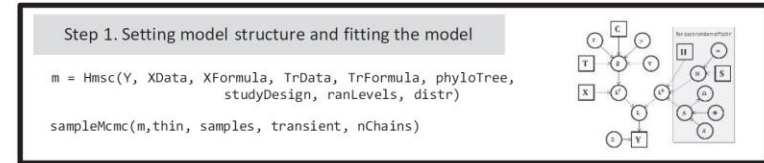


HMSC in practice: the syntax and typical workflow of the R-package Hmsc

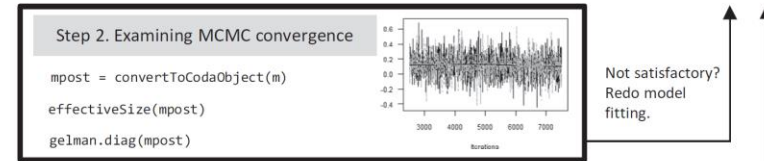
4	An Overview of the Structure and Use of HMSC	39
4.1	HMSC Is a Multivariate Hierarchical Generalised Linear Mixed Model	39
4.2	The Overall Structure of HMSC	41
4.3	Linking HMSC to Community Ecology Theory	45
4.4	The Overall Workflow for Applying HMSC	47

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

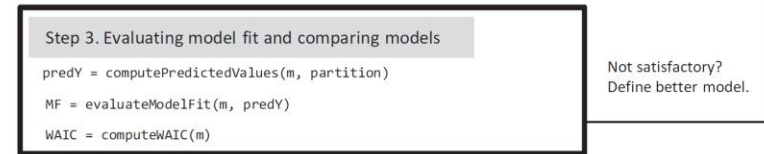
Step 1. Setting model structure and fitting the model



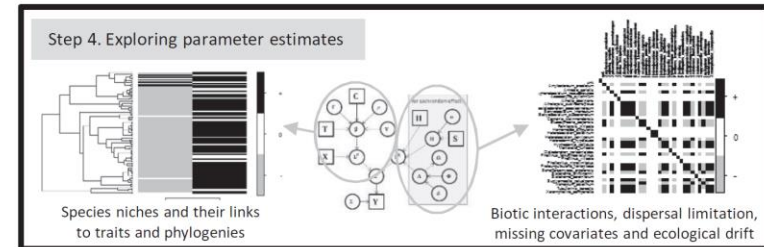
Step 2. Examining MCMC convergence



Step 3. Evaluating model fit and comparing models



Step 4. Exploring parameter estimates



Step 5. Making predictions

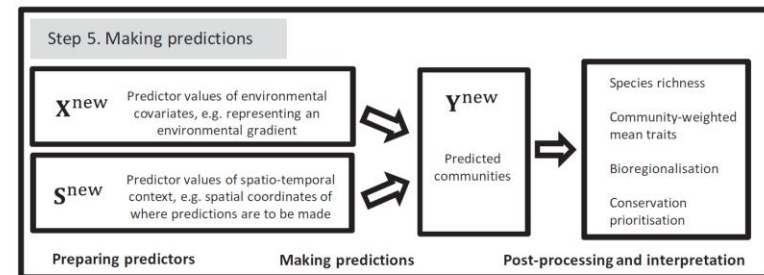


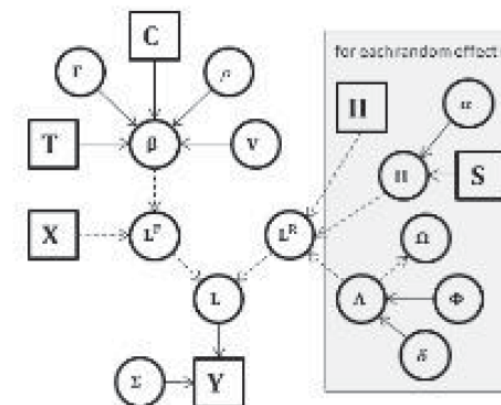
Figure 4.3 The five steps of a typical workflow of HMSC analyses. The computer code in Steps 1–3 illustrates the syntax of the R-package Hmsc. The graph in Step 2 shows an MCMC trace plot, and the graphs in Step 4 illustrate the estimates of some key model parameters.

