Bayesian AMMI models for ordinal data

Cristian Tiago Erazo Mendes[*] Carlos Pereira da Silva[†]
Julio Silvio de Sousa Bueno Filho[‡] Fabio Mathias Correa[§]

03 February, 2022

Contents

1 Introduction

The focus of this vignette is to present the *ammiBayes* package for analyzing Multi Environment Trials. The package was developed so that each MCMC chain runs in parallel on a Linux SO. For Windows system the code will be running serially.

Diagnosis of MCMC chains can be viewed with tha *bayesplot* package.

2 Setup

In addition to **ammiBayes** we'll load the package **bayesplot** and **coda**

```
library(bayesplot)
library(ammiBayes)
library(coda)
```

[*] Universidade Federal de Lavras, tiagoerazo@gmail.com
[†] Universidade Federal de Lavras, ccpsilva81@hotmail.com
[‡] Universidade Federal de Lavras, jssbueno@ufla.br
[§] Rhodes University, f.correa@ru.ac.za, Maintainer

3 Example

```r
data(ammiData) # see help("ammiData")

head(ammiData)
```

```
##   amb rep gen  prod
## 1   1   1   1 1.794
## 2   1   1   2 1.134
## 3   1   1   3 0.718
## 4   1   1   4 1.852
## 5   1   1   5 2.245
## 6   1   1   6 1.111
```

Vectors that define the effects of genotypes, environments and repetitions should be defined as factors.

```r
Env  <- factor(ammiData$amb)
Rep <- factor(ammiData$rep)
Gen  <- factor(ammiData$gen)
Y  <- ammiData$prod
```

We are using default prioris. To change the prioris see: `help("ammiBayes")`

```r
model <- ammiBayes(Y=Y, Gen=Gen, Env=Env, Rep=Rep, iter=4000,
                                     burn=500, jump=5, chains=4)
```

```r
summary(model)
```

```
## Additive Main Effects and Multiplicative Interaction Model
##
## Total of genotypes: 12
##
## Random effects
##             Mean St.dev LI(2.5%) LS(97.5%)
## Var.gen   0.1499 0.0863   0.0434    0.3023
## Var.error 0.0363 0.0058   0.0258    0.0478
##
## Lambda
##       Mean St.dev LI(2.5%) LS(97.5%)
## L.1 1.7719 0.1107   1.5605    1.9865
## L.2 0.8157 0.1152   0.5873    1.0320
## L.3 0.4010 0.1348   0.1291    0.6567
##
##                        L.1      L.2     L.3
## Acumulated.lambda 1.771935 2.587639 2.98862
##
## Explained
##                       D.1       D.2        D.3
## Percentage      0.7916849 0.1677730 0.04054208
## Perc.acumulated 0.7916849 0.9594579 1.00000000
##
## Elapsed time: 4.258 minutes
##
## Iterations: 4000 Jump: 5 Burn: 500
```

4 Diagnosis

The extraction of the MCMC chains must be defined for each object of interest. See `help("diagnosis.ammiBayes")`.

```
# Genotype chains
diag.gen <- diagnosis.ammiBayes(model, pars="Genotype")

# PC1 for environment
diag.env1 <- diagnosis.ammiBayes(model, pars="Env.PC1")

# PC2 for environment
diag.env2 <- diagnosis.ammiBayes(model, pars="Env.PC2")
```

For each object extracted with the diagnosis.ammiBayes function, it is possible to use the functions of the bayesplot and coda package to diagnose the chains.

