Hmsc 3.0: Getting started with Hmsc: high-dimensional multivariate models

Gleb Tikhonov Øystein H. Opedal Nerea Abrego Aleksi Lehikoinen
Melinda M. J. de Jonge Jari Oksanen Otso Ovaskainen

28 May 2020

Introduction

The Hierarchical Modelling of Species Communities (HMSC) framework is a statistical framework for analysis of multivariate data, typically from species communities. We assume that the reader has already gone through the vignettes "Hmsc 3.0: Getting started with Hmsc: univariate models" and "Hmsc 3.0: Getting started with Hmsc: low-dimensional multivariate models". Here we continue with high-dimensional multivariate models, i.e. models for species-rich communities consisting of many species.

The main difference between the low- and high-dimensional cases is that only the latter truly benefits from the hierarchical structure of `Hmsc`. By the hierarchical structure we refer to the species responses to environmental covariates being modelled jointly, and as a function of species traits and phylogenetic relationships. In this context, each species can be considered as one data point. Thus, if there are only a handful of species, there is insufficient data to accurately model the influence of traits and phylogenies. In contrast, if there are many (say, several tens) of species, accounting for traits and phylogenies becomes meaningful.

To get `Hmsc` in use, you need to load it.

```
library(Hmsc)
library(parallel)
library(corrplot)
set.seed(1)
```

We also loaded the `corrplot` package which will be used for plotting, and we set the random number seed with `set.seed(1)` to make the results presented here reproducible.

Generating simulated data for a large community

Simulating phylogeny and species traits

As an example, we will consider a community of 50 species. To bring an evolutionary perspective to our analyses, we start by simulating a phylogeny describing how the species have evolved from their common ancestor. We simulate a phylogeny with the `rcoal` function of the `ape` package.

```
ns = 50
phy = ape::rcoal(n=ns, tip.label = sprintf('species_%.3d',1:ns), br = "coalescent")
plot(phy, show.tip.label = FALSE, no.margin = TRUE)
```

We will next generate simulated trait values for two traits for each of the 50 species. For the sake of illustration we call these traits 'habitat preference for forest' (trait 1, to be abbreviated as H) and 'thermal optimum'

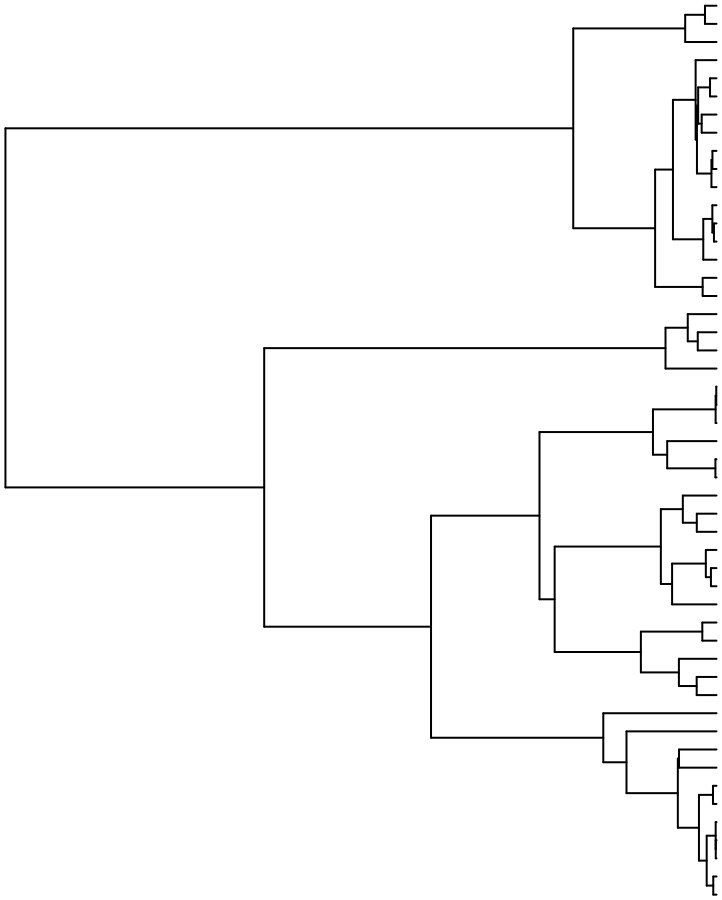

Figure 1: Simulated phylogeny of 50 species

(trait 2, to be abbreviated as T).

We assume that the traits are phylogenetically constrained so that they have evolved according to the diffusion model of trait evolution. These assumptions can be implemented by randomizing the traits from a multivariate normal distribution where the variance-covariance matrix is the phylogenetic correlation matrix `C`. The phylogenetic correlation matrix `C` can be generated from the phylogenetic tree using the function `vcv`of the `ape` package.

```
C = vcv(phy, model = "Brownian", corr = TRUE)
spnames = colnames(C)
traits = matrix(NA,ncol =2,nrow = ns)
for (i in 1:2){
   traits[,i] = matrix(mvrnorm(n = 1, mu = rep(0,ns), Sigma=C))
}
rownames(traits) = spnames
colnames(traits) = c("habitat.use","thermal.optimum")
traits = as.data.frame(traits)
```

We next visualize the trait distributions by plotting them next to the phylogenetic tree.

```
par(fig = c(0,0.6,0,0.8), mar=c(6,0,2,0))
plot(phy, show.tip.label = FALSE)
par(fig = c(0.6,0.9,0.025,0.775), mar=c(6,0,2,0), new=T)
plot.new()
image.plot(t(traits),axes=FALSE,legend.width = 3,legend.shrink=1,
           col = colorRampPalette(c("blue","white","red"))(200))
text(x=1.1, y=0.72, srt = 90,  "H", cex=0.9, pos = 4)
text(x=1.4, y=0.72, srt = 90,  "T", cex=0.9, pos = 4)
```

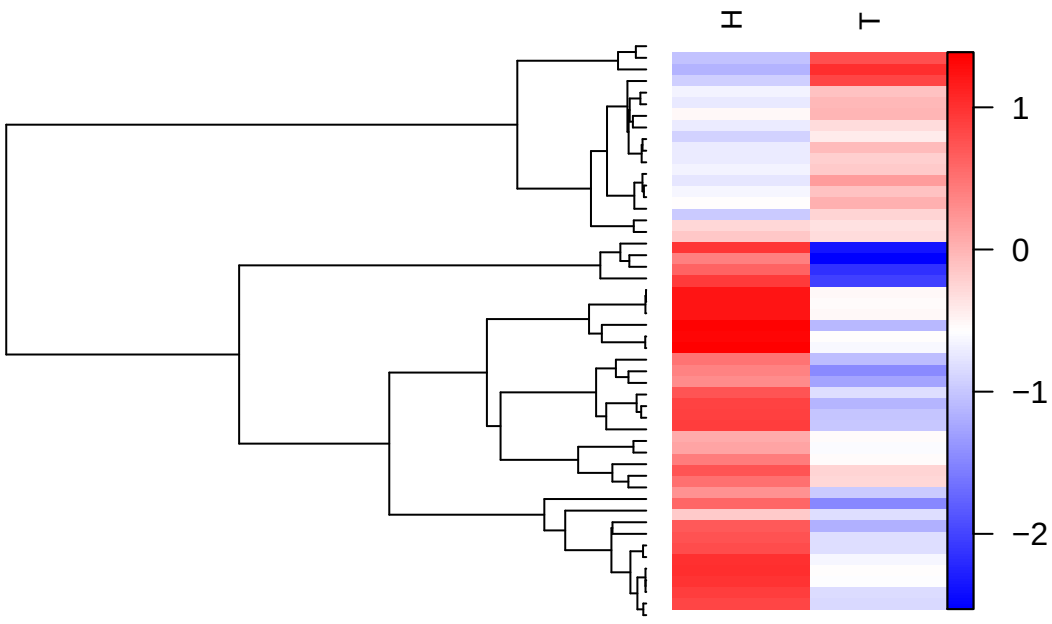

Figure 2: Phylogenetic patterns of species' trait values

As expected, both traits exhibit a clear phylogenetic signal, i.e. related species have similar trait values.

Simulating environmental and species data

Next, we simulate the environmental and species data. With respect to the environmental conditions, we assume that there is variation in habitat type and in climatic conditions. Concerning habitat, we assume that the sampling units are located either in open habitats or in forested habitats. Concerning climate, we assume that the sampling units differ in their thermal conditions, which we characterize by a continuous covariate.

```
n = 200
habitat = factor(sample(x = c("forest","open"), size = n, replace=TRUE))
climate = rnorm(n)
```

The next step is to define the species niches, i.e. how the species abundances depend on the match between their traits and the environmental conditions. Below, we assume that the trait 'habitat use' measures the species' preference for forests, so that for forest habitats we add this trait value to the linear predictor whereas for open habitats we subtract it. We further add the term $-(C_i - T_j)^2/4$ where C_i is the climatic condition at location i and T_j is the thermal optimum of the species j. This makes the linear predictor and hence species abundances the largest when the species thermal optimum matches with the climatic conditions. We can write these choices as the equation $E[L_{ij}] = H_j F_i - H_j(1 - F_i) - (C_i - T_j)^2/4$. Here $E[L_{ij}]$ is the expected value of the linear predictor, H_j is the habitat-use trait of the species j, $F_i = 1$ if location i is located in a forest and $F_i = 0$ if it is located in an open area. To see how this equation is implemented below, we note that $E[L_{ij}] = [-T_j^2/4 - H_j] + [2H_j]F_i + [T_j/2]C_i + [-1/4]C_i^2$. Note that we have also added some random noise to the species niches β so that they are not fully determined by the traits.

```
nc = 4
mu = matrix(0,nrow=nc,ncol=ns)
#expected niche of each species related to the "covariate" intercept
mu[1, ] = -traits$thermal.optimum^2/4-traits$habitat.use
#expected niche of each species related to the covariate forest
#(open area as reference level, so included in intercept)
mu[2, ] = 2*traits$habitat.use
#expected niche of each species related to the covariate climate
mu[3, ] = traits$thermal.optimum/2
#expected niche of each species related to the covariate climate*climate
mu[4, ] = -1/4
beta = mu + 0.25*matrix(rnorm(n = ns*nc), ncol=ns)
X = cbind(rep(1,ns), as.numeric(habitat=="forest"), climate, climate*climate)
L = X%*%beta
```

Finally, to generate the species data, we need to decide about the link function and error distribution. For simplicity, we assume here that the data are normally distributed. We thus simply add some independent residual variation, where we assume the variance parameter to be identical for each species. We note that as the elements of the matrix `Y` can include any real numbers, they might measure species abundance e.g. in the units of log-transformed biomass.