Step 1. Setting model structure and fitting the model

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)
```



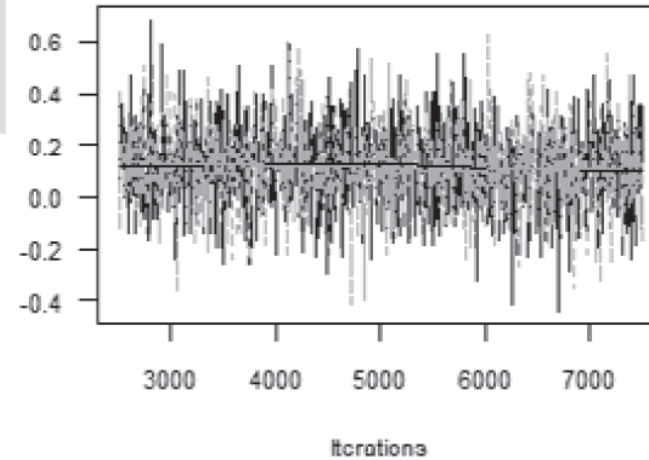
Step 2. Examining MCMC convergence

Step 2. Examining MCMC convergence

```
mpost = convertToCodaObject(m)
```

```
effectiveSize(mpost)
```

```
gelman.diag(mpost)
```

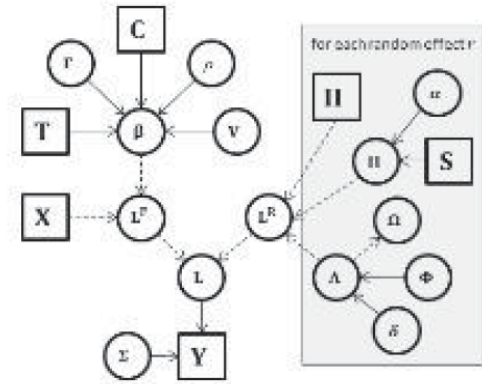


Step 2. Examining MCMC convergence

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)
```

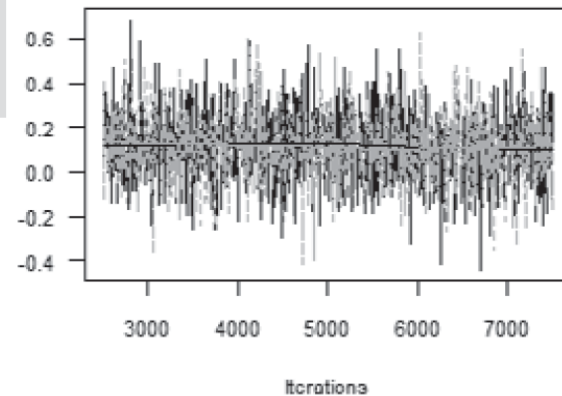


Step 2. Examining MCMC convergence

```
mpost = convertToCodaObject(m)
```

```
effectiveSize(mpost)
```

```
gelman.diag(mpost)
```



Not satisfactory?
Redo model
fitting.

Step 3. Evaluating model fit and comparing models

Step 3. Evaluating model fit and comparing models

```
predY = computePredictedValues(m, partition)
```

```
MF = evaluateModelFit(m, predY)
```

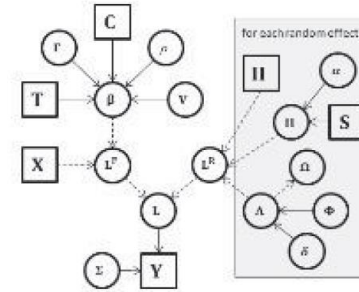
```
WAIC = computeWAIC(m)
```

Step 3. Evaluating model fit and comparing models

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)
```

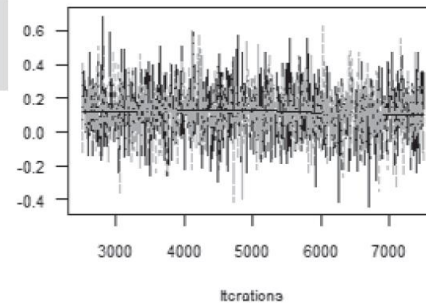


Step 2. Examining MCMC convergence

```
mpost = convertToCodaObject(m)
```

```
effectiveSize(mpost)
```

```
gelman.diag(mpost)
```



Not satisfactory?
Redo model
fitting.

Step 3. Evaluating model fit and comparing models

```
predY = computePredictedValues(m, partition)
```

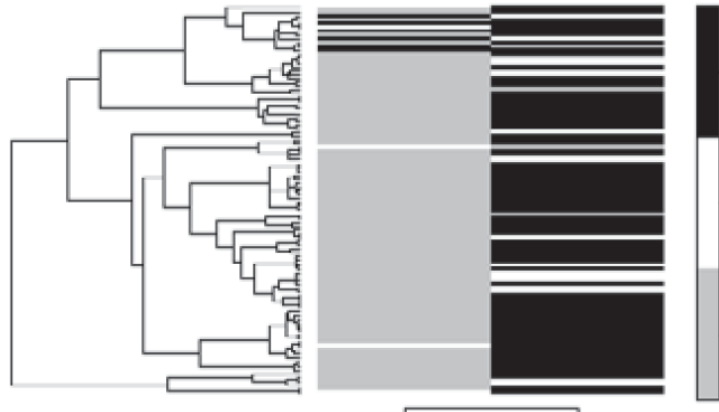
```
MF = evaluateModelFit(m, predY)
```

