direct & Indirect Gradient Analysis

Direct and indirect gradient analysis refers to a case where you have two datasets with variables that have cause-and-effect or mutual influences on each other. This may be a clear case of independent variables (e.g. climate) and a set of dependent variables (e.g. species frequencies), but sometimes that distinction is not so easy to make if you have mutual influence (e.g. herbivore community data vs. plant community data). You want to know how they are related (e.g. herbivore species 3, 5, and 7 tend to co-occur with plant species A, B, D, and F. Or Climate variables 1, 2, 5 influence the frequency of species K, J, and X).

Because you tend to have lots of variables relative to observations, conventional statistical tests are almost guaranteed to fail because of insufficient degrees of freedom due to many variables (columns) relative to independent observations (rows). Therefore, reduction of dimensions is a key step in a gradient analysis, where we try to determine whether the species-environment relationship is higher, than expected due to chance. There are several descriptive and statistical methods available.

A direct gradient analysis ordinates the data according to the independent variable (e.g. climate) and then investigates how the dependent variables (e.g. plant species) correlate to the ordination scores. The problem is that you may miss an important pattern in your plant community data, perhaps one that may be driven by a climate variable that you did not measure.

An indirect gradient analysis ordinates your dependent variable (e.g. community data according to their similarity in species composition). Relationship of the species frequencies with environmental gradients is then investigated by correlating the ordination scores with the environmental variables in the second step.

A constrained analysis utilizes both datasets and finds the 'best possible' relationships (defined in a mathematical sense) between species composition and the environment. (Note that some researchers use the terms differently: using “direct” instead of “constrained” and “indirect” instead of both “direct” and “indirect” as defined above)

7.1. Direct and indirect gradient analysis with NMDS

Direct and indirect gradient analyses are very easy to implement with NMDS in R. Essentially, all you do is adding a second set of vectors to your NMDS. To try this out, I have added a dataset of average tree species frequencies to the climate dataset that we used in the previous exercise.

- Import the dataset AB_Climate_Trees.csv into R and subset the dataset into separate climate and tree frequency datasets. If any of the code below is not clear to you, let the TA or myself explain this to you individually:

```
means=read.csv("AB_Climate_Trees.csv")
fix(means)

rownames(means)=means$ECOSYS  # creates rownames
ecolabels=means$ECOSYS  # creates labels that we need later
trees=means[,11:23]
climate=means[,3:10]
fix(trees)  # check it
fix(climate) # check it
```
Next, we run an NMDS exactly as in the previous lab. For the direct gradient analysis we ordinate the independent variable (i.e. climate). I choose the Mahalanobis distance here, so this is probably exactly what you have done in the previous lab on NMDS.

```r
library(ecodist)
mahal_climate=distance(climate, "mahal") # distance matrix
nmds_climate=nmds(mahal_climate, mindim=2, maxdim=2) #NMDS
scores_climate=nmds.min(nmds_climate) # generate scores

plot(scores_climate, col="white")
text(scores_climate, labels=ecolabels)
vectors=vf(scores_climate, climate, nperm=100)
plot(vectors*0.8, col="red")
```

Here comes the twist: we add a second set of vectors. We use a different color, and we make the vectors a bit longer to avoid crowding: just add a multiplier to the vectors to lengthen or shorten them (1.2)

```r
vectors=vf(scores_climate, trees, nperm=100)
plot(vectors*1.2, ascale=1, col="blue")
```

Alright, now we do it the other way around. We ordinate the community data, then overlay climate vectors. This is an indirect gradient analysis:

```r
library(ecodist)
braycurtis_trees=bcdist(trees) # distance matrix
nmds_trees=nmds(braycurtis_trees, mindim=2, maxdim=2) #NMDS
scores_trees=nmds.min(nmds_trees) # generate scores

plot(scores_trees, col="white")
text(scores_trees, labels=ecolabels)
vectors=vf(scores_trees, trees, nperm=100)
plot(vectors, ascale=1, col="red")
vectors=vf(scores_trees, climate, nperm=100)
plot(vectors, ascale=1, col="blue")
```