To make the species data interpretable with respect to the phylogenetic and trait information, we associate the data matrix `Y` with the species names.

```
Y = L + mvrnorm(n=n, mu=rep(0,ns), Sigma=diag(ns))
colnames(Y) = spnames
```

HMSC analyses of the data with the "correct" model

We are now ready to analyse the data with HMSC. In this section, we consider the "correct" model for which the assumptions are exactly in line with how we generated the data. Let us note in passing that with real data, the model will always be to some extent misspecified as the underlying mechanisms are not known. In

the next sections, we will examine the influence of model misspecification by examining how the results of the analyses change if we e.g. fail to account for some of the traits or environmental covariates.

To define a HMSC model that best corresponds to how the data were simulated, we include as fixed effects the categorical variable habitat type and the continuous covariate climate. To allow for the possibility of an intermediate thermal optimum, we also include the squared effect of climate. As species traits, we include habitat use (H) and thermal optimum (T). We also include the phylogenetic relationships among the species to estimate the strength of the phylogenetic signal in the data. We thus construct the `Hmsc` model as

```
XData = data.frame(climate = climate, habitat = habitat)
XFormula = ~habitat + poly(climate,degree = 2,raw = TRUE)
TrFormula = ~habitat.use + thermal.optimum
studyDesign = data.frame(sample = sprintf('sample_%.3d',1:n))
rL = HmscRandomLevel(units = studyDesign$sample)
rL$nfMax = 15
m = Hmsc(Y = Y, XData = XData, XFormula = XFormula,
         TrData = traits, TrFormula = TrFormula,
         phyloTree = phy,
         studyDesign = studyDesign, ranLevels = list(sample = rL))
```

In the above model, we included a random effect at the sampling unit level. In the data we simulated, we assumed independent sampling units and no residual associations, and thus the absence of any random effect. Thus, we expect that the random effect will play a minor role also in the fitted model.

We then choose the MCMC sampling parameters and fit the model. If the aim of the reader is to run through the markdown behind the vignette fast to see how the model objects are constructed and how the functions are called, we recommend running minimal amount of MCMC sampling, which can be obtained by setting `test.run = TRUE`. If the aim is to replicate the results that are given in the pdf version of the vignette, the user should increase the amount of MCMC sampling, e.g. by setting `test.run = FALSE`.

```
nChains = 2
test.run = FALSE
if (test.run){
   #with this option, the vignette evaluates in ca. 10 minutes in a laptop
   thin = 1
   samples = 100
   transient = 50
} else {
   #with this option, the vignette evaluates in ca. 2 hrs in a laptop
   thin = 10
   samples = 1000
   transient = 500
}
verbose = 0
m = sampleMcmc(m, thin = thin, samples = samples, transient = transient,
                    nChains = nChains, nParallel = nChains, verbose = verbose)
```

```
## Setting updater$Gamma2=FALSE due to specified phylogeny matrix
```

MCMC convergence

We evaluate MCMC convergence in terms of four kinds of parameters that we are especially interested in: the species niches `Beta`, the influence of traits on species niches `Gamma`, residual species associations `Omega`, and the strength of phylogenetic signal `rho`. As there are 50 species, the matrix `Omega` is a 50 x 50 matrix with 2500 elements. To avoid excessive computational times, we evaluate MCMC convergence for `Omega` only for a subsample of 100 randomly selected species pairs.

```
mpost = convertToCodaObject(m)
par(mfrow=c(3,2))
ess.beta = effectiveSize(mpost$Beta)
psrf.beta = gelman.diag(mpost$Beta, multivariate=FALSE)$psrf
hist(ess.beta)
hist(psrf.beta)
ess.gamma = effectiveSize(mpost$Gamma)
psrf.gamma = gelman.diag(mpost$Gamma, multivariate=FALSE)$psrf
hist(ess.gamma)
hist(psrf.gamma)
sppairs = matrix(sample(x = 1:ns^2, size = 100))
tmp = mpost$Omega[[1]]
for (chain in 1:length(tmp)){
   tmp[[chain]] = tmp[[chain]][,sppairs]
}
ess.omega = effectiveSize(tmp)
psrf.omega = gelman.diag(tmp, multivariate=FALSE)$psrf
hist(ess.omega)
hist(psrf.omega)
```

```
print("ess.rho:")
```

```
## [1] "ess.rho:"
```

```
effectiveSize(mpost$Rho)
```

```
##     var1
## 1482.132
```

```
print("psrf.rho:")
```

```
## [1] "psrf.rho:"
```

```
gelman.diag(mpost$Rho)$psrf
```

```
##      Point est. Upper C.I.
## [1,]  0.9991766  0.9992628
```

The convergence diagnostics suggest reasonably good MCMC convergence, as most potential scale reduction factors are close to one and effective sample sizes relatively high. We note that this is largely thanks to the assumption of a normal model, so that for e.g. probit model for presence-absence data we should most likely run much longer MCMC chains.

Model fit and variance partitioning

With normally distributed data, we measure the explanatory power of the model by R^2.

```
preds = computePredictedValues(m)
MF = evaluateModelFit(hM=m, predY=preds)
hist(MF$R2, xlim = c(0,1), main=paste0("Mean = ", round(mean(MF$R2),2)))
```

We observe that the explanatory power is relatively high for all species, which is not surprising, as we assumed that the species respond strongly to variation in habitat and climatic conditions.

The explained variation can be partitioned into components related to the fixed and the random effects. Before doing so, let us have a look at the design matrix `X` of the model.

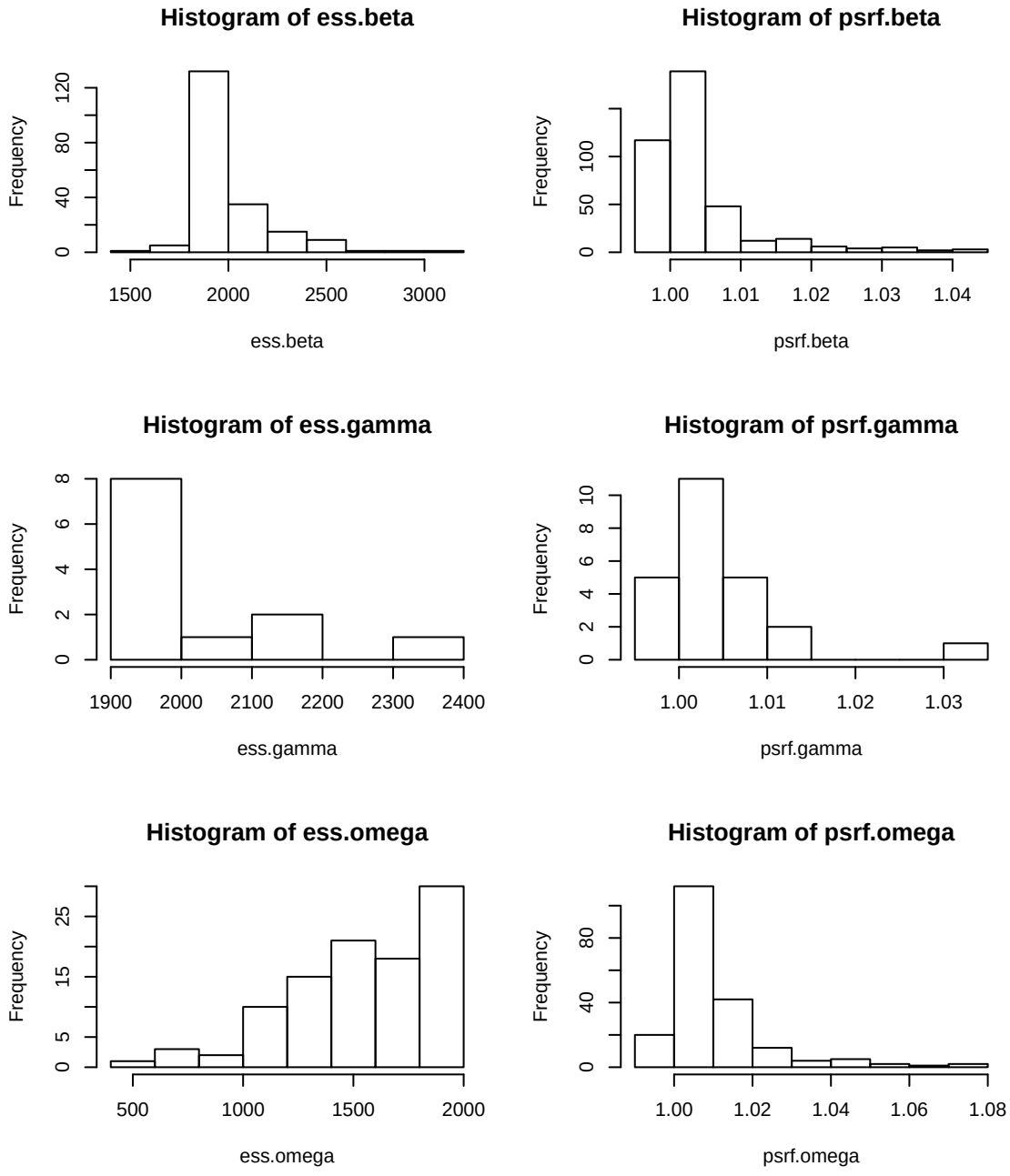

Figure 3: Histograms of effective sample sizes and potential scale reduction factors (psrf) for Beta, Gamma, and Omega parameters