```
WAIC = computeWAIC(m)
```

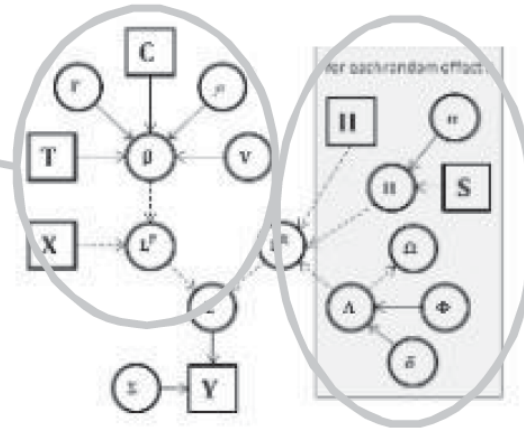
Not satisfactory?
Define better model.

Step 4. Exploring parameter estimates

Step 4. Exploring parameter estimates



Species niches and their links
to traits and phylogenies



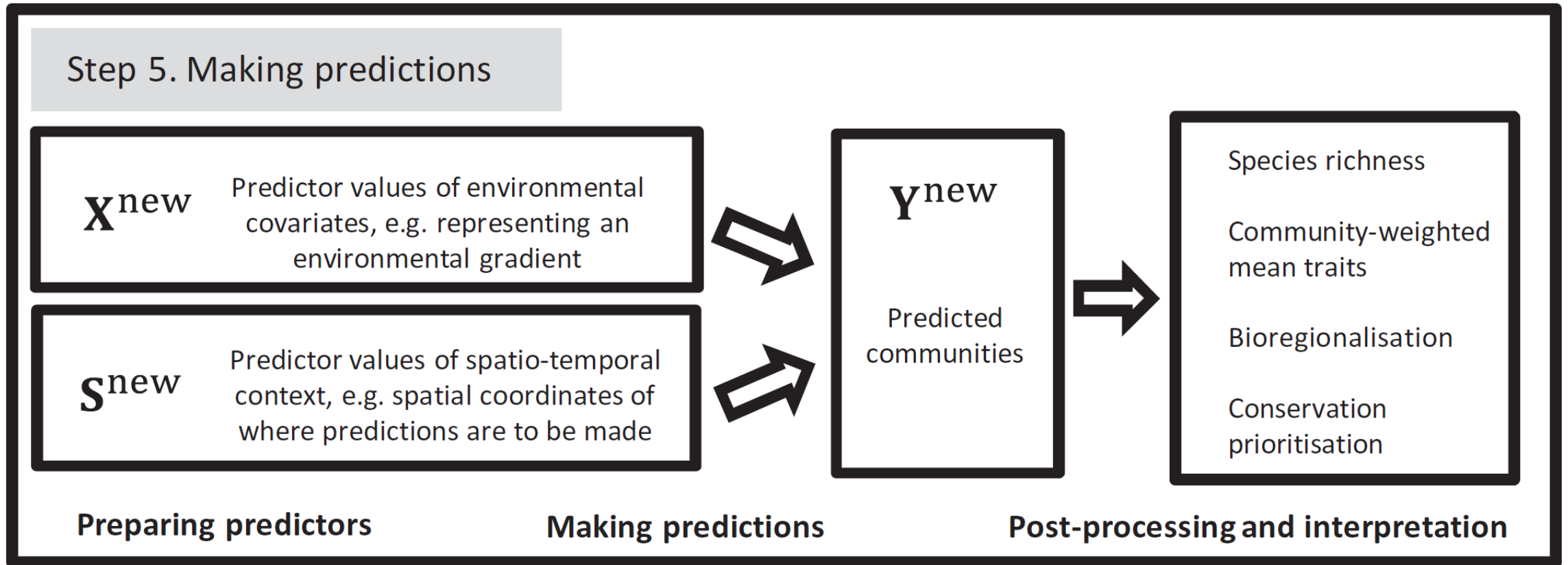
Biotic interactions, dispersal limitation,
missing covariates and ecological drift

```
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)
```