Analysing multivariate abundance data using gllvm

Jenni Niku

2019-08-13

This document illustrates the main features and functionality of the **gllvm** package, developed for the analysis of multivariate abundance data. Examples are provided in order to show how generalized linear latent variable models (GLLVM) can be applied to data.

Multivariate abundance data, consisting of observations of multiple interacting species from a set of samples, are often collected in ecological studies to characterise a community or assemblage of organisms. The term "abundance" is taken here to mean counts, presence-absence records, biomass data or any other measure of the extent to which a species may be present at a site.

GLLVM extends the basic generalized linear model to multivariate data using a factor analytic approach, *i.e.* incorporating a small number of latent variables for each site accompanied by species specific factor loadings to model correlations between responses. These latent variables have a natural interpretation as ordination axes, but with additional capacity, *e.g.* predicting new values, controlling for known environmental variables, using standard model selection tools to choose number of ordination axes.

Generalized linear latent variable models

A multivariate abundance dataset can be defined by a matrix of abundances, with n rows (usually sites) and m columns of responses (usually species). Denote the abundance of the jth species at the ith site as y_{ij}. A set of k environmental variables, or experimental treatments, may also be recorded at each site and stored in the vector $\boldsymbol{x}_i = (x_{i1}, ..., x_{ik})$. A GLLVM regresses the mean abundance μ_{ij} against environmental variables and a vector of $d < m$ latent variables, $\boldsymbol{u}_i = (u_{i1}, ..., u_{id})'$:

$$g(\mu_{ij}) = \eta_{ij} = \alpha_i + \beta_{0j} + \boldsymbol{x}_i' \boldsymbol{\beta}_j + \boldsymbol{u}_i' \boldsymbol{\theta}_j,$$

where $\boldsymbol{\beta}_j$ and $\boldsymbol{\theta}_j$ are vectors of species specific coefficients related to the covariates and latent variables, respectively. The latent variables $\boldsymbol{u}_i$ can be thought of as unmeasured environmental variables, or as ordination scores, capturing the main axes of covariation of abundance (after controlling for observed predictors $\boldsymbol{x}_i$). We assume these latent variables are independent across sites and standard normally distributed. The parameters β_{0j} are species specific intercepts, while α_i are optional site effects which can be chosen as either fixed or random effects ($\alpha_i \sim N(0; \sigma^2)$). If row effects α_i are included, all other terms in the model can be subsequently interpreted as modelling relative abundance or compositional effects (Hui et al., 2015).

If q trait covariates $\boldsymbol{t}_j = (t_{i1}, \ldots, t_{iq})^\top$ are also recorded, we can use them to help explain inter-specific variation in environmental response. This leads to an extension of the so-called "fourth corner model" where multivariate abundance is regressed against a function of traits and environment, and the environment-trait interactions represents the fourth corner association between traits and environment. The associated fourth corner GLLVM then has mean model:

$$g(\mu_{ij}) = \eta_{ij} = \alpha_i + \beta_{0j} + \boldsymbol{x}_i^\top \boldsymbol{\beta}_e + (\boldsymbol{t}_j \otimes \boldsymbol{x}_i)^\top \boldsymbol{\beta}_I + \boldsymbol{u}_i^\top \boldsymbol{\theta}_j,$$

where $\boldsymbol{\beta}_e$ is a vector of main effects for environmental covariates, and $\boldsymbol{\beta}_I$ are the fourth corner coefficients. A main effect for traits is not included, because main effects on abundance across species are absorbed by the intercept term β_{0j}. This model assumes that all inter-specific variation in response to covariates is mediated by species, which reduces the number of parameters related to covariates.

1

A difficulty fitting the GLLVM is that the $\boldsymbol{u}_i$'s are unobserved and we must integrate over their possible values. Specifically, the log-likelihood function we wish to maximise has the form

$$l(\boldsymbol{\Psi}) = \sum_{i=1}^{n} \log(f(y_{ij}, \boldsymbol{\Psi})) = \sum_{i=1}^{n} \log \left(\int_{\mathbb{R}^d} \prod_{j=1}^{m} f(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\Psi}) f(\boldsymbol{u}_i) d\boldsymbol{u}_i, \right),$$

where $\boldsymbol{\psi}$ includes all model parameters. In this expression we have assumed abundances are independent across sites and any correlation across responses is captured by the latent variables $\boldsymbol{u}_i$. Thus conditional on $\boldsymbol{u}_i$, the y_{ij} are independent of each other within sites.

The **gllvm** package overcomes these computational problems using three key innovations:

- Maximising an approximation to the log-likelihood that is (almost completely) closed form. We provide two ways to do this – using Gaussian variational approximations for overdispersed counts, binary and ordinal responses, or using Laplace approximations for other exponential family distributions when a fully closed form variational approximation cannot be obtained *e.g.* biomass data can be modelled by the Tweedie distribution.

- Parameter estimation makes use of automatic differentiation software in C++ to accelerate computation times, via the interface provided by the R package TMB.

- Careful choice of starting values. In particular, we use a factor analysis on Dunn-Smyth residuals to obtain starting values close to the anticipated solution, optionally, with jittering to check the sensitivity of the approach.

In the next examples we illustrate the usage of the gllvm package using `antTraits` data from the `mvabund` package. The data consists of counts of 41 ant species measured at 30 sites across south-east Australia, along with records of five environmental variables and five trait variables for each species. The package and the data can be loaded as follows.

```
library(gllvm)
#> Loading required package: TMB
#> Loading required package: mvabund
#>
#> Attaching package: 'gllvm'
#> The following objects are masked from 'package:stats':
#>
#>     predict, simulate
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- scale(as.matrix(antTraits$env))
TR <- antTraits$traits
```

Data input

Main function of the **gllvm** package is `gllvm()`, which can be used to fit GLLVMs for multivariate data with the most important arguments listed in the following:

```
gllvm(y = NULL, X = NULL, TR = NULL, data = NULL, formula = NULL, num.lv = 2, family,
      method = "VA", row.eff = FALSE, offset = NULL, Power = 1.5, starting.val = "res", ...)
```

Data input can be specified using the "wide format" matrices via y, X and TR arguments.

```
# Model without predictors:
gllvm(y, family = "negative.binomial")
```

```r
# Model where environmental variables, that is, all variables in X are included:
gllvm(y, X, family = "negative.binomial")
```

```r
# Fourth corner model, where all main effects for environmental variables in X and
# all interactions between variables in X and variables in TR are included:
gllvm(y, X, TR, family = "negative.binomial")
```