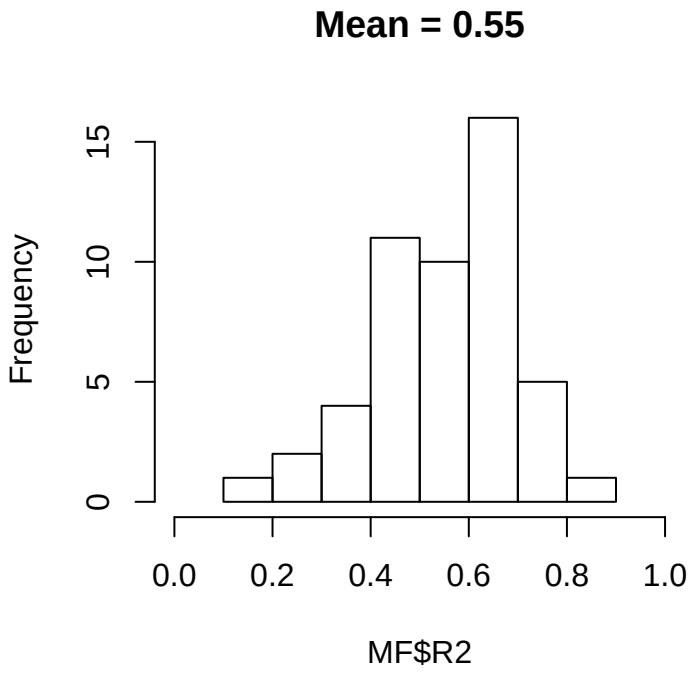

Figure 4: Histogram of species-specific explanatory r2 values

```
head(m$X)
```

```
##   (Intercept) habitatopen poly(climate, degree = 2, raw = TRUE)1
## 1           1           0                              -1.15801525
## 2           1           0                              -0.59274327
## 3           1           0                               0.76606538
## 4           1           1                               0.03892812
## 5           1           0                               0.01464681
## 6           1           0                              -0.18631697
##   poly(climate, degree = 2, raw = TRUE)2
## 1                           1.3409993286
## 2                           0.3513445848
## 3                           0.5868561726
## 4                           0.0015153983
## 5                           0.0002145291
## 6                           0.0347140126
```

The design matrix `X` has been derived from `XData` and `XFormula` when constructing the HMSC object. Following the usual conventions of building generalized linear models, the design matrix includes the intercept by default. The categorical explanatory variables are expanded into dummy variables indicating which level is present in the focal sampling unit. Here the second column corresponds to the explanatory variable habitat, and a value of 1 means that the habitat is open habitat. To avoid confounding with the intercept, one of the levels is left as the reference level, which is why there is no column indicating when the habitat is forest (this is the case when habitat is not in habitat). The continuous covariate climate is present here in two columns modelling its linear and quadratic effects, respectively.

When performing a variance partitioning, the user can group the fixed effects (i.e., the columns of the design matrix) in any way that works best for presenting the results. It is generally recommended to group variables related to the same theme, as then the variance component accounts also for co-variation within the group, whereas co-variation among groups is ignored in the way variance partitioning is computed in HMSC. Here

we wish to show separately the influences of habitat and climate. We call habitat group 1, and climate group 2. The column 2 of the design matrix belongs to group 1, so we define `group[2]=1`. The columns 3 and 4 of the design matrix belongs to group 2, so we define `group[3]=2` and `group[4]=2`. The intercept does not have any variation, and thus it will not contribute to the variance partitioning. Thus we may place it e.g. to group 1 as `group[1]=1`.

```
VP = computeVariancePartitioning(m, group = c(1,1,2,2), groupnames = c("habitat","climate"))
plotVariancePartitioning(m, VP = VP)
```

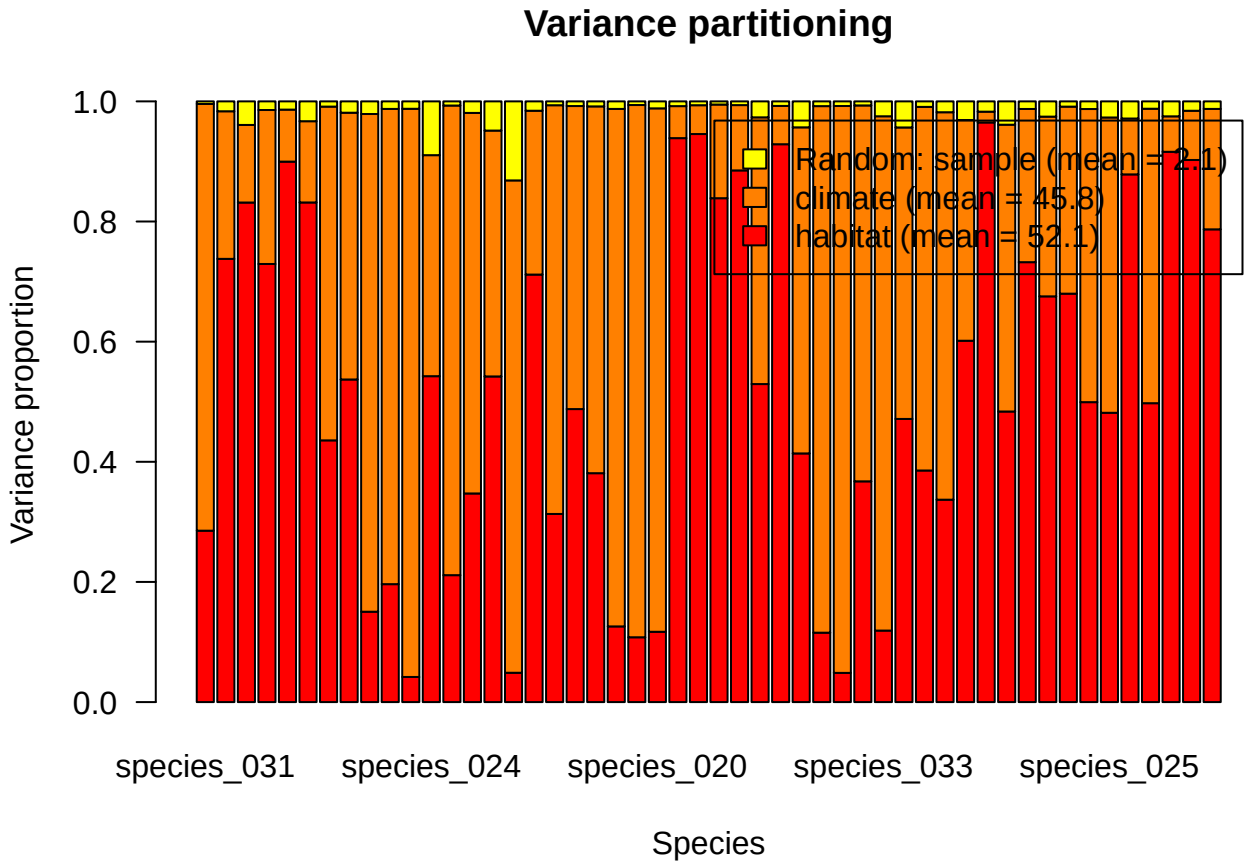

Figure 5: Variance partitioning for each of 50 species

We observe that both components of environmental variation (climate and habitat) explain substantial proportions of the explained variance. Which one of these is more important varies among the species. As expected, the random effect explains next to nothing.

Let us then ask how much the traits explain out of the variation among the species in their responses to environmental covariates.

```
kable(VP$R2T$Beta)
```

	x
(Intercept)	0.7895760
habitatopen	0.9780455
poly(climate, degree = 2, raw = TRUE)1	0.5879268
poly(climate, degree = 2, raw = TRUE)2	0.0110393

The traits explain a substantial proportion of the variation in the species niches. This is because we have included in the HMSC model those traits that truly matter in how species respond to the environmental variation. We next ask how the influence of traits on species niches propagates into the influence of traits on species abundances.

```
VP$R2T$Y
```

```
## [1] 0.5898486
```

The traits explain a substantial proportion of variation also in species abundances.

