HMSC in practice: the syntax and typical workflow of the R-package Hmsc

4 An Overview of the Structure and Use of HMSC
  4.1 HMSC Is a Multivariate Hierarchical Generalised Linear Mixed Model
  4.2 The Overall Structure of HMSC
  4.3 Linking HMSC to Community Ecology Theory
  4.4 The Overall Workflow for Applying HMSC

39
39
41
45
47
HMSC in practice: the syntax and typical workflow of the R-package Hmsc

Step 1. Setting model structure and fitting the model

m = Hmsc(Y, X0, xformula, trtformu, phyloTree, studyDesign, randomEffects, distr)
sampleHmsc(m, nhs, samples, transient, nchains)

Step 2. Examining MCMC convergence

mpost = convertToDataObject(m)
effectiveSize(mpost)
gelman.diag(mpost)

Step 3. Evaluating model fit and comparing models

predY = computePredictedValues(m, partition)
pp = evaluateModelFit(m, mpost)
WAIC = computeWAIC(m)

Step 4. Exploring parameter estimates

Species richness and their links to traits and phylogenies
Biotic interactions, dispersal limitations, missing covariates and ecological drift

Step 5. Making predictions

$x_{new}$: Predictor values of environmental covariates, e.g. representing an environmental gradient
$y_{new}$: Predicted community
$z_{new}$: Predictor values of spatio-temporal context, e.g. spatial coordinates of where predictions are to be made

Figure 4.3 The five steps of a typical workflow of HMSC analyses. The computer code in Steps 1–3 illustrates the syntax of the R-package Hmsc. The graph in Step 2 shows an MCMC trace plot, and the graphs in Step 4 illustrate the estimates of some key model parameters.
Step 1. Setting model structure and fitting the model

\[ m = \text{Hmuc}(Y, X_{\text{Data}}, X_{\text{Formula}}, T_{\text{Data}}, T_{\text{Formula}}, \text{phyloTree}, \text{studyDesign}, \text{ranLevels}, \text{distr}) \]

\[
\text{sampleMcmc}(m, \text{thin}, \text{samples}, \text{transient}, \text{nChains})
\]
Step 2. Examining MCMC convergence

Step 2. Examining MCMC convergence

\[ mpost = \text{convertToCodaObject}(m) \]

\[ \text{effectiveSize}(mpost) \]

\[ \text{gelman.diag}(mpost) \]
Step 1. Setting model structure and fitting the model

```
m = Hmsc(Y, XData, XFormula, TrData, TrFormula, phyloTree,
           studyDesign, ranLevels, distr)
```

```
sampleMcmc(m, thin, samples, transient, nChains)
```

Step 2. Examining MCMC convergence

```
mpost = convertToCodaObject(m)
effectiveSize(mpost)
gelman.diag(mpost)
```

Not satisfactory? Redo model fitting.
Step 3. Evaluating model fit and comparing models

\[
\begin{align*}
predY &= \text{computePredictedValues}(m, \text{partition}) \\
MF &= \text{evaluateModelFit}(m, predY) \\
\text{WAIC} &= \text{computeWAIC}(m)
\end{align*}
\]
Step 3. Evaluating model fit and comparing models

Step 1. Setting model structure and fitting the model

\[ m = \text{Hmsc}(Y, XData, XFormula, TrData, TrFormula, phyloTree, \]
\text{studyDesign, ranLevels, distr)}

\[ \text{sampleMcmc}(m, \text{thin, samples, transient, nChains}) \]

Step 2. Examining MCMC convergence

\[ \text{mpost} = \text{convertToCodaObject}(m) \]

\[ \text{effectiveSize}(\text{mpost}) \]

\[ \text{gelman.diag}(\text{mpost}) \]

Step 3. Evaluating model fit and comparing models

\[ \text{predY} = \text{computePredictedValues}(m, \text{partition}) \]

\[ \text{MF} = \text{evaluateModelFit}(m, \text{predY}) \]

\[ \text{WAIC} = \text{computeWAIC}(m) \]
Step 4. Exploring parameter estimates

post =getPostEstimate(m, parName="Beta")
plotBeta(m, post, supportLevel)

OmegaCor = computeAssociations(m)
corrplot(OmegaCor)

VP = computeVariancePartitioning(m)
plotVariancePartitioning(m, VP)
Step 5. Making predictions

Gradient = constructGradient(m, focalVariable)
pred = predict(m, Gradient)
plotGradient(m, Gradient, pred, measure, showData)