An alternative way is to use the long format via `data` argument along with `formula`. Notice that `formula` can also be used for model specification when data is given in wide format. In the following lines the ant data is reorganised to a long format.

```r
yX <- reshape(data.frame(cbind(y, X)), direction = "long", varying =
              colnames(y), v.names = "y", timevar = "sp")
TR2 <- data.frame(sp = 1:41, TR)
datalong <- merge(yX, TR2, by = "sp")
datalong[1:3, ]
#>   sp Bare.ground Canopy.cover Shrub.cover Volume.lying.CWD
#> 1  1  -0.3926017    0.2290869  -0.5748133        0.4373291
#> 2  1  -0.8007744   -0.4475411  -0.8053533       -0.2309758
#> 3  1  -0.3585873   -0.4475411   2.9293959       -0.3609125
#>   Feral.mammal.dung y id Femur.length No.spines Pilosity Polymorphism
#> 1        -0.1755352 0  1   -0.3827814         0        2            1
#> 2         1.5171255 0  2   -0.3827814         0        2            1
#> 3        -0.9278288 0  3   -0.3827814         0        2            1
#>   Webers.length
#> 1      1.094225
#> 2      1.094225
#> 3      1.094225
```

The model can then be fitted as

```r
# Model without predictors:
gllvm(formula = y ~ 1, data = datalong, family = "negative.binomial")
```

```r
# Model with environmental variables Bare.ground and Shrub.cover as predictors
gllvm(formula = y ~ (Bare.ground + Shrub.cover), data = datalong,
      family = "negative.binomial")
```

The number of latent variables can be defined using the argument `num.lv`. The default number of latent variables is two. Other important arguments in the `gllvm` call are `row.eff` for defining the type of row effects (none, fixed or random), `offset` for potential inclusion of offsets, `Power` for defining the power parameter of the Tweedie distribution and `starting.val` for judicious choice of starting values for the latent variables.

List of available distributions with the mean, $E(y_{ij})$, and mean-variance, $V(\mu_{ij})$, functions, estimation methods and link functions for various response types are in the table below.

Response	Distribution	Method	Link	Description
Counts	Poisson	VA/LA	log	$E(y_{ij}) = \mu_{ij}$, $V(\mu_{ij}) = \mu_{ij}$
	NB	VA/LA	log	$E(y_{ij}) = \mu_{ij}$, $V(\mu_{ij}) = \mu_{ij} + \phi_j \mu_{ij}^2$, where $\phi_j > 0$ is a dispersion parameter
	ZIP	LA	log	$E(y_{ij}) = (1 - p_j)\mu_{ij}$, $P(y_{ij} = 0) = p_j$, $V(\mu_{ij}) = \mu_{ij}(1 - p_j)(1 + \mu_{ij}p_j)$
Binary	Bernoulli	VA/LA	probit	$E(y_{ij}) = \mu_{ij}$, $V(\mu_{ij}) = \mu_{ij}(1 - \mu_{ij})$
		LA	logit	

Response	Distribution	Method	Link	Description
Biomass	Tweedie	LA	log	$E(y_{ij}) = \mu_{ij}$, $V(\mu_{ij}) = \phi_j \mu_{ij}^{\nu}$, where $1 < \nu < 2$ is a power parameter and $\phi_j > 0$ is a dispersion parameter
Ordinal	Multinomial	VA	probit	Cumulative probit model
Normal	Gaussian	VA/LA	identity	$E(y_{ij}) = \mu_{ij}$, $V(y_{ij}) = \phi_j^2$

Model-based ordination

GLLVMs can be used as a model-based approach to unconstrained ordination by including (*e.g.*) two latent variables in the model but no predictors. The corresponding ordination plot then provides a graphical representation of which sites are similar in terms of their species composition. Such a model can be fitted to the ant data using the function `gllvm` as below. We will consider two count distributions for the data – the Poisson and negative binomial (NB).

```
fitp <- gllvm(y, family = poisson())
fitp
#> Call:
#> gllvm(y = y, family = poisson())
#> family:
#> [1] "poisson"
#> method:
#> [1] "VA"
#>
#> log-likelihood:  -2128.633
#> Degrees of freedom:  122
#> AIC:  4501.267
#> AICc:  4178.557
#> BIC:  4672.213
```

```
fit_ord <- gllvm(y, family = "negative.binomial")
fit_ord
#> Call:
#> gllvm(y = y, family = "negative.binomial")
#> family:
#> [1] "negative.binomial"
#> method:
#> [1] "VA"
#>
#> log-likelihood:  -1895.086
#> Degrees of freedom:  163
#> AIC:  4116.171
#> AICc:  3717.186
#> BIC:  4344.566
```

As a model-based ordination method, GLLVMs offer tools for the model checking in contrast to the classical algorithm-based ordination methods such as nMDS. For example, the default printout above includes information criteria, which all suggest that the NB distribution is a better choice than the Poisson distribution for modelling the response.

In addition to information criterias, we can use residual plots for diagnosing model fit. As a default, `plot` function utilizes `residual` fuction to calculate Dunn-Smyth residuals, also known as randomized quantile

residuals, and then prints five residual diagnostic plots, including a plot of residuals against linear predictors of fitted values, a Normal Q-Q plot of residuals with a simulated point-wise 95% confidence interval envelope, residuals against row index and column index and scale-location plot. With an argument `which` one can define which plots will be produced.