Parameter estimates

As we have 50 species and four species-specific parameters (one for each column of the design matrix), there are 200 β parameters in total. Thus it is more convenient to look at these as a figure rather than a vast table of numbers. In the function `plotBeta`, we use the option `plotTree = TRUE` to plot the phylogenetic tree next to the parameter estimates, thus 'mapping' the environmental responses onto the phylogeny.

```
postBeta = getPostEstimate(m, parName = "Beta")
plotBeta(m, post = postBeta, param = "Support",
         plotTree = TRUE, supportLevel = 0.95, split=.4, spNamesNumbers = c(F,F))
```

```
## [1] 0.4 1.0 0.0 1.0
```

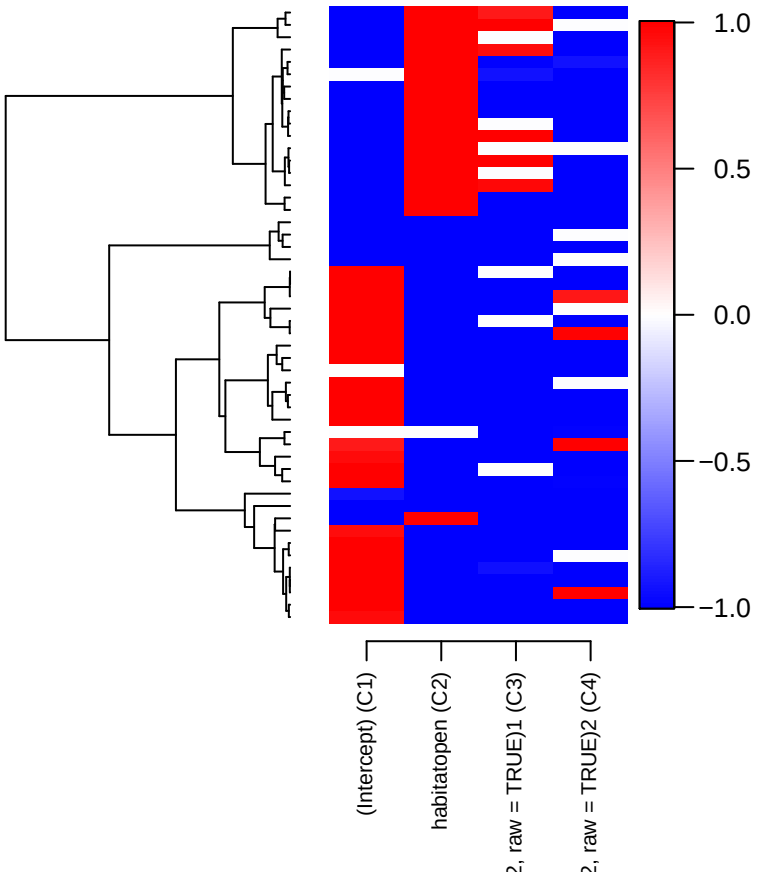

Figure 6: Heatmap of species' environmental responses mapped onto the phylogeny

This figure contains a lot of information. First of all, there is strong posterior support for many parameters

being positive (red) or negative (blue), i.e. there are strong signals of environmental filtering. Some species exhibit preference for open habitats (red colour in the second column C2), other exhibit avoidance of open habitats and thus preference for forests (blue colour in the column C2). Species with preference for open and forest habitats are not randomly distributed across the phylogeny, but sister species exhibit similar responses, corresponding to our assumption that the species trait controlling for habitat preference is phylogenetically constrained. Essentially all species exhibit negative responses to the second-order term of climate (column C4), meaning that their estimated climatic niches have an intermediate optimum. A phylogenetic signal is clear also in column C3 which shows the linear effect of the climatic niche and thus determines at which temperature the species niche peaks.

While the `Beta` parameters model species niches, the `Gamma` parameters model how species traits influence their niches. Let us next visualize these parameters.

```
postGamma = getPostEstimate(m, parName = "Gamma")
plotGamma(m, post=postGamma, param="Support", supportLevel = 0.95)
```

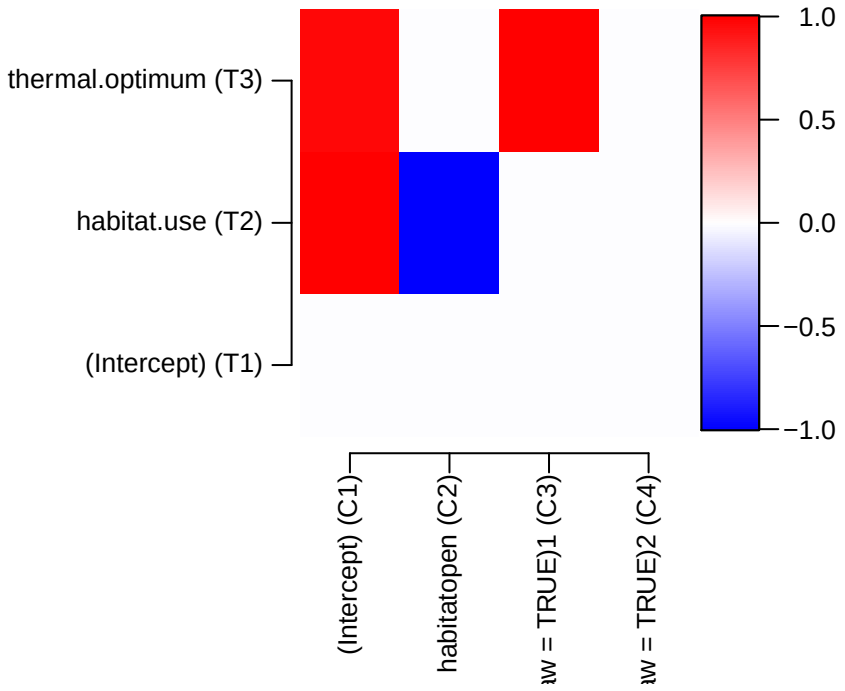

Figure 7: Heatmap showing how species traits influence environmental responses

In this figure, the rows correspond to the species traits (more precisely, the columns of the design matrix `Tr` that is derived from `TrData` and `TrFormula`), and the rows to the environmental covariates (more precisely, the columns of the design matrix `X` illustrated above). This plot yields a compact summary of the results at the community level. First, we observe a negative association between T2 (the trait `habitat.use`) and C2 (the level `open` of the environmental covariate `habitat`), meaning that those species which have a high value of the trait 'habitat use' tend to be less common in open habitats than in the reference habitat (forest). This makes perfect sense, as we assumed that habitat use influences the preference for forests. Second, we observe a positive association between T4 (the trait `thermal.optimum`) and the covariate C3 (the linear effect of the environmental covariate `climate`). This also makes perfect sense, as it means that species with a higher thermal optimum are more abundant in sampling units with warmer climates. Third, we observe a negative association between T1 (the trait `intercept`) and the covariate C4 (`second order response to climate`). This corresponds to the result already observed in the `Beta` plot, that all species generally exhibit a negative response to the second-order term of climate, and thus their response curve peaks at some intermediate temperature.

Let us next visualize the estimated residual associations among the species.

```
OmegaCor = computeAssociations(m)
supportLevel = 0.95
toPlot = ((OmegaCor[[1]]$support>supportLevel)
          + (OmegaCor[[1]]$support<(1-supportLevel))>0)*OmegaCor[[1]]$mean
corrplot(toPlot, method = "color",
         col=colorRampPalette(c("blue","white","red"))(200),
         tl.cex=.6, tl.col="black",
         title=paste("random effect level:", m$rLNames[1]), mar=c(0,0,1,0))
```

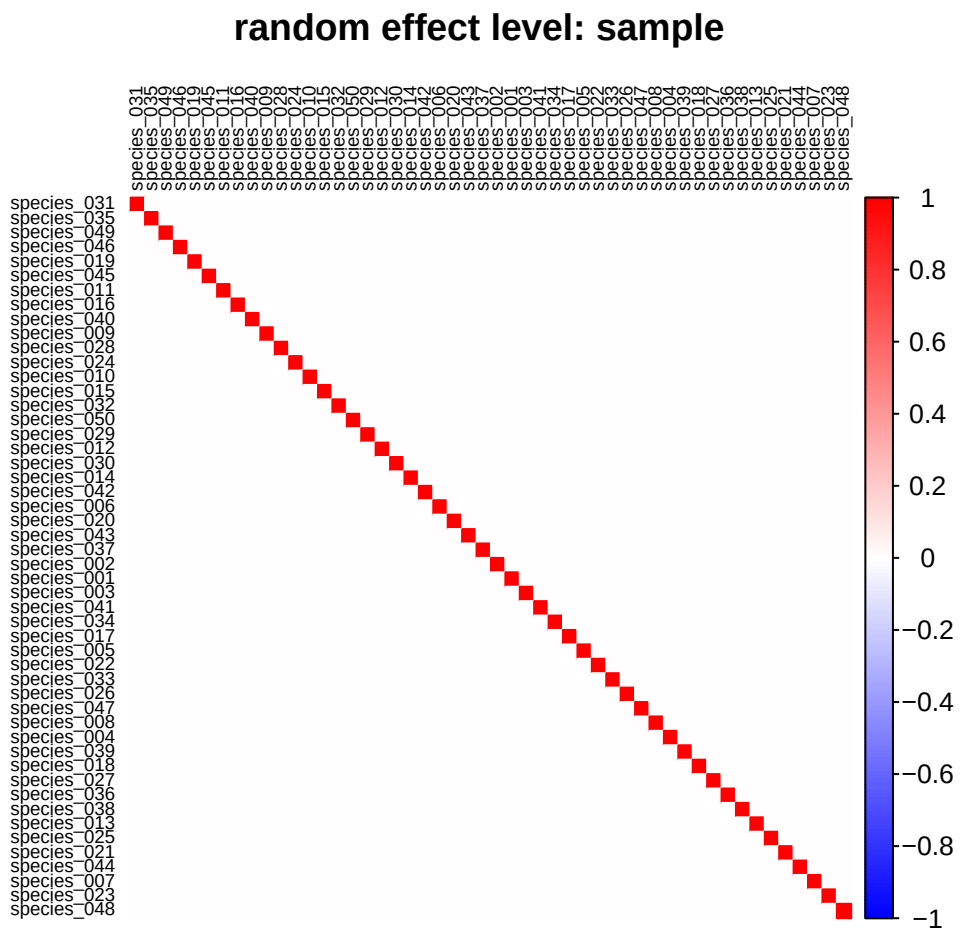

Figure 8: Heatmap showing species residual associations

The fact that there are no associations with high statistical support is in line with how we generated the data.