4.1 Diagnosis with coda package

```
gelman.diag(diag.gen)
```

```
## Potential scale reduction factors:
##
##    Point est. Upper C.I.
## 1        1.01       1.03
## 2        1.01       1.03
## 3        1.01       1.03
## 4        1.01       1.04
## 5        1.01       1.03
## 6        1.01       1.03
## 7        1.01       1.03
## 8        1.01       1.03
## 9        1.01       1.03
## 10       1.01       1.03
## 11       1.02       1.04
## 12       1.01       1.03
##
## Multivariate psrf
##
## 1.01
```

```
raftery.diag(diag.gen)
```

```
## [[1]]
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##     Burn-in  Total Lower bound  Dependence
##     (M)      (N)   (Nmin)       factor (I)
##  1  18       21318 3746         5.69
##  2  20       21590 3746         5.76
##  3  24       26904 3746         7.18
##  4  24       25032 3746         6.68
##  5  20       20630 3746         5.51
##  6  20       22258 3746         5.94
##  7  20       23570 3746         6.29
##  8  36       37260 3746         9.95
##  9  18       17810 3746         4.75
##  10 18       20506 3746         5.47
##  11 24       24764 3746         6.61
##  12 28       36908 3746         9.85
##
##
## [[2]]
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##     Burn-in  Total Lower bound  Dependence
##     (M)      (N)   (Nmin)       factor (I)
##  1  28       34056 3746         9.09
##  2  21       17763 3746         4.74
##  3  22       23786 3746         6.35
##  4  20       26320 3746         7.03
##  5  27       30345 3746         8.10
##  6  32       30588 3746         8.17
##  7  36       37260 3746         9.95
##  8  18       20199 3746         5.39
##  9  27       30753 3746         8.21
##  10 28       30628 3746         8.18
##  11 24       31248 3746         8.34
##  12 24       26598 3746         7.10
##
##
## [[3]]
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##     Burn-in  Total Lower bound  Dependence
##     (M)      (N)   (Nmin)       factor (I)
##  1  10       12636 3746         3.37
##  2  10       12266 3746         3.27
##  3  14       14620 3746         3.90
##  4  12       13260 3746         3.54
##  5  18       18042 3746         4.82
##  6  18       18372 3746         4.90
##  7  10       12356 3746         3.30
##  8  12       11922 3746         3.18
##  9  12       16310 3746         4.35
##  10 14       16226 3746         4.33
##  11 12       15734 3746         4.20
##  12 14       16902 3746         4.51
##
##
## [[4]]
##
```

```
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##     Burn-in  Total Lower bound  Dependence
##     (M)      (N)   (Nmin)       factor (I)
##  1  21       22953 3746         6.13
##  2  16       16992 3746         4.54
##  3  24       22388 3746         5.98
##  4  28       30868 3746         8.24
##  5  14       13706 3746         3.66
##  6  21       22953 3746         6.13
##  7  20       23066 3746         6.16
##  8  21       21186 3746         5.66
##  9  18       19743 3746         5.27
##  10 20       19950 3746         5.33
##  11 16       17662 3746         4.71
##  12 12       10794 3746         2.88
```

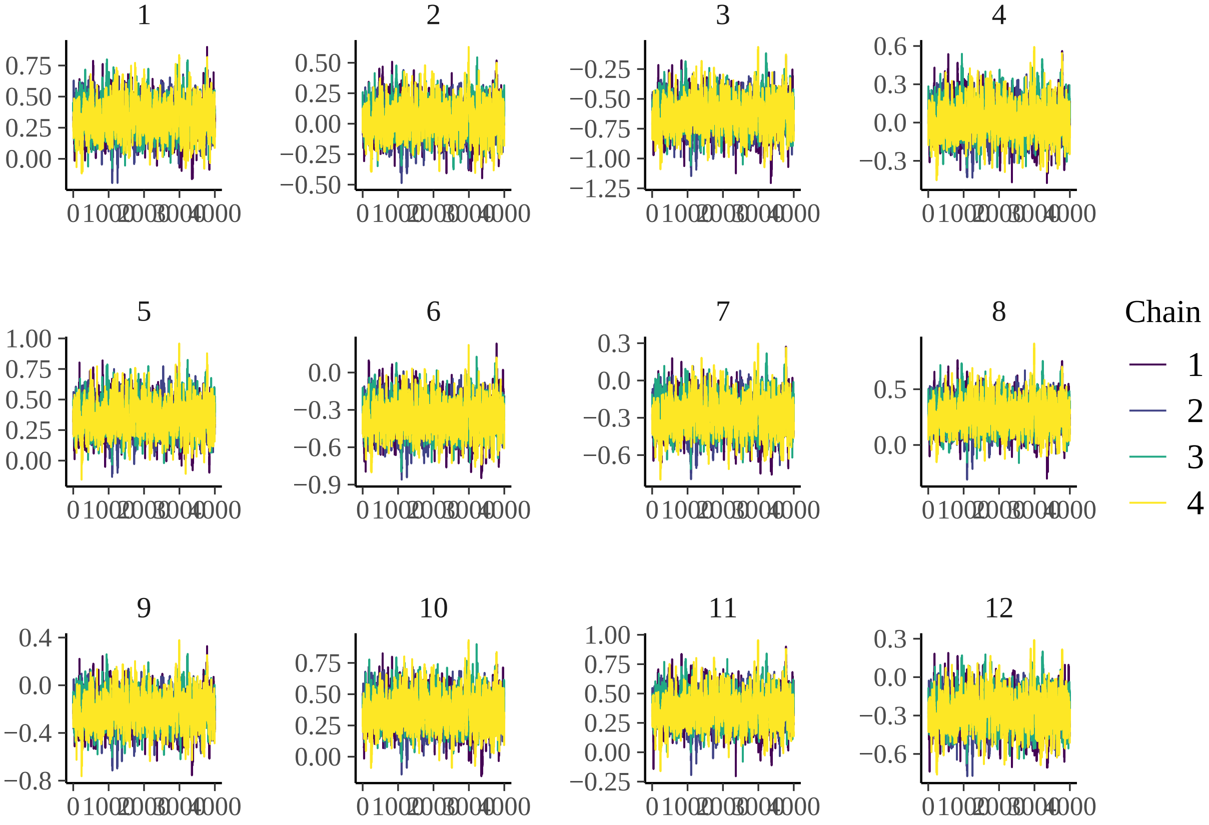
4.2 Diagnosis with bayesplot package

We will proceed only with the chain for genotypes.