The residual diagnostics below for the Poisson model shows some overdispersion in residuals, in particular, a telltale fan-shape in the plot of residuals against fitted values. These issues are largely resolved in the NB model. Note that the latent variables in the model provide some capacity to account for overdispersion, so overdispersed counts do not always require us to move beyond the Poisson distribution, although there is clear evidence of such a need in this example.

```
# Plot residuals for the Poisson model
par(mfrow = c(3, 2), mar = c(4, 4, 2, 1))
plot(fitp, var.colors = 1)
```
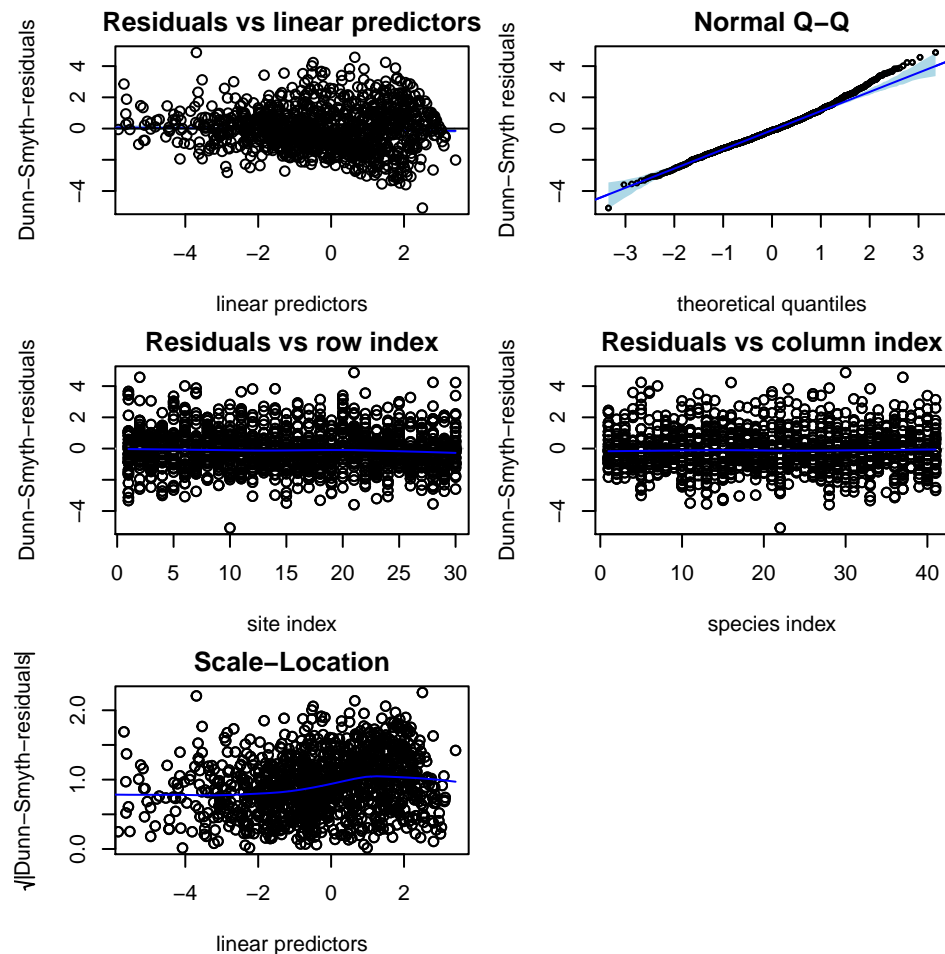


```
# Plot residuals for the NB model
par(mfrow = c(3, 2), mar = c(4, 4, 2, 1))
plot(fit_ord, var.colors = 1)
```
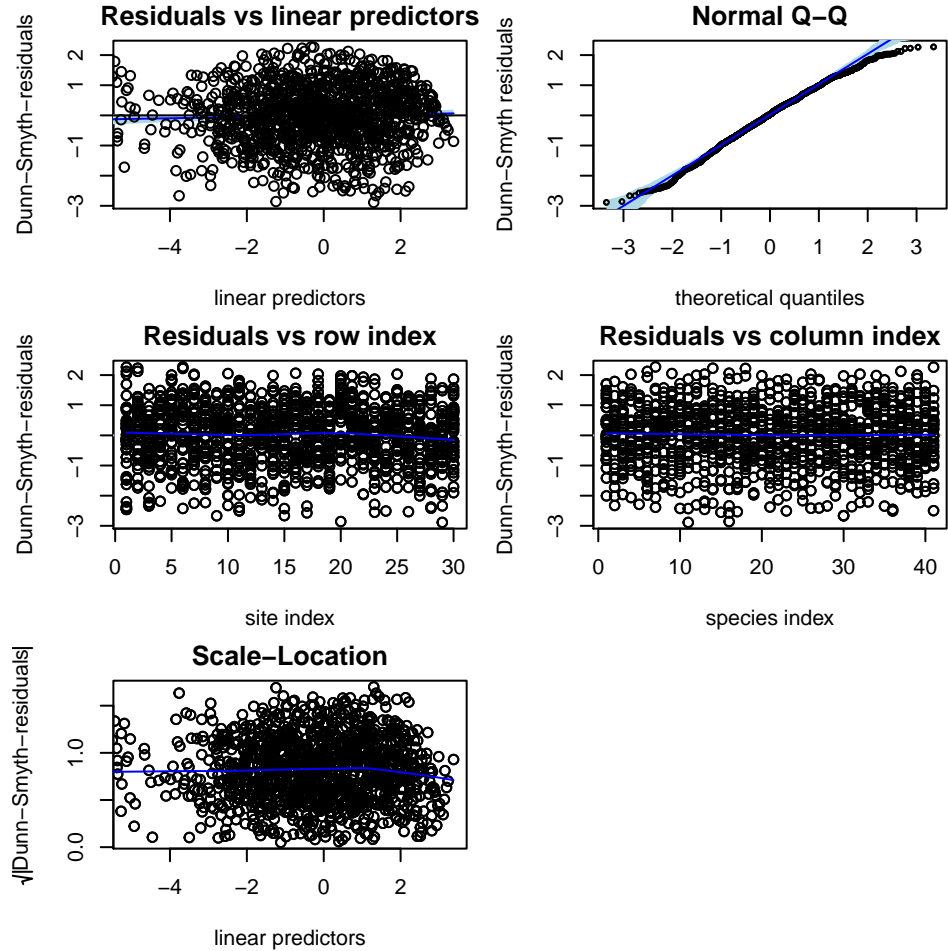
Once an appropriate model has been established for the data, we can construct an ordination as a scatter plot of the predicted latent variables via the `ordiplot` function. The species with the largest factor loadings (largest norms, $||\boldsymbol{\theta}_j||$), and hence most strongly associated with ordination scores, can be added using the logical argument `biplot`, leading to a biplot for finding indicator species corresponding to specific sites. The `ind.spp` argument defines the number of species to be plotted.

The command below creates the biplot and the ordination plot based on the GLLVM fitted to the `antTraits` data. We can see one large cluster of sites on the top with many indicator species, and few smaller clusters with only few indicator species *e.g.* sites 12–15. Notice that the rotation and also the scales of the two plots below slightly differ as in order to produce the biplot the predicted latent variables and their loadings are rotated and scaled so that the factor loadings of the species are in the same direction with the sites where they are most abundant.

```
ordiplot(fit_ord, biplot = TRUE, ind.spp = 15, xlim = c(-3, 3), ylim = c(-2, 1.6),
         main = "Biplot")
ordiplot(fit_ord, biplot = FALSE, ind.spp = 15, xlim = c(-3, 3), ylim = c(-2, 1.6),
         main = "Ordination plot")
```