We finally look at the strength of the phylogenetic signal that was estimated from the data.

```
summary(mpost$Rho)
```

```
##
## Iterations = 510:10500
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
```

```
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##       0.651475       0.100552       0.002248       0.002612
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75% 97.5%
##  0.41  0.59  0.66  0.73  0.81
```

We note that the phylogenetic signal parameter ρ of `Hmsc` does not ask whether the species traits are correlated with the phylogeny, a question that is often in the focus of phylogenetic comparative analyses. Instead, the phylogenetic signal parameter ρ measures whether the species niches (i.e., their responses to the environmental covariates, as measured by the β parameters) show phylogenetic correlations. Furthermore, this is evaluated after accounting for the influence of species traits on species niches (as modelled by the γ parameters). We do not see strong evidence for phylogenetic signal, as the 95%CI of ρ includes zero. This is consistent with how we generated the data, because we propagated the influence of phylogeny solely through those species traits that we included in the HMSC model.

Plotting variation over environmental gradients

The values of the primary parameters `Beta` and `Gamma` can sometimes be difficult to interpret, especially if there are categorical traits or environmental covariates in which case their effects are coded through dummy variables. One alternative for visualizing the results is to plot how the community changes over some environmental gradient of interest. This can be done in HMSC by constructing environmental gradients with the function `constructGradient`, then predicting communities over those gradients by the function `predict`, and finally using `plotGradient`to visualize the predicted variation. We consider first a climatic gradient.

```
Gradient = constructGradient(m,focalVariable = "climate",
                             non.focalVariables = list("habitat"=list(3,"open")))
Gradient$XDataNew
```

```
##        climate habitat
## 1  -2.87514168    open
## 2  -2.55190188    open
## 3  -2.22866208    open
## 4  -1.90542228    open
## 5  -1.58218248    open
## 6  -1.25894268    open
## 7  -0.93570288    open
## 8  -0.61246308    open
## 9  -0.28922328    open
## 10  0.03401652    open
## 11  0.35725632    open
## 12  0.68049612    open
## 13  1.00373592    open
## 14  1.32697572    open
## 15  1.65021552    open
## 16  1.97345532    open
## 17  2.29669512    open
## 18  2.61993492    open
## 19  2.94317472    open
## 20  3.26641452    open
```

As `climate` is a continuous covariate, the constructed gradient involves a grid of its values ranging from the

smallest to the largest value observed in the data. As we will make predictions of species communities, we need to assume values for all explanatory variables, not only to the focal one. We have set here the value of the non-focal variable habitat to open habitat, and we thus imagine that we sample different parts of the climatic gradient solely in open habitats. We then generate predictions for this community, and plot them over the environmental gradient.

```
predY = predict(m, XData=Gradient$XDataNew, studyDesign=Gradient$studyDesignNew,
                ranLevels=Gradient$rLNew, expected=TRUE)
plotGradient(m, Gradient, pred=predY, measure="S", showData = TRUE)
```

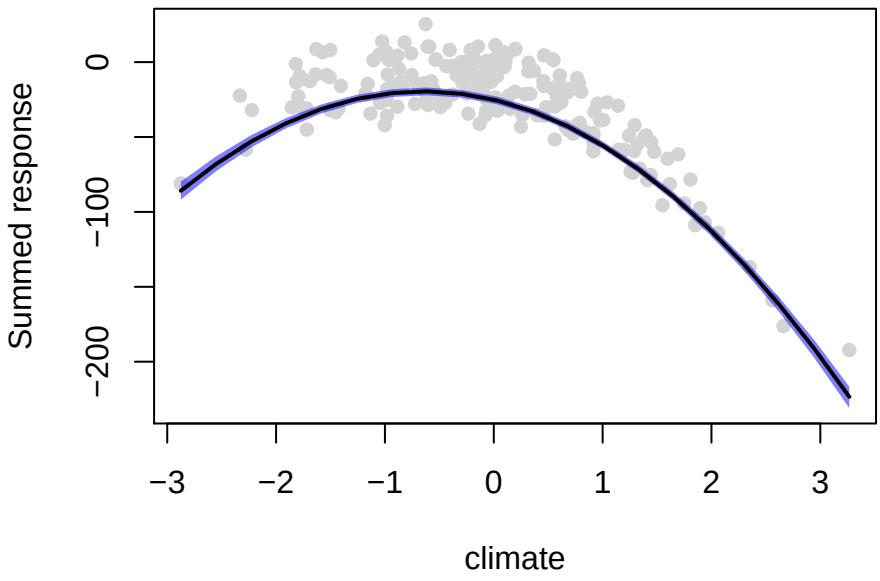

Figure 9: Effect of climate on total abundance

```
## [1] 0
```

We set `measure="S"` to plot the summed abundance over all species, i.e. the row sum of the predicted communities. The predicted response peaks at intermediate climate, a response that is also visible in the raw data. This is because, by our assumptions, some species are specialized to cold climates, some to warm climates, and others to intermediate climates, leading to the pattern where intermediate climate is on average the most suitable one. We note that with presence-absence data modelled with the probit model, the measure `S` would give the expected species richness.

We can visualize the same predictions for individual species by setting `measure="Y"` and by using `index` to select the species to be visualized (as ordered in the matrix `m$Y`).

```
plotGradient(m, Gradient, pred=predY, measure="Y", index = 1, showData = TRUE)
```

```
## [1] 0
```

This figure shows that `species_089` is most abundant under relatively cold climates. Finally, by selecting `measure="T"` we can visualize how community-weighted mean trait values behave over the environmental gradient. Now `index` selects the trait (as ordered in the matrix `m$Tr`).

```
plotGradient(m, Gradient, pred=predY, measure="T", index = 3, showData = TRUE)
```

```
## [1] 1
```

This figure shows that, as expected, the community-weighted mean of thermal optimum is higher under warmer climates. For normally-distributed data, HMSC computes community-weighted mean traits by using exponentially transformed predictions as weights, to avoid weighting with negative numbers.

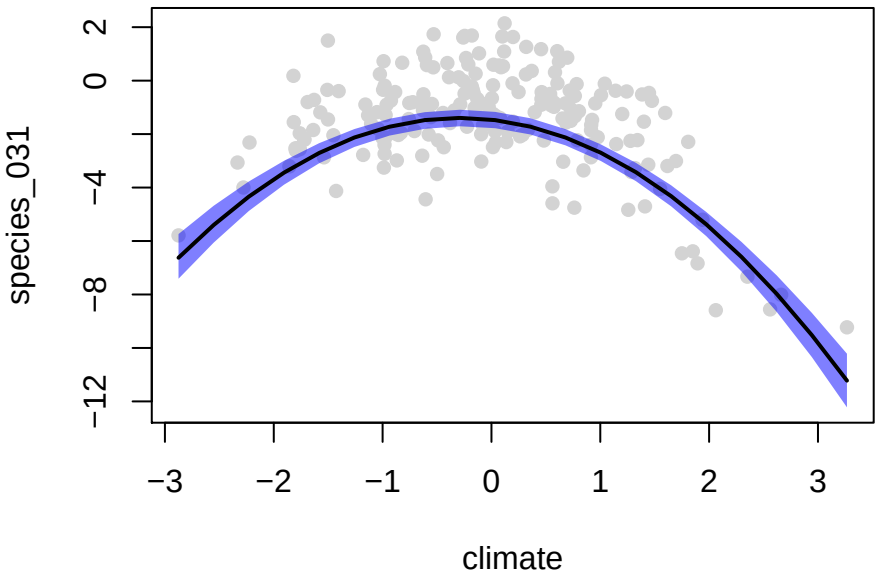

Figure 10: Effect of climate on the abundance of species 1

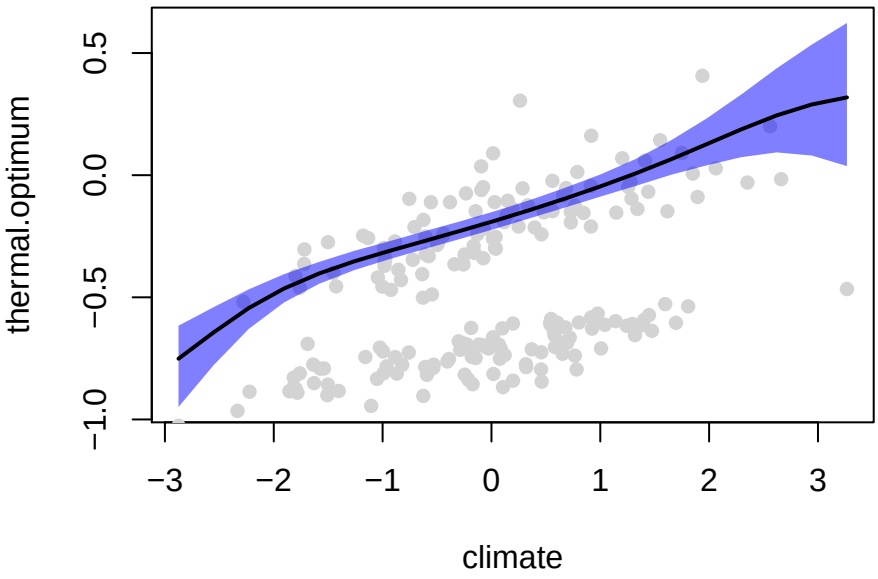

Figure 11: Effect of climate on community-weighted mean thermal optimum

Let us then construct an environmental gradient over the habitat types.

```
Gradient = constructGradient(m,focalVariable = "habitat",
                             non.focalVariables = list("climate"=list(1)))
Gradient$XDataNew
```

```
##   habitat    climate
## 1  forest -0.0208926
## 2    open -0.0208926
```

As habitat is a categorical variable, the gradient involves only two sampling units, one belonging to forest and the other to open habitat. We have decided to normalize the climatic variable to its overall mean in the data (for other options, see the F1-help for `constructGradient`).