Have a look at the direct versus indirect gradient analysis side-by-side. Are you missing structure in the secondary datasets? Give me or the TA some examples of relationships or differences that you can or cannot see in the alternate approaches. If you are familiar with the ecology of Alberta, does all this make good sense? On the next page is the key to the abbreviations to facilitate interpretation:

Ecosystem names: A=alpine, BSA=boreal subarctic, CM=central mixedwood, CP=central parkland, DM=dry mixedwood, DMG=dry mixed grass, FF=foothill fescue, LBH=lower boreal highlands, LF=lower foothills, M=montane, MG=mixedgrass, NF=northern fescue, NM=northern mixedwood, PRP=peace river parkland, SA=subalpine, UBH=upper boreal highlands, UF=upper foothills.

Climate variable names: MAT=mean annual temperature, MWMT=mean warmest month temperature, MCMT=mean coldest month temperature, TD=Jul-Jan temperature difference (continentality), lnMAP=logarithm of mean annual precipitation, lnMSP=logarithm of mean summer precipitation, lnAHM=annual dryness index, lnSHM=summer dryness index.

7.2. Gradient analysis with PCA

You can do gradient analysis with any ordination technique. To get additional vectors you simply have to calculate the loadings, which is the correlation between axis scores and an original variable. Obviously you can also calculate correlations with more variables that come from a second dataset and that were not used for ordination.

Now, since we calculate linear correlations, there is an important assumption of linearity between the ordination scores and the variables represented by vectors (NOT necessarily between the original response and predictor variables). This is also true for the NMDS above, and this is why I log-transformed the precipitation variables for this exercise.

If you have different relationships (e.g. a hump-shaped relationships) they may remain undetected. So the point here is that you are not in danger of violating statistical assumptions – you will simply not be able to detect some relationships. We’ll explore this later with artificial datasets.

- Let’s first do a direct gradient analysis with PCA
  ```r
  output=princomp(climate, cor=T) # PCA
  biplot(output) # Generates a bi-plot with vectors
  ```

- Next we add the second set of environmental vectors by fitting the species data (trees) to the plotted climate component scores (output$score[,1:2])
  ```r
  library(vegan)
  vec1=envfit(output$score[,1:2], trees, permutations=0)
  plot(vec1, col="blue")
  ```

- If you wish, you can also manually calculate a second set of component loadings for the species data that you may report in a table together with the original PCA loadings for climate:
  ```r
  scores12=output$score[,1:2] # get the PC1 and PC2 scores
  scores12_trees=cbind(scores12,trees) # merge with species
  correlations=cor(scores12_trees) # calculate correlations
  correlations12=correlations[3:15,1:2] # the loadings we want
This result is not bad at all for the venerable PCA and linear correlation techniques. Again, have a look at the abbreviations above and you will see that this is an ecologically quite reasonable representation of forest types and climate in Alberta.

7.3. Constrained Gradient Analysis with CANCOR

Canonical correlation analysis does not ordinate your data but can be considered a constrained analysis, i.e. both datasets are used to determine relationship. Technically, we rotate two datasets independently, so that the correlation between the corresponding components is maximized.

Constrained gradient analysis also has a few disadvantages. You will not recognize any structure in either the independent or dependent datasets if the variables are not correlated between the two datasets. I.e. you just pull out the cross-correlated components from the datasets and ignore everything else. This may be a fine objective – you should just be aware of the fact.

Also, this technique assumes linear relationships. That may be OK for detecting curvilinear and even L-shaped relationships, but the technique fails to detect unimodal or "hump-shaped" relationships. Let's work with an artificial dataset to explore this in more detail.

- Download the zip file "Lab7Data.zip", which contains the datasets linear.csv and unimodal.csv. Import, and make a multipanel scatter plot to investigate the relationships. You can see that species 1 is weakly correlated to environment 1 and species 2 and 3 are moderately correlated to environment 3:

```r
linear=read.csv("linear.csv")
fix(linear)
plot(linear[,2:7])
```

- For the unimodal dataset, you see that environments 1 and 2 and species 2 and 3 are linearly correlated, and there are unimodal relationships between environments 1 and 2 and species 1, and environment 3 and species 2 and 3:

```r
unimodal=read.csv("unimodal.csv")
fix(unimodal)
plot(unimodal[,2:7])
```
- Install the package CCA, subset the data, and execute a canonical correlation analysis. The first command gives you the r-values of the canonical correlations and the second command writes a number of additional statistics into an output file. From this output file we can call the loadings of the original variables with the canonical functions

\[
\text{env}=\text{linear}[2:4] \quad \# \text{ subset data} \\
\text{spec}=\text{linear}[5:7]
\]

```
library(CCA)
cancor(env, spec) \# canonical correlation analysis
output=cc(env, spec)
```

# significance tests
```
cor.test(output$scores$xscores[,1],output$scores$yscores[,1])
cor.test(output$scores$xscores[,2],output$scores$yscores[,2])
cor.test(output$scores$xscores[,3],output$scores$yscores[,3])
```

# loadings
```
output$scores$corr.X.xscores
output$scores$corr.Y.yscores
```

This is how the results from canonical correlation analysis are reported: The loadings tells you what the nature of the correlations are. The interpretation of loadings on the table to the right would be:

```
<table>
<thead>
<tr>
<th></th>
<th>CAN1</th>
<th>CAN2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spec1</td>
<td>-0.42</td>
<td>0.88</td>
</tr>
<tr>
<td>Spec2</td>
<td>-0.84</td>
<td>-0.04</td>
</tr>
<tr>
<td>Spec3</td>
<td>-0.98</td>
<td>-0.19</td>
</tr>
<tr>
<td>Env1</td>
<td>-0.44</td>
<td>0.85</td>
</tr>
<tr>
<td>Env2</td>
<td>-0.66</td>
<td>0.52</td>
</tr>
<tr>
<td>Env3</td>
<td>-0.99</td>
<td>-0.10</td>
</tr>
</tbody>
</table>
```

"We have two significant canonical functions CAN1 and CAN2. The first indicates that Spec2 and Spec3 are at lower frequencies when Env3 is small (both negative). Spec1 appears to prefer high values of Env1."

In the bottom line you can indicate the significance and also the correlation coefficients if you like. Since CAN3 was not significant, we won't discuss it.

- See if you can detect the hump-shaped relationships in the unimodal.csv dataset.

**7.4. CANCOR with SAS**

- And you may also do the analysis in SAS if you have time. In the output window, look for the significance of the eigenvalue, which tells you how many correlated dimensions exist in the dataset. The loadings are called canonical structure.

```
proc cancorr data=linear;
  var env1 env2 env3;
  with spec1 spec2 spec3;
run;
```
7.5. Constrained Gradient Analysis with CCA

Canonical correspondence analysis is another type of constrained gradient analysis. The procedure executes a distance-based ordination with the dependent variable (e.g. community data), but at the same time it also considers the independent data (environmental variable) to maximize the variance explained by environmental data in the ordination axes CCA1, CCA2, etc.

The method is generally advertised as being able to handle unimodal relationships and even categorical data.

- Here is the required R code. Much of this should make intuitive sense now. If not, ask the TA or me to explain:

```R
library(vegan)
output = cca(spec ~ Env1 + Env2 + Env3, data = env)
plot(output, choices = c(1, 2), scaling = 3)
anova(output)
anova(output, by = "terms", permu = 200)
```

The output gives you the total variance in the species dataset (total inertia) and variance explained by the environmental data (constrained inertia) as well as the residual, unexplained variance (unconstrained inertia). The Eigenvalues indicate how much of the variation is explained by individual axes of an ordination that you can plot. You have to do the math yourself to translate all this into percentage values.

The subsequent ANOVA tells you if significant relationships exist, and the second ANOVA breaks down which environmental variables are significantly associated with the species.

7.6. Additional exercises with CCA

- Try how the analysis fares with the unimodal dataset.

- Analyze the Climate_trees.csv dataset (the plot on the right is certainly not a bad result if you know a bit about the ecology of Alberta):

- The technique is also capable of testing and ordinating categorical environmental variables Try the artificial dataset categorical.csv with two models for comparison: (1) spec ~ Env1 + Env2 + Env3 and (2) spec ~ Cat1 + Env2 + Env3, where the first environmental variable is replaced by a categorical equivalent (which could be something like soil type).
7.7. Constrained Gradient Analysis with RDA

Redundancy analysis is another type of constrained gradient analysis. This procedure executes a rotation-based ordination in order to find linear combinations of independent data (e.g. environmental variables) to represent as much variance in the dependent variables (e.g. community data) as possible. While CCA focuses more on species composition it struggles to detect relationships where all responses to a predictor gradient are the same. Alternatively RDA will be able to detect gradients in which all species are similarly associated (e.g. all species are positively correlated along an environmental gradient).

Like CANCOR and CCA, RDA assumes there is a linear dependence of the response variables (dependent data) on the explanatory variables (independent data). Like CCA, RDA can include categorical data.

- Download the zip file "Lab7Data.zip" from the class website. It will contain the data file "fisheries.csv", which is an artificial dataset representing fish abundances at 10 sites along a tropical reef transect. The first 3 sites are on "sand" and the others alternate between "coral" and "other substrate". Water depth at the sites is also provided. In this example the 6 species are the response variables (dependent variables) and the site characteristics are the predictor variables (independent variables).

- Here is the required R code. Again, much of this should make intuitive sense now. If not, ask the TA or me to explain:

  ```r
  fish = read.csv("fisheries.csv")
  fish.spec = fish[,2:7]  # subset data
  fish.env = fish[,8:11]
  library(vegan)
  output2 = rda(fish.spec~Depth+Sand+Coral+Other,data=fish.env)
  output2
  ```

  Like CCA the RDA output gives you the total variance in the species dataset (total inertia) and variance explained by the environmental data (constrained inertia) as well as the residual, unexplained variance (unconstrained inertia). The Eigenvalues indicate how much of the variation is explained by individual axes of an ordination that you can plot. You have to do the math yourself to translate all this into percentage values.

  R will also tell you if some of your constraints were aliased (removed) because they were collinear (i.e it is redundant to include both). R will always take the variables in their column order, so that dropped variables will be the latter columns. When we plot the biplot you will notice that the "other substrate" column is not considered as an environmental predictor in this RDA. If your variables are collinear and you want to make sure a particular predictor variable is considered in your RDA make sure that variable is one of the first columns included in the predictor variables.

- Again you can use `anova()` to tell you if significant relationships exist, and the second ANOVA breaks down which environmental variables are significantly associated with the species:

  ```r
  anova(output2)
  anova(output2, by="terms", permu=200)
  ```

- With RDA it is possible to use response variables that are measured in different units. However if this is the case, the dependent data must be centered and standardized before executing the analysis. In our example the species measurements are in the same units, but
if your response variables are measured in different scales you can include the option `scale=TRUE` in the `rda()` function.

- Finally we can look at a biplot of the RDA results to visualize how the fish species are associated with the predictor environmental variables

```r
plot(output2, type="n", scaling=2)
segments(x0=0, y0=0,
         x1=scores(output2, display="species", scaling=2)[,1],
         y1=scores(output2, display="species", scaling=2)[,2])

text(output2, display="sp", scaling=2, col=2)
text(output2, display="bp", scaling=2, row.names(scores(output2, display="bp")),
      col=3)
text(output2, labels=fish$Site, display="sites", scaling=2)
```

- Here we build the RDA biplot using the RDA scores and additionally include both the species name (sp) and environmental variables (bp) as vectors.

- The interpretation of these vectors should make intuitive sense to you not. If not, ask the TA or me to explain. For starters all species were found to dislike environments characterized by sand.