Biplot

Ordination plot
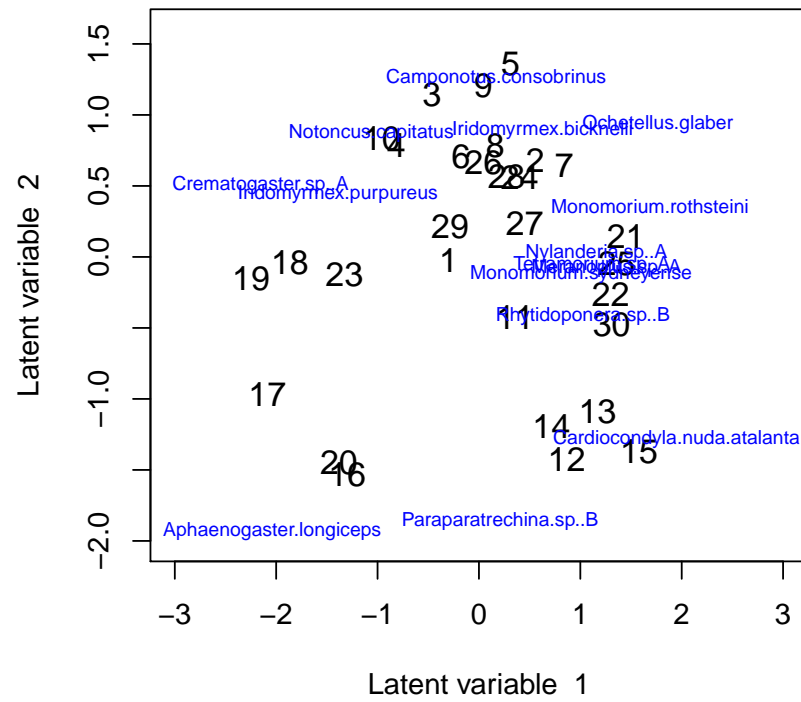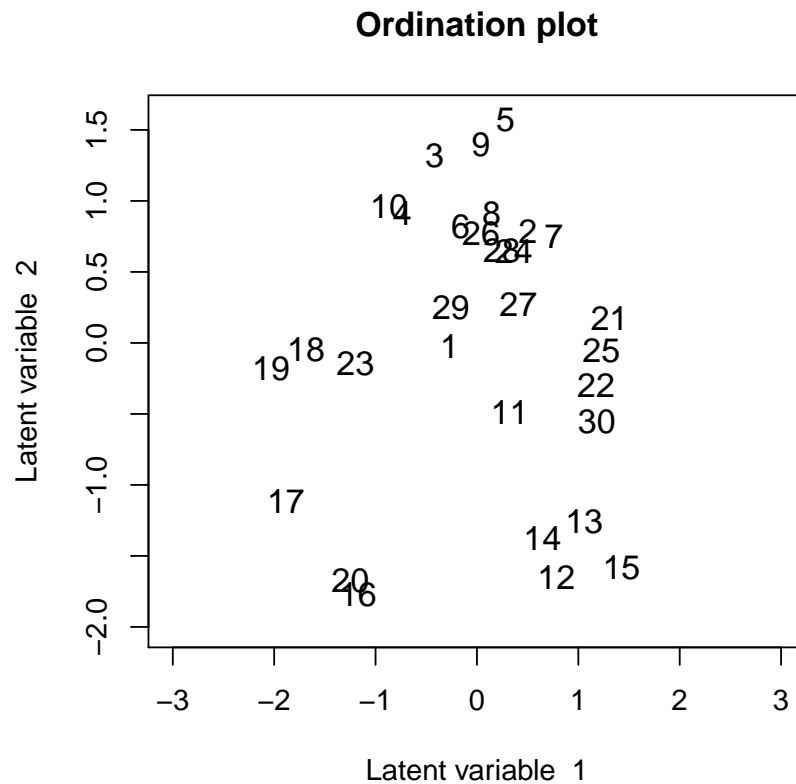
Sometimes species names in the biplot might be hard to read, especially when species names are long or if they are close to each other. In order to make the figure easier to read we can change the sizes of the species names in the biplot using `cex.spp`. If species or site names are too close to each other we can use `jitter = TRUE` in order to add some random variation in the species and site locations. We can also shorten the species names. This can also be done after the model is fitted by directly changing the row names in the loading matrix, see the example below.

```r
rownames(fit_ord$params$theta) <- paste("spp", 1:ncol(fit_ord$y), sep = "")
ordiplot(fit_ord, biplot = TRUE, ind.spp = 15, xlim = c(-3, 3), ylim = c(-2, 1.6),
         main = "Biplot", jitter = TRUE, cex.spp = 0.8)
```

Biplot
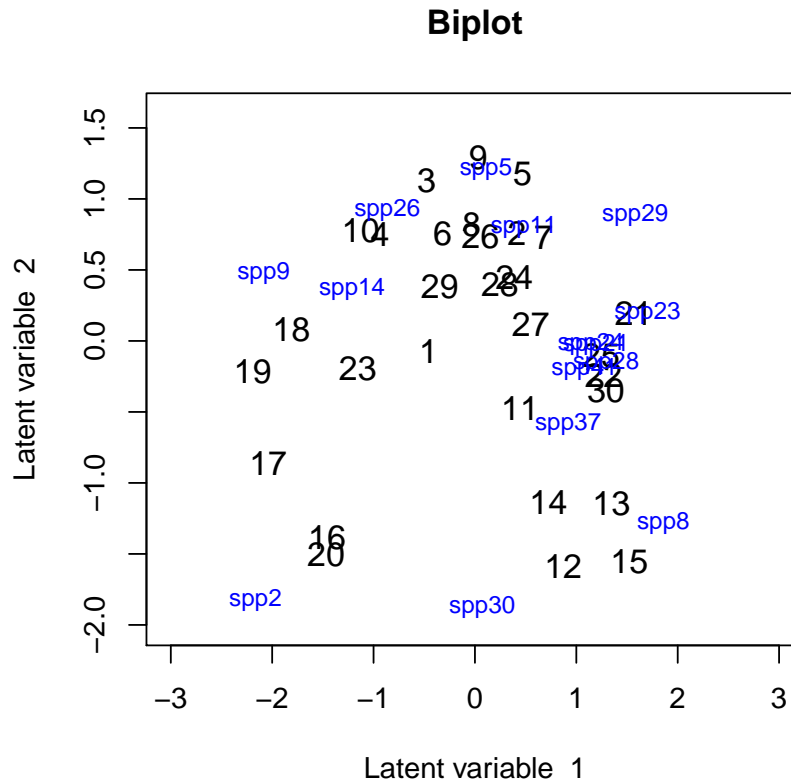
Model with environmental variables

Environmental variables can be included in the model, whether to study their effects on assemblages, or to study patterns of species co-occurrence after controlling for environmental variables. We illustrate here this model by including three environmental variables `Bare.ground`, `Shrub.cover` and `Volume.lying.CWD` to the model. Similarly to previous example, we can use residual analysis and information criterias to study which distribution offers the most suitable mean-variance relationship for the responses, and how many latent variables are needed.

```r
criterias <- NULL
for(i in 0:5){
  fiti <- gllvm(y, X, family = "negative.binomial", num.lv = i, sd.errors = FALSE,
                formula = ~ Bare.ground + Shrub.cover + Volume.lying.CWD, seed = 1234)
  criterias[i + 1] <- summary(fiti)$AICc
  names(criterias)[i + 1] = i
}
```

```r
# Compare AICc values
criterias
#>        0        1        2        3        4        5
#> 3644.572 3570.070 3538.663 3522.019 3522.582 3553.064
```