Let us select the species for which the trait value for habitat use is the highest, and then plot how that species responds to the habitat gradient.

```
predY = predict(m, XData=Gradient$XDataNew, studyDesign=Gradient$studyDesignNew,
                ranLevels=Gradient$rLNew, expected=TRUE)
plotGradient(m, Gradient, pred=predY, measure="Y", index=which.max(m$TrData$habitat.use),
             showData = TRUE, jigger = 0.2)
```

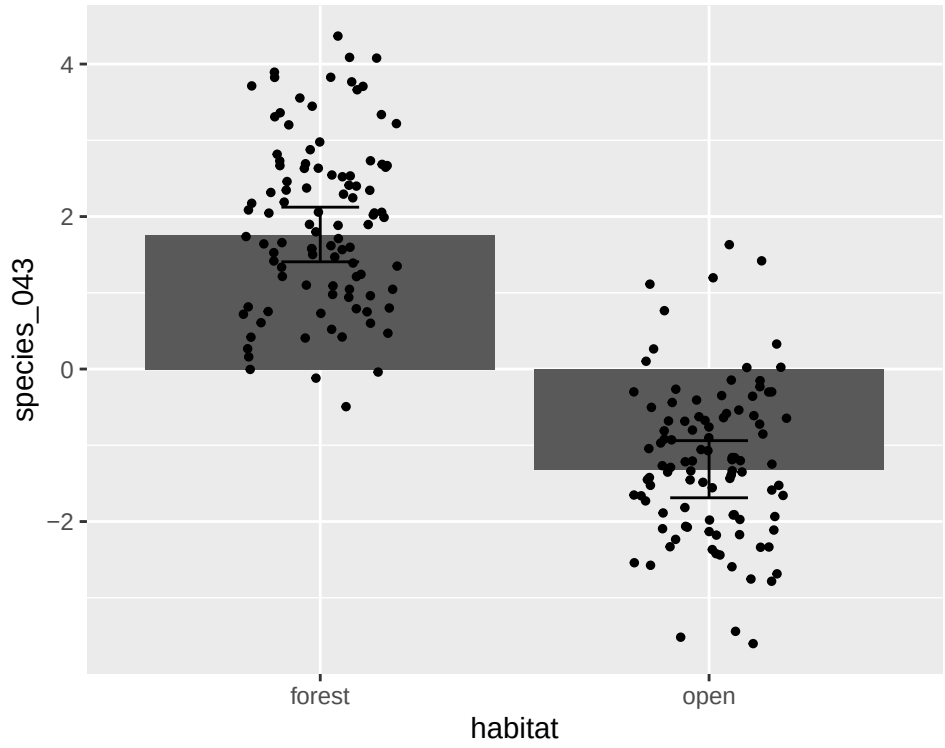

Figure 12: Effect of habitat on the abundance of the most habitat-responsive species

As expected, this species exhibits a stronger predicted response to forests than to open habitats, a pattern visible also in the raw data. We note that because habitat is a categorical covariate, the environmental gradient is now more naturally presented as a boxplot. We further note that we have set `jigger = 0.2` to randomly move the observed data (the dots) in the horizontal direction to avoid overlapping points.

We finally set `measure="T"` to examine how the community-weighted mean of the trait habitat use varies between the two habitat types.

```
plotGradient(m, Gradient, pred=predY, measure="T", index=2, showData = TRUE, jigger = 0.2)
```

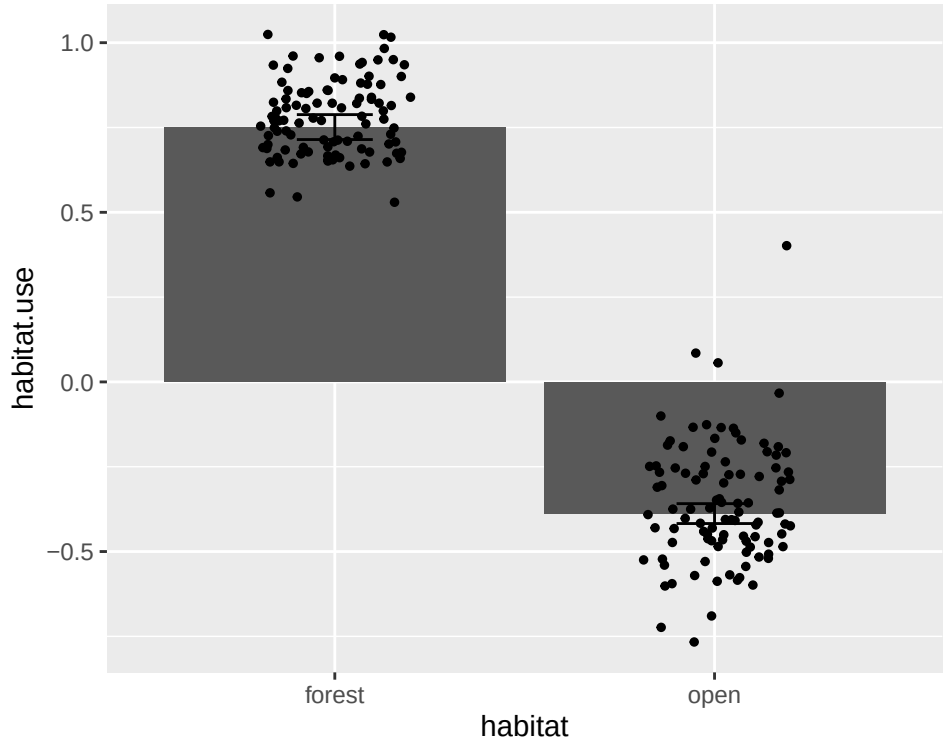

Figure 13: Effect of habitat on community-weighted mean habitat use

As expected, the average value of this trait (weighted by exponentially transformed species abundances) is higher in forests than in open habitats.

HMSC analyses of misspecified models

With real data, there is no "correct" model structure that would exactly correspond to the community-assembly processes that generated the data. Thus, like any model, also the HMSC model will always be "wrong". We next briefly examine what happens if we misspecify the model by leaving out some environmental covariates or traits.

Missing environmental covariate

We first repeat some of the above analyses with a missing environmental covariate. We assume that the researcher would not have realized that the data are sampled from two different habitats, or that the researcher did not consider it to matter for the species abundances. Thus, we fit otherwise the same model, but include only climate as a fixed effect.

```
XFormula.1 = ~poly(climate, degree = 2, raw = TRUE)
ma50 = Hmsc(Y=Y, XData=XData, XFormula = XFormula.1,
         TrData = traits, TrFormula = TrFormula,
         phyloTree = phy,
         studyDesign=studyDesign, ranLevels=list(sample=rL))
```

```
ma50 = sampleMcmc(ma50, thin = thin, samples = samples, transient = transient,
                  nChains = nChains, nParallel = nChains, verbose = verbose)
```

```
## Setting updater$Gamma2=FALSE due to specified phylogeny matrix
```

```
VP = computeVariancePartitioning(ma50, group = c(1,1,1), groupnames=c("climate"))
plotVariancePartitioning(ma50, VP = VP)
```

