Indirect ordination and implementation in R

Overview of indirect ordination

Ordination, in its many forms, is a process that reduces dimensionality of a multivariate data set and allows for visualization of the main trends in the data in a continuous form. This is accomplished by projecting high dimensionality data onto a bivariate graph with axes that are essentially independent of each other and explain a large fraction of the variation in the high dimension data. Distances between objects in the bivariate ordination space represent similarity or difference in descriptors amongst the objects. Indirect (or unconstrained) ordination refers to ordination that relies solely on an association matrix generated from an object x descriptor matrix. No additional information is incorporated into the ordination process and the structure of the descriptors amongst objects dictates the major axes or gradients of variation amongst objects.

Classifying ordination methods

A multitude of ordination methods exist and each method varies in the underlying algorithm, the distance that is preserved, and the types of data allowed by the algorithm. The underlying algorithm for most indirect ordination methods is eigenvector-based, but we will discuss one non-eigenvector- based method. The association metric used for pairwise comparison of objects or descriptors will determine the distance that is preserved by most ordination methods and also dictates the type of data (quantitative, semi-quantitative, or binary) used.

Common indirect ordination methods and their implementation in R

1) Principle Components Analysis (PCA) is an eigenvector-based method that preserves Euclidean Distances and therefore requires quantitative data. The association metric used is correlation coefficients or variances and covariances amongst objects, which means linear relationships between objects and descriptors are assumed. Also remember that the use of correlations or variances and covariances means that double zeros are "counted" in pairwise comparisons of objects.

Essentially, PCA completes a series of axes rotations sequentially identifying new axes (principle components) that are orthogonal to each other and capture maximum remaining variance in the data cloud. These principle components make up the new coordinate system in which the objects can be plotted in reduced dimensionality. Both objects and descriptors can be plotted in this coordinate space in what is commonly referred to as a "biplot". When plotting both objects and descriptors a choice must be made whether the axes are scaled to better represent the objects or the descriptors. When interpreting the biplot, the relative position of objects indicates their similarity and the angle formed by a vector connecting the origin and a descriptor indicates the contribution of each descriptor to similarity amongst objects. Descriptors with similar angles make similar contributions to the objects. The principle components are scaled eigenvectors and therefore the eigenvalues associated with each principle component indicate the variance captured by each component. These can be scaled to a total of 100 and then provide the percent variance explained by each principle component.

PCA can be implemented in R using a number of functions, including `prcomp()`, `princomp()`, and `rda()`. These all generate objects of similar structure and a biplot of the resulting PCA object can be plotted with the `plot()` function. Be sure you are aware of whether the function you choose is using correlations or variances and covariances, but all functions can use either association metric.

A common question with PCA is "how many principle components to consider?". There is no statistical means by which to make this decision, but some useful guidelines exist. One criterion is the Kaiser-Guttman criterion, which says to include all principle components whose eigenvalue is greater than the mean of all eigenvalues.

2) Principle Coordinates Analysis (PCoA) is an eigenvector-based that preserves any chosen distance and therefore can handle quantitative, semi-quantitative, or binary data. PCoA is essentially a generalized version of PCA that allows the use of any association metric. In fact, a PCoA using a Euclidean distance matrix will give the same results as a PCA using a variance-covariance association matrix. PCoA is useful because of the flexibility in association metric provided and the metric nature of the principle coordinates.
PCoA can be used to consider objects (Q mode) or descriptors (R mode) depending on what association matrix is provided, but biplots of both objects and descriptors are not a natural outcome of the analysis. However, a biplot can be generated post hoc by considering correlation amongst objects and descriptors. Some problems can occur when non-Euclidean distances are forced into Euclidean space by the algorithm in PCoA. This problem manifests itself in the form of negative eigenvalues and will only occur for certain datasets with certain association metrics. It is often the case that the first few principle coordinates are robust to this issue, but transformations to the association matrix (square root transform or adding a small constant to the association matrix) can address this issue when it arises.
PCoA is implemented in R using the `cmdscale()` function or `pcoa()` from the `ape` package. The function `wascores()` can be used to generate associations between descriptors and object-based ordination axes.

3) Correspondence Analysis (CA) is a third eigenvector-based ordination method. CA preserves chi-squared distances and assumes, or is at least appropriate for, unimodal distribution of descriptors along environmental gradients. The data used for CA must be homogeneous in its units and contains positive integers or zeros. Importantly, the chi-squared distance ignores double zeros and CA is therefore commonly used with both quantitative and binary species data. As in PCA and PCoA, the resulting CA axes are orthologous and ranked in order of variance explained, but to estimate variance explained eigenvalues associated with each axis is divided by the total inertia, which is an outcome of the underlying matrix of associations within the CA algorithm. CA is capable of directly generating a biplot of objects and descriptors. As in PCA, two scalings of the biplot, one better suited for objects and the other better suited for descriptors, can be used for the biplot. Unlike inPCA, the objects and descriptors in CA are represented by points in the biplot. The Kaiser-Guttman criterion can be used with CA to select an appropriate number of axes to consider.
A potential side effect of attempting to represent non-linear, often unimodal, distributions of descriptors (species) across objects (sites) in the Euclidean, two-dimensional space of an ordination is something called “the horseshoe effect”. This issue can arise in any eigenvalue-based ordination technique, but has been treated most explicitly in the context of CA. The horseshoe effect can be accounted for with an approach called Detrended Correspondence Analysis (DCA), but much debate has occurred surrounding the utility of detrending and in general it is not widely recommended. In theory, the horseshoe effect generates a second axis that is a quadratic function of the first axis and so the second axis can be detrended by using the residuals of a quadratic function of first axis scores fit to second axis scores.
CA is implemented in R using the `cca()` function in the `vegan` package. One can also investigate correlations amongst environmental characteristics of objects (sites) and the CA, or other, ordination axes using the function `envfit()` in the `vegan` package.

4) Nonmetric Multidimensional scaling (NMDS) is a non-eigenvector-based method and therefore the sequential axes do not maximize the remaining variance explained. NMDS does not focus on preserving distances amongst objects, but rather focuses on the ordering of objects relative to each other along a specified, small number of axes. As a result of this order-based approach and absence of distance conservation, the ordination can be rotated, scaled, and inverted along its axes without changing the overall interpretation. The algorithm that produces an NMDS

picks a starting distribution of objects in the specified number of dimensions (usually 2 or 3) and estimates a value describing how well the distances in the reduced dimension space reflect the distances in the provided association matrix. This value, called the stress, varies between 0 and 1. The algorithm then iteratively alters the distribution of objects in the reduced space in an attempt to minimize the stress value. The algorithm is susceptible to being trapped in a local minimum, as with any complex optimization process, and so it is recommended that multiple starting distributions of objects. Because NMDS is not attempting to represent multidimensional distances in Euclidean space, a 2D or 3D NMDS ordination is often less deformed than many of the eigenvector based approaches. However, this comes at a tradeoff with the metric nature and interpretation of the ordination axes.

NMDS can be implemented in R using `isoMDS()` from the `vegan` package, which runs one optimization series, and `metaMDS()`, which runs a number of optimization series in hopes of finding the globally optimum ordination.