Based on these tools a model with NB distributed responses and three latent variables was chosen.

```r
fit_env <- gllvm(y, X, family = "negative.binomial", num.lv = 3,
                 formula = ~ Bare.ground + Shrub.cover +
```

9

```
                    Volume.lying.CWD, seed = 1234)
```

The estimated coefficients for predictors and their confidence intervals can be plotted using the `coefplot` function, in order to study the nature of effects of environmental variables on species. Function `coefplot` plots the point estimates (ticks) for coefficients of the environmental variables and their 95% confidence intervals (lines) for the `fit_env` model, with those colored in grey (black) denoting intervals (not) containing zero. The x-axis of the coefficient plot of the third variable is truncated due to very wide confidence interval for one of the coefficients using an argument `xlim.list` below.

```
coefplot(fit_env, cex.ylab = 0.7, mar = c(4, 9, 2, 1),
         xlim.list = list(NULL, NULL, c(-4, 4)), mfrow=c(1,1))
```
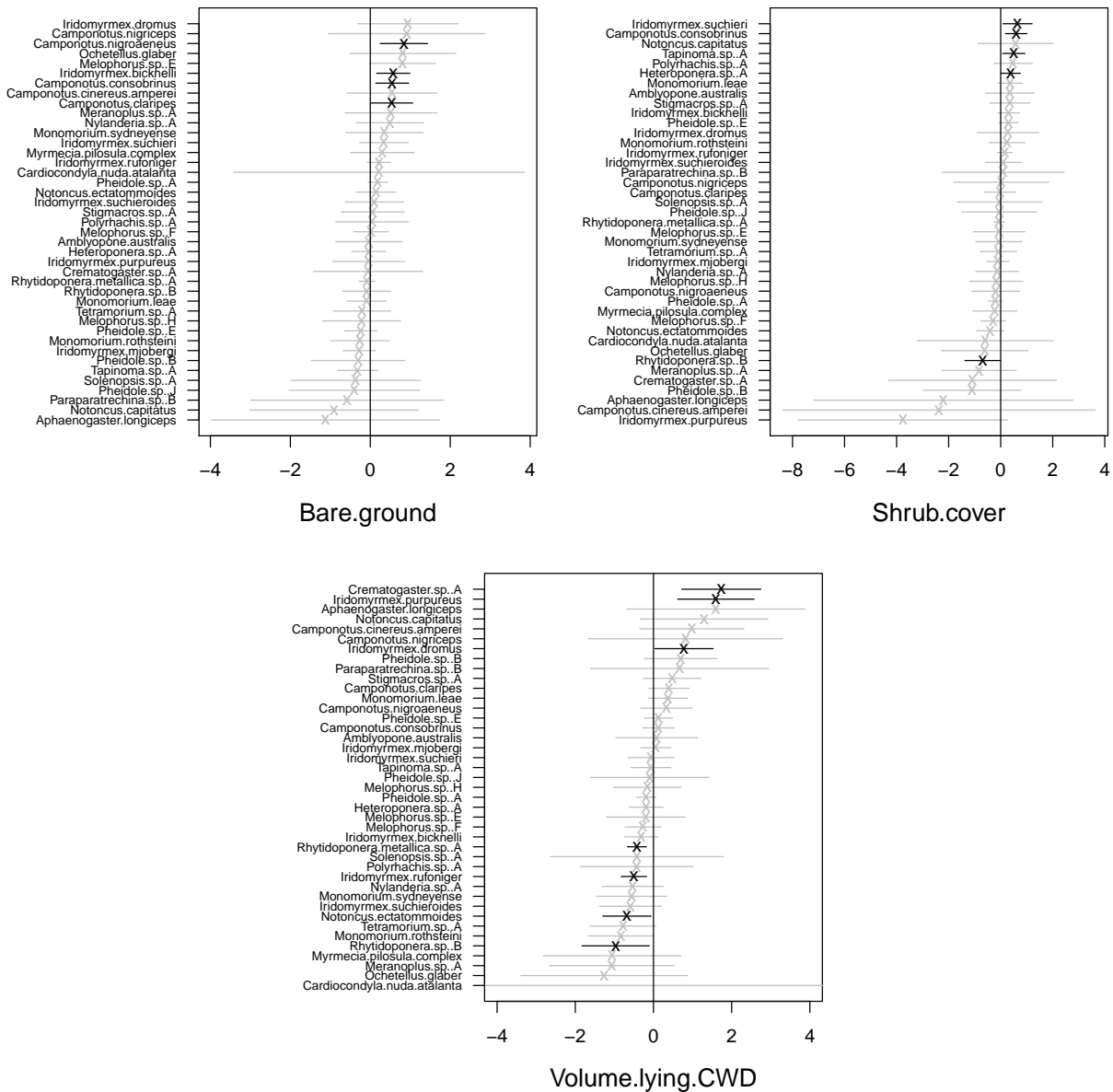

In the resulting plot above most of the 95% confidence intervals include zero, indicating that the majority of the species do not exhibit evidence of a strong association between environment and species abundance. This may be due to a lack of information in the data, as much as being due to a lack of environmental association

10

after accounting for potential residual species covariation.

Similarly to the biplot, the species names in the coefficient plot can also be changed by changing the row names in the coefficient matrix:

```r
rownames(fit_env$params$Xcoef) <- paste("spp", 1:ncol(fit_env$y), sep = "")
coefplot(fit_env, cex.ylab = 0.7, xlim.list = list(NULL, NULL, c(-4, 4)),
         mfrow=c(1,1))
```