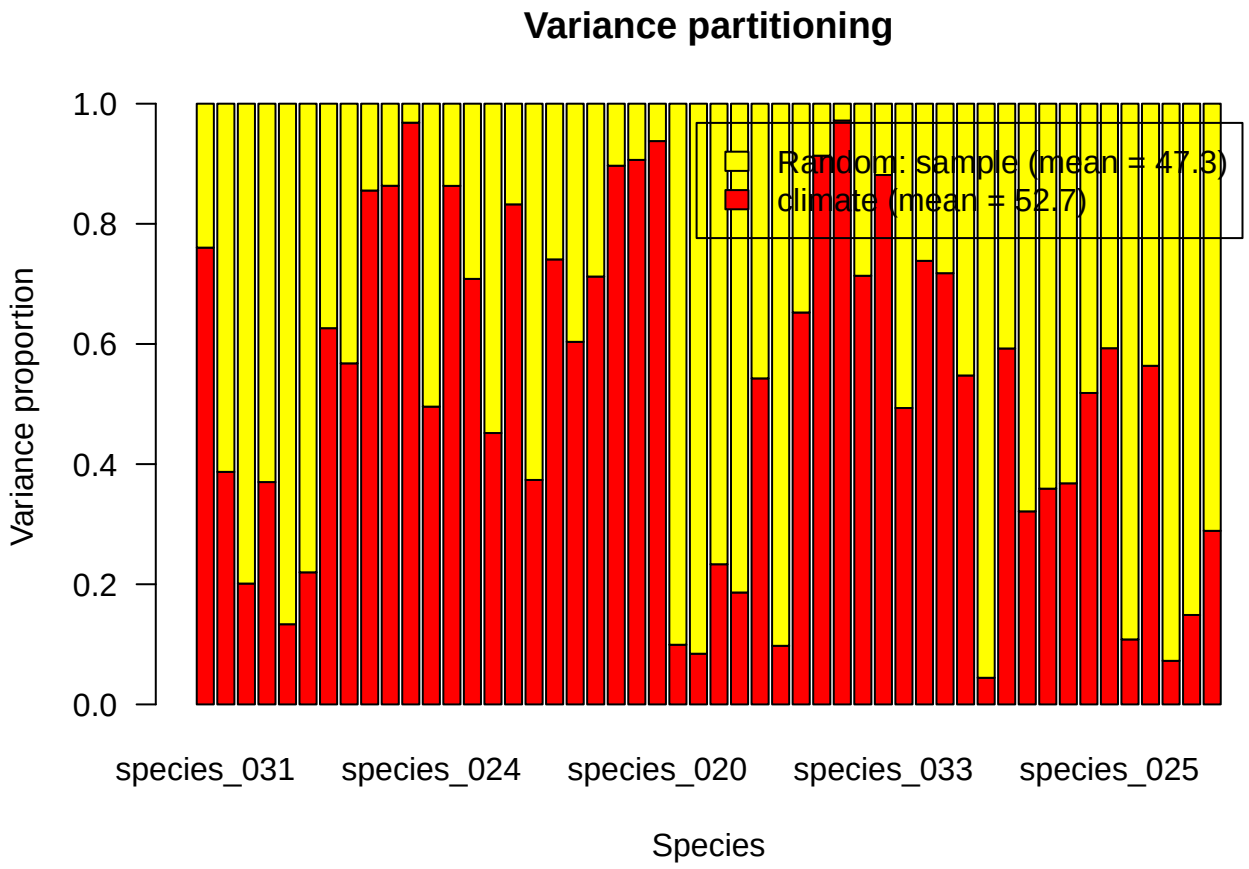

Figure 14: Variance partitioning for each of 50 species

Expectedly, the random effects now play a bigger role. Let us plot the structure of the species associations, as estimated by the random effect.

```
OmegaCor = computeAssociations(ma50)
supportLevel = 0.95
toPlot = ((OmegaCor[[1]]$support>supportLevel)
          + (OmegaCor[[1]]$support<(1-supportLevel))>0)*OmegaCor[[1]]$mean
corrplot(toPlot, method = "color",
         col=colorRampPalette(c("blue","white","red"))(200),
         tl.cex=.6, tl.col="black",
         title=paste("random effect level:", m$rLNames[1]), mar=c(0,0,1,0))
```

We now observe a much richer structure of associations. This is because the responses of the species to habitat are not modelled through the fixed effects, and thus the random effect is capturing part of that variation. Species pairs that prefer the same habitat type are now estimated to have a positive association, whereas species pairs that prefer different habitat types are now estimated to have a negative association.

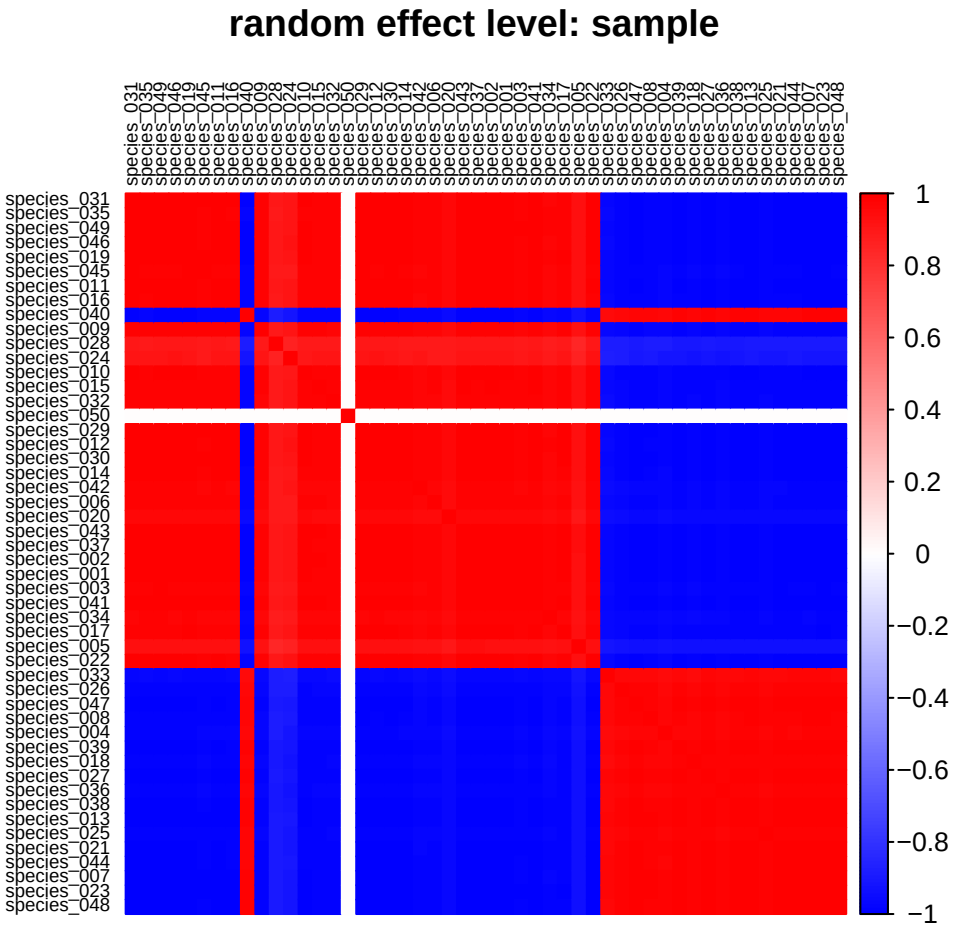

Figure 15: Heatmap showing species residual associations

Missing traits

Let us next assume that we would have trait data only on habitat use, but we would lack information on thermal optima. We repeat the above analyses with a model modified to correspond to this assumption.

```
TrFormula.1 = ~habitat.use
m = Hmsc(Y=Y, XData=XData, XFormula = XFormula,
         TrData = traits, TrFormula = TrFormula.1,
         phyloTree = phy,
         studyDesign=studyDesign, ranLevels=list(sample=rL))

m = sampleMcmc(m, thin = thin, samples = samples, transient = transient,
                    nChains = nChains, nParallel = nChains, verbose = verbose)
```

```
## Setting updater$Gamma2=FALSE due to specified phylogeny matrix
```

```
VP = computeVariancePartitioning(m, group = c(1,1,2,2), groupnames=c("habitat","climate"))
plotVariancePartitioning(m, VP = VP)
```

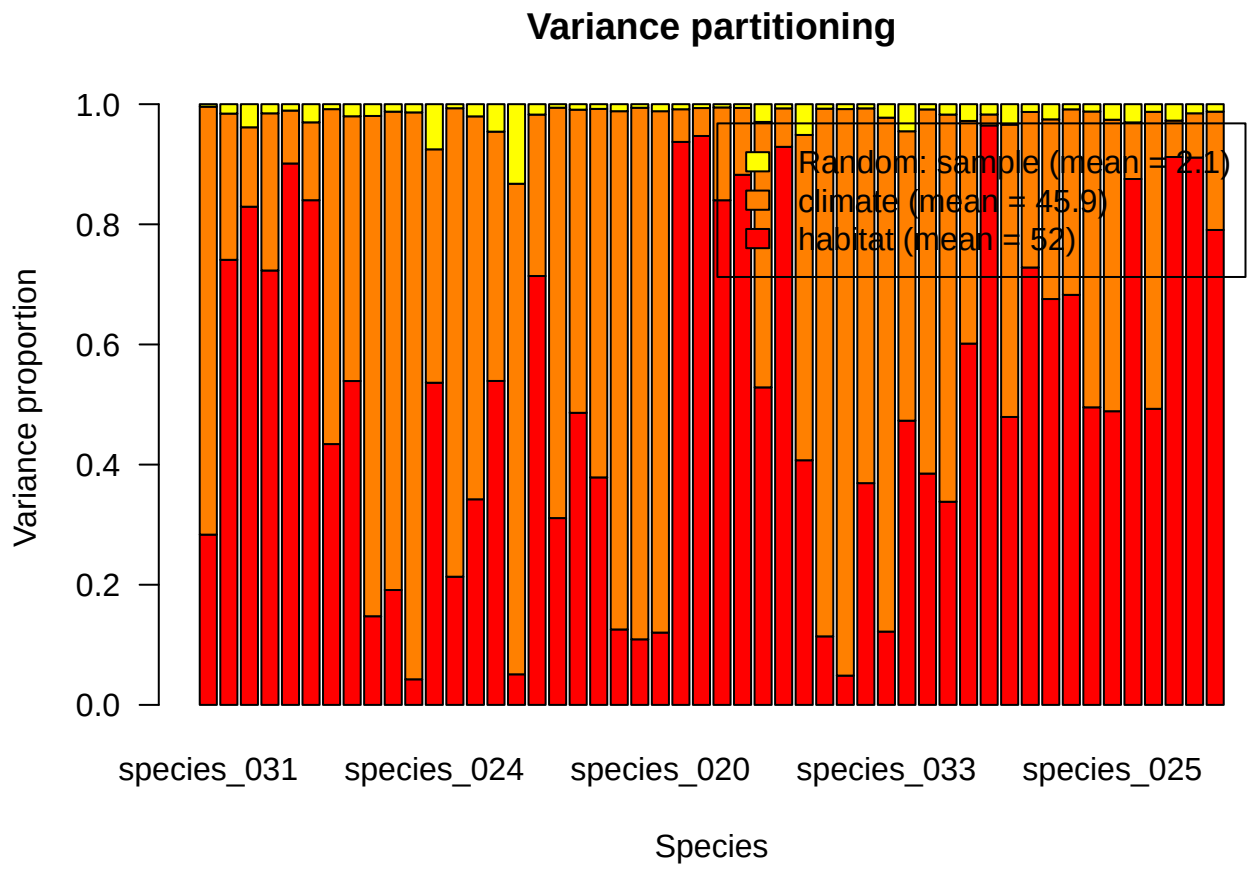

Figure 16: Variance partitioning for each of 50 species

This model is very similar to the full model in terms of the variance partitioning. Let us then ask how much the traits explain out of the variation among the species in their responses to environmental covariates.

```
kable(VP$R2T$Beta)
```

	x
(Intercept)	0.6985090
habitatopen	0.9792336
poly(climate, degree = 2, raw = TRUE)1	0.2224042
poly(climate, degree = 2, raw = TRUE)2	0.0042489