Bare.ground

Shrub.cover

Volume.lying.CWD
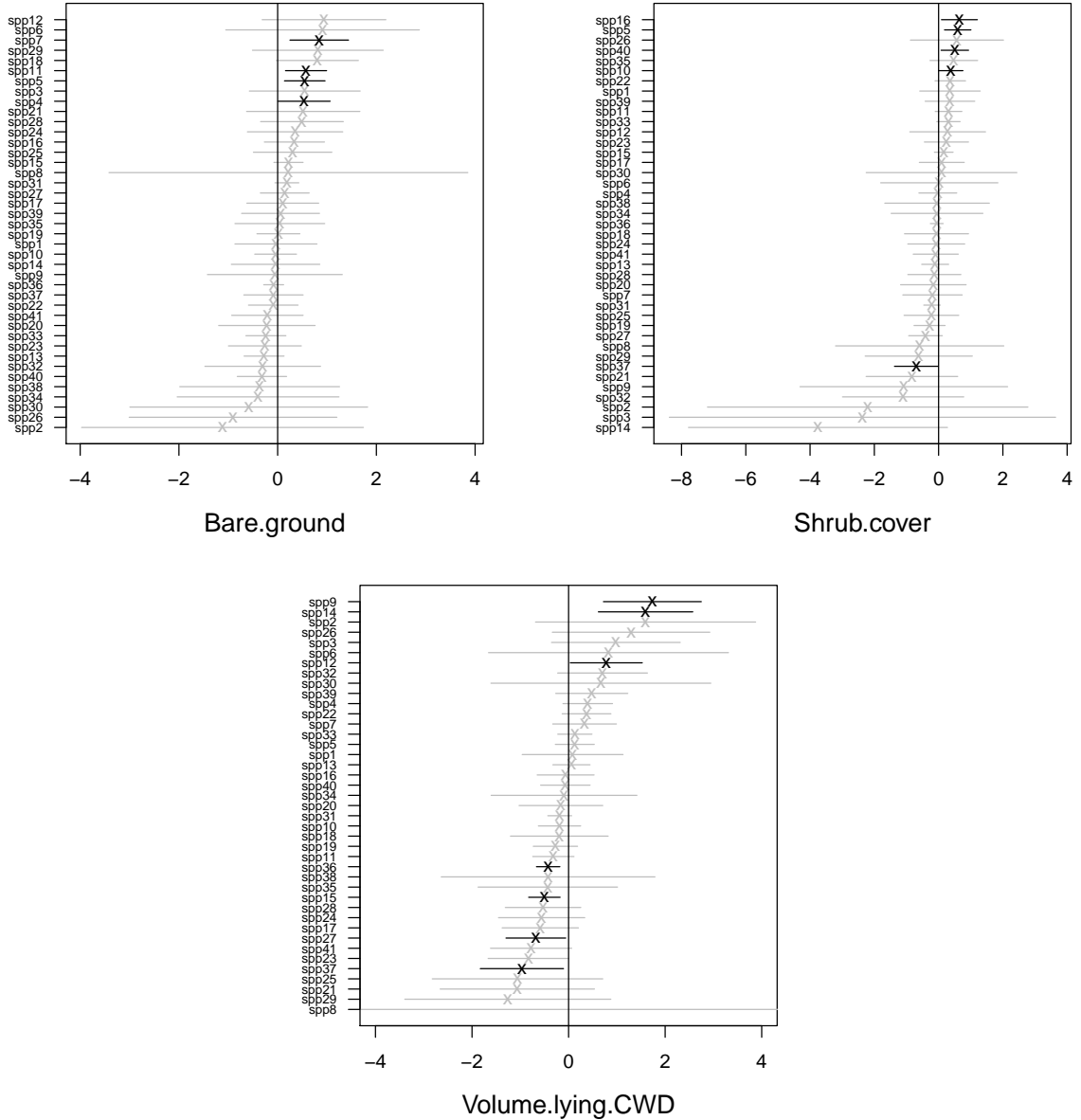
Studying co-occurrence patterns

Latent variables induce correlation across response variables, and so provide a means of estimating correlation patterns across species, and the extent to which they can be explained by environmental variables. Information on correlation is stored in the factor loadings $\boldsymbol{\theta}_j$, so the residual covariance matrix, storing information on species co-occurrence that is not explained by environmental variables, can be calculated as $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\top}$, where

$\mathbf{\Gamma} = [\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_m]'$. The `getResidualCor` function can be used to estimate the correlation matrix of the linear predictor across species. This can be visualised using the `corrplot` package:

```
# Residual correlation matrix:
cr <- getResidualCor(fit_env)
library(corrplot); library(gclus)
#> corrplot 0.84 loaded
#> Loading required package: cluster
```

```
corrplot(cr[order.single(cr), order.single(cr)], diag = FALSE, type = "lower",
         method = "square", tl.cex = 0.5, tl.srt = 45, tl.col = "red")
```
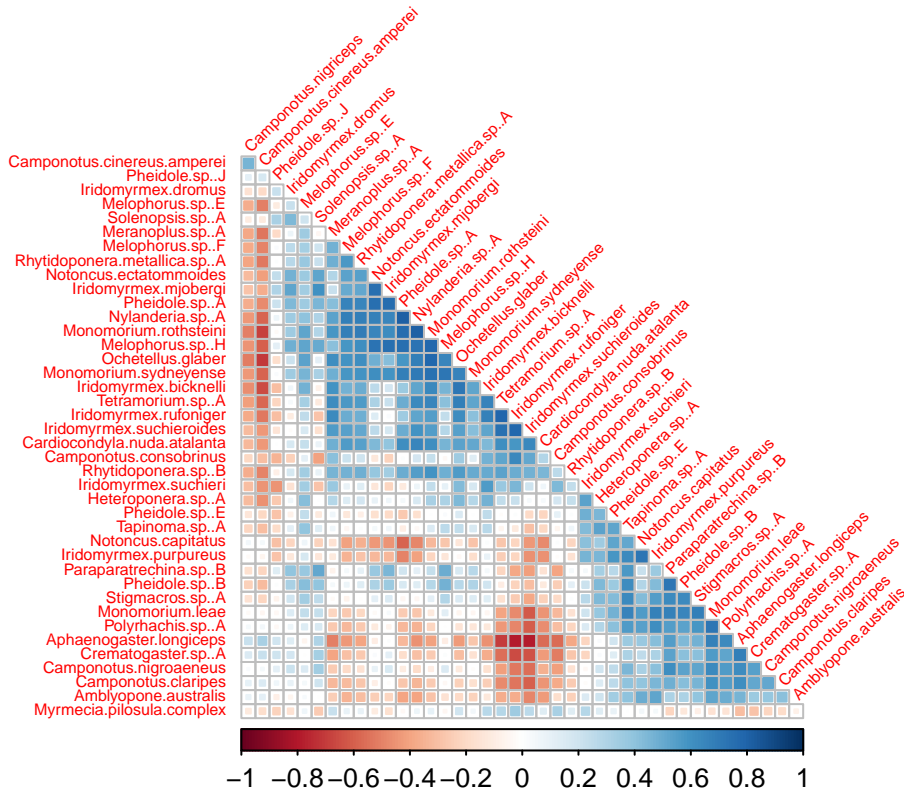

Regions coloured in dark blue in correlation plot indicate clusters of species that are positively correlated with each other, after controlling for (co)variation in species explained by environmental terms in `fit_env` (which had linear terms for `Bare.ground`, `Shrub.cover` and `Volumn.lying.CWD`). There are also two regions coloured in red, indicating negative correlation between pairs of species. The effect of the environmental variables on the between species correlations can be seen by comparing the residual correlation matrix (above) to the correlation matrix given by the model without environmental variables (below).

```
# Fit GLLVM without environmental variables and 3 latent variables:
fit3lv <- gllvm(y, family = "negative.binomial", num.lv = 3, seed = 1234)
```

```
# Correlation matrix
cr0 <- getResidualCor(fit3lv)
corrplot(cr0[order.single(cr0), order.single(cr0)], diag = FALSE, type = "lower",
```

```
                method = "square", tl.cex = 0.5, tl.srt = 45, tl.col = "red")
```
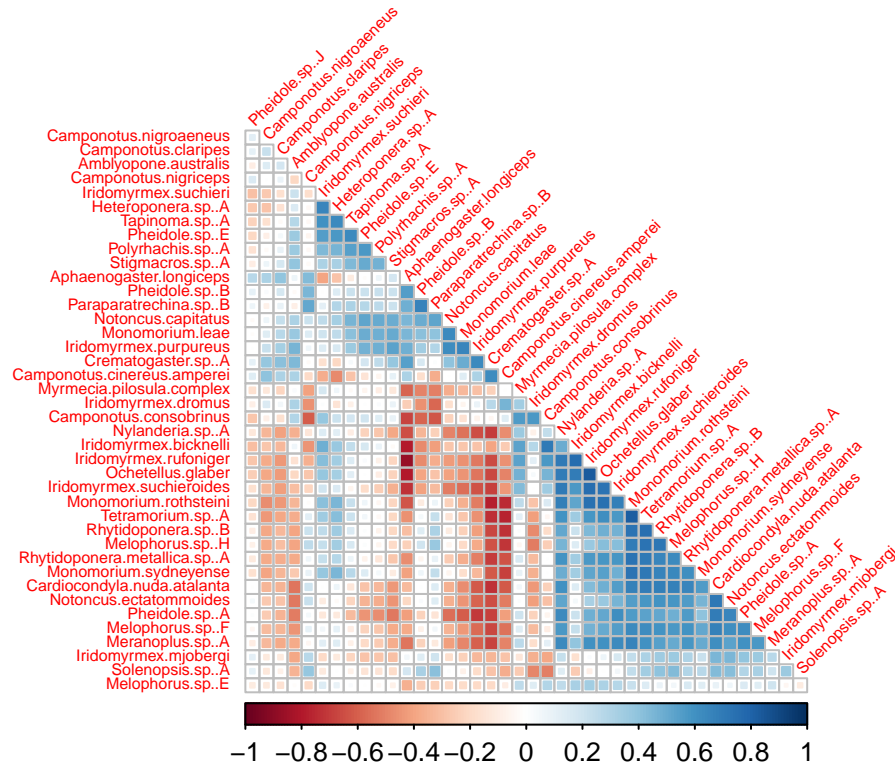

Residual correlations can also be visualized in a residual biplot, where highly correlated species tend to lie in the same direction and negatively correlated species at the opposite side of the biplot, see figure (b) below. In order to do that we fit the `fit_env` model with two latent variables and compare it to the model without predictors `fit_ord` (a).

```
fit_env2 <- gllvm(y, X, family = "negative.binomial", num.lv = 2,
                  formula = ~ Bare.ground + Shrub.cover +
                      Volume.lying.CWD, seed = 1234)
rownames(fit_env2$params$theta) <- paste("sp", 1:ncol(fit_env2$y), sep = "")
ordiplot(fit_ord, biplot = TRUE, ind.spp = 15, jitter = TRUE, cex.spp = 1,
         xlim = c(-4, 3.5), ylim = c(-2.5, 2), main = "(a)")
ordiplot(fit_env2, biplot = TRUE, ind.spp = 15, jitter = TRUE, cex.spp = 1,
         xlim = c(-4, 3.5), ylim = c(-2.5, 2), main = "(b)")
```

(a)   **(b)**
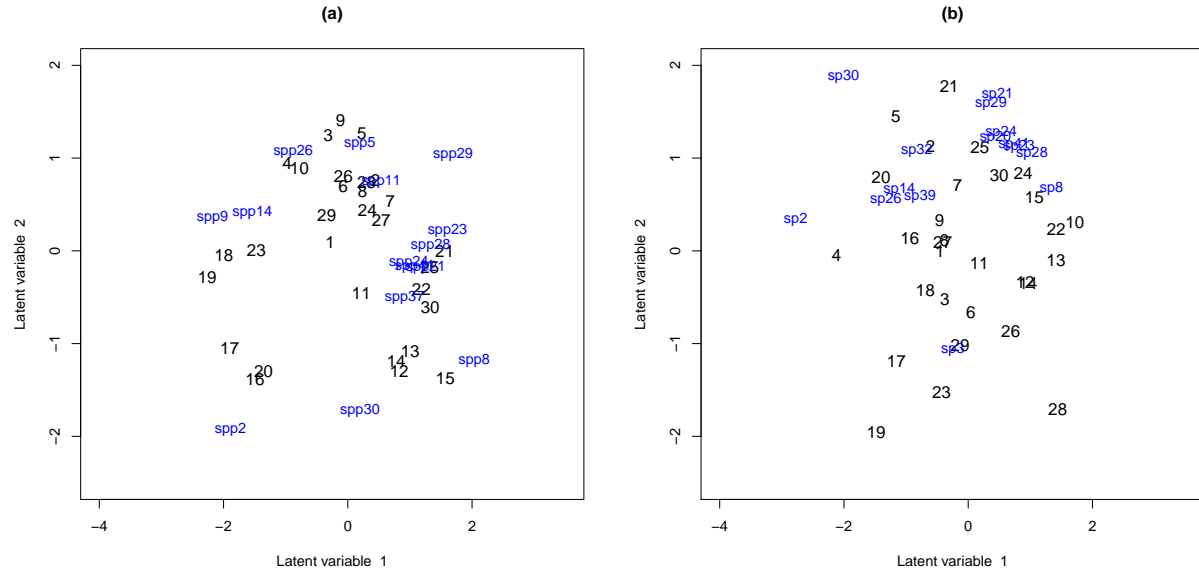
In both biplots we can see that species 23 (Monomorium.rothsteini), 24 (Monomorium.sydneyense), 28 (Nylanderia.sp..A) and 41 (Tetramorium.sp..A) are close to each other. This can also be seen in the correlation plots as their between species correlations are highly positive.