As expected, now the traits do not explain how species respond to climatic variation. Let us now ask how much the traits explain out of their variation in species abundances.

```
VP$R2T$Y
```

```
## [1] 0.5097989
```

The traits now explain a smaller part of the variation in abundance than was the case for the full model. This is because we are missing some relevant traits that would have explained the remaining part.

Finally, we examine the posterior distribution for the phylogenetic signal parameter.

```
mpost = convertToCodaObject(m)
summary(mpost$Rho)
```

```
##
## Iterations = 510:10500
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##       0.727855       0.071013       0.001588       0.001737
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75% 97.5%
##  0.56  0.69  0.73  0.78  0.85
```

Now residual variation in species niches (not explained by the traits) include a phylogenetic signal. This is generated by the phylogenetic structure of the missing trait of thermal optimum.

Changing prior distribution for the species loadings

The prior for the species loadings is the multiplicative gamma process shrinking prior that Bhattacharya and Dunson (2011) proposed for modelling of high-dimensional covariance matrices. Of particular interest is the a_1 and a_2 parameters of the prior, which controls the amount of shrinkage applied. The default values are $a = (50, 50)$. Increasing the value of the a_1 parameter increases shrinkage in general, whereas increasing the value of the a_2 parameter increasese how much additional shrinkage is applied to factor $h + 1$ compared to factor h.

To illustrate the consequences of this choice for the inference of residual species associations, we will refit the model with $a_1 = a_2 = 5$ (less shrinkage), and $a1 = a2 = 500$ (more shrinkage). To do this, we use the `setPriors` function.

```
XFormula.1 = ~poly(climate, degree = 2, raw = TRUE)
rL=setPriors(rL, a1=5, a2=5)
str(rL)
```

```
## List of 17
##  $ pi           : Factor w/ 200 levels "sample_001","sample_002",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ s            : NULL
##  $ sDim         : num 0
##  $ spatialMethod: NULL
##  $ x            : NULL
##  $ xDim         : num 0
##  $ N            : int 200
##  $ distMat      : NULL
##  $ nfMax        : num 15
##  $ nfMin        : num 2
##  $ nNeighbours  : NULL
##  $ nu           : num 3
##  $ a1           : num 5
##  $ b1           : num 1
##  $ a2           : num 5
##  $ b2           : num 1
##  $ alphapw      : NULL
##  - attr(*, "class")= chr "HmscRandomLevel"
```

```
ma5 = Hmsc(Y=Y, XData=XData, XFormula = XFormula.1,
         TrData = traits, TrFormula = TrFormula,
         phyloTree = phy,
         studyDesign=studyDesign, ranLevels=list(sample=rL))

ma5 = sampleMcmc(ma5, thin = thin, samples = samples, transient = transient,
                    nChains = nChains, nParallel = nChains, verbose = verbose)
```

```
## Setting updater$Gamma2=FALSE due to specified phylogeny matrix
```

```
XFormula.1 = ~poly(climate, degree = 2, raw = TRUE)
rL = HmscRandomLevel(units = studyDesign$sample)
rL=setPriors(rL, a1=500, a2=500)
str(rL)
```

```
## List of 17
##  $ pi           : Factor w/ 200 levels "sample_001","sample_002",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ s            : NULL
##  $ sDim         : num 0
##  $ spatialMethod: NULL
##  $ x            : NULL
##  $ xDim         : num 0
##  $ N            : int 200
##  $ distMat      : NULL
##  $ nfMax        : num Inf
##  $ nfMin        : num 2
##  $ nNeighbours  : NULL
##  $ nu           : num 3
##  $ a1           : num 500
##  $ b1           : num 1
##  $ a2           : num 500
```

```
##  $ b2           : num 1
##  $ alphapw      : NULL
##  - attr(*, "class")= chr "HmscRandomLevel"
```

```
ma500 = Hmsc(Y=Y, XData=XData, XFormula = XFormula.1,
         TrData = traits, TrFormula = TrFormula,
         phyloTree = phy,
         studyDesign=studyDesign, ranLevels=list(sample=rL))

ma500 = sampleMcmc(m, thin = thin, samples = samples, transient = transient,
                   nChains = nChains, nParallel = nChains, verbose = verbose)
```

```
## Setting updater$Gamma2=FALSE due to specified phylogeny matrix
```

```
par(mfrow=c(1,3), mar=c(0,0,0,0))

OmegaCor = computeAssociations(ma5)
supportLevel = 0.95
toPlot = ((OmegaCor[[1]]$support>supportLevel)
          + (OmegaCor[[1]]$support<(1-supportLevel))>0)*OmegaCor[[1]]$mean
corrplot(toPlot, method = "color",
         col=colorRampPalette(c("blue","white","red"))(200),
         tl.cex=.6, tl.col="black",
         title="a1 = a2 = 5", mar=c(0,0,1,0))

OmegaCor = computeAssociations(ma50)
toPlot = ((OmegaCor[[1]]$support>supportLevel)
          + (OmegaCor[[1]]$support<(1-supportLevel))>0)*OmegaCor[[1]]$mean
corrplot(toPlot, method = "color",
         col=colorRampPalette(c("blue","white","red"))(200),
         tl.cex=.6, tl.col="black",
         title="a1 = a2 = 50", mar=c(0,0,1,0))

OmegaCor = computeAssociations(ma500)
toPlot = ((OmegaCor[[1]]$support>supportLevel)
          + (OmegaCor[[1]]$support<(1-supportLevel))>0)*OmegaCor[[1]]$mean
corrplot(toPlot, method = "color",
         col=colorRampPalette(c("blue","white","red"))(200),
         tl.cex=.6, tl.col="black",
         title="a1 = a2 = 500", mar=c(0,0,1,0))
```

By plotting the inferred residual associations side by side, we can see that when we apply very strong shrinkage, the inferred associations disappear. At low to medium shrinkage, the inferred associations remain largely similar. This example illustrates how too much shrinkage can lead to losing the signal. In contrast, too little shrinkage may lead to overfitting and thus modelling also noise, not just signal. Whether the estimated associations model signal or noise can be tested e.g. by conditional cross-validation, i.e. by testing if it helps to account the known occurrences of some species when predicting the occurrences of other species."

Other prior parameters of the `Hmsc` model can also be adjusted, see Appendix S1 for details.

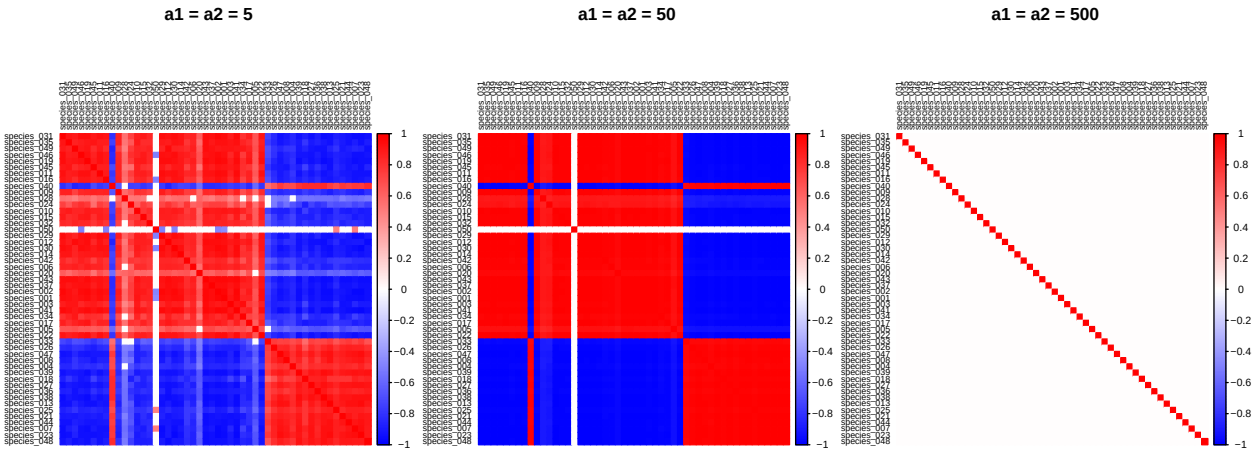

Figure 17: Effects of prior shrinkage values on inferred residual associations