The `getResidualCov` function can be used to quantify the amount of variation in the data that can be explained by environmental variables. Specifically, if we use the trace of the residual covariance matrix Σ as a measure of unexplained variation, then we can compare this quantity before and after environmental variables are included in the model. The ratio of traces suggests that environmental variables explain approximately 25% of the (co)variation in ant species abundances.

```
rcov <- getResidualCov(fit_env, adjust = 0)
rcov0 <- getResidualCov(fit3lv, adjust = 0)
rcov0$trace; rcov$trace
#> [1] 71.66508
#> [1] 52.78108
1 - rcov$trace / rcov0$trace
#> [1] 0.2635035
```

Incorporating functional traits into fourth corner models

In the previous section, environmental associations were studied by fitting separate terms for each species, without attempting to explain why different species respond differently to the environment. Adding functional traits to the model offers the potential to explain why species differ in environmental response. Again,we choose the number of latent variables using AICc criterion:

```
criterias4th <- NULL
for(i in 0:5){
  fiti <- gllvm(y, X, TR, family = "negative.binomial", num.lv = i,
                formula = y ~ (Bare.ground + Shrub.cover + Volume.lying.CWD) +
                  (Bare.ground + Shrub.cover + Volume.lying.CWD) :
                  (Pilosity + Polymorphism + Webers.length),
                seed = 1234)
  criterias4th[i+1] <- summary(fiti)$AICc
  names(criterias4th)[i+1]= i
}
```

```
# Compare AICc values
criterias4th
#>        0        1        2        3        4        5
#> 3817.289 3710.570 3679.402 3662.244 3666.919 3683.995
```

The fourth corner model can be fitted using the argument `TR` to include traits. The argument `formula` is used to specify the model.

```
fit_4th <- gllvm(y, X, TR, family = "negative.binomial", num.lv = 3,
                 formula = y ~ (Bare.ground + Shrub.cover + Volume.lying.CWD) +
                 (Bare.ground + Shrub.cover + Volume.lying.CWD) : (Pilosity +
                 Polymorphism + Webers.length))
```

As previously, coefficients can be plotted using the function `coefplot`. The environmental-trait interaction terms, also known as the fourth corner terms, can be visualized using the function `levelplot` from the package `lattice`, see below. The resulting plots indicate that interactions of the trait variable `Polymorphism` with `Bare.ground` and `Webers.length` with `Volume.lying.CWD` have the strongest effects on ant abundances.

```
library(lattice)
coefplot(fit_4th, mar = c(4, 11, 1, 1), cex.ylab = 0.8)
fourth <- fit_4th$fourth.corner
a <- max( abs(fourth) )
colort <- colorRampPalette(c("blue", "white", "red"))
plot.4th <- levelplot((as.matrix(fourth)), xlab = "Environmental Variables",
                      ylab = "Species traits", col.regions = colort(100), cex.lab = 1.3,
                      at = seq(-a, a, length = 100), scales = list(x = list(rot = 45)))
plot.4th
```
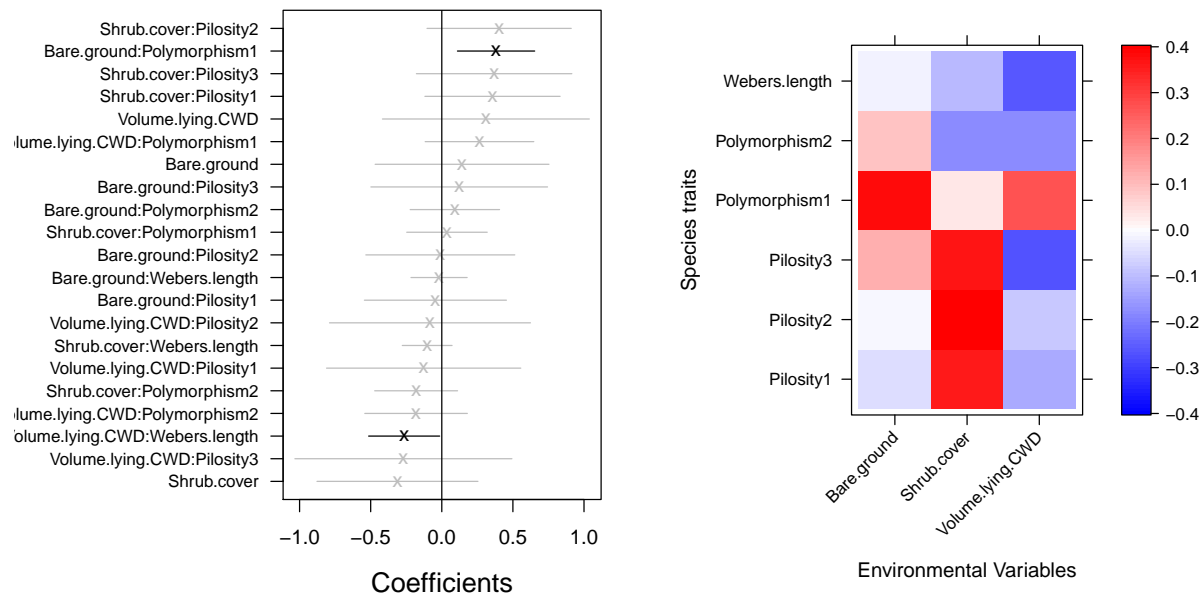

Example: Model-based inference

A key point of departure in the `gllvm` package, as compared to alternative ways to fit GLLVMs, is that by using a maximum likelihood framework, likelihood-based machinery becomes available for inference. A particular example is likelihood ratio testing, available via the `anova` function when comparing nested models.

15

It should be noted however that such tests can only be expected to maintain (close to) valid Type I error rates when the number of parameters involved in the model is small (they rely on large-N-small-m asymptotics).

In the previous example, trait-environment interactions seem to be relatively small and most of the confidence intervals include zero values. Using a likelihood ratio test we can test if the traits vary with environment. Below, the model without traits is fitted and the likelihood ratio test comparing this model to the previous one is performed. Based on the output when applying the `anova` function, the p value suggests that the simpler model without interactions is more appropriate. Notice that in order to separate the model with only main effects for environmental variables from the one which has species specific coefficients for environmental variables, we include `TR` matrix to the function call.

```
fit_4th2 <- gllvm(y, X, TR, family = "negative.binomial", num.lv = 3,
        formula = y ~ (Bare.ground + Shrub.cover + Volume.lying.CWD), seed = 1234)
# Test interactions using likelihood ratio test:
anova(fit_4th, fit_4th2)
#> Model  1 :  y ~ (Bare.ground + Shrub.cover + Volume.lying.CWD)
#> Model  2 :  y ~ (Bare.ground + Shrub.cover +  Volume.lying.CWD) + (Bare.ground +
#> Shrub.cover + Volume.lying.CWD):(Pilosity + Polymorphism + Webers.length)
#>    Resid.Df       D Df.diff  P.value
#> 1     1025  0.00000       0
#> 2     1007 18.90272      18 0.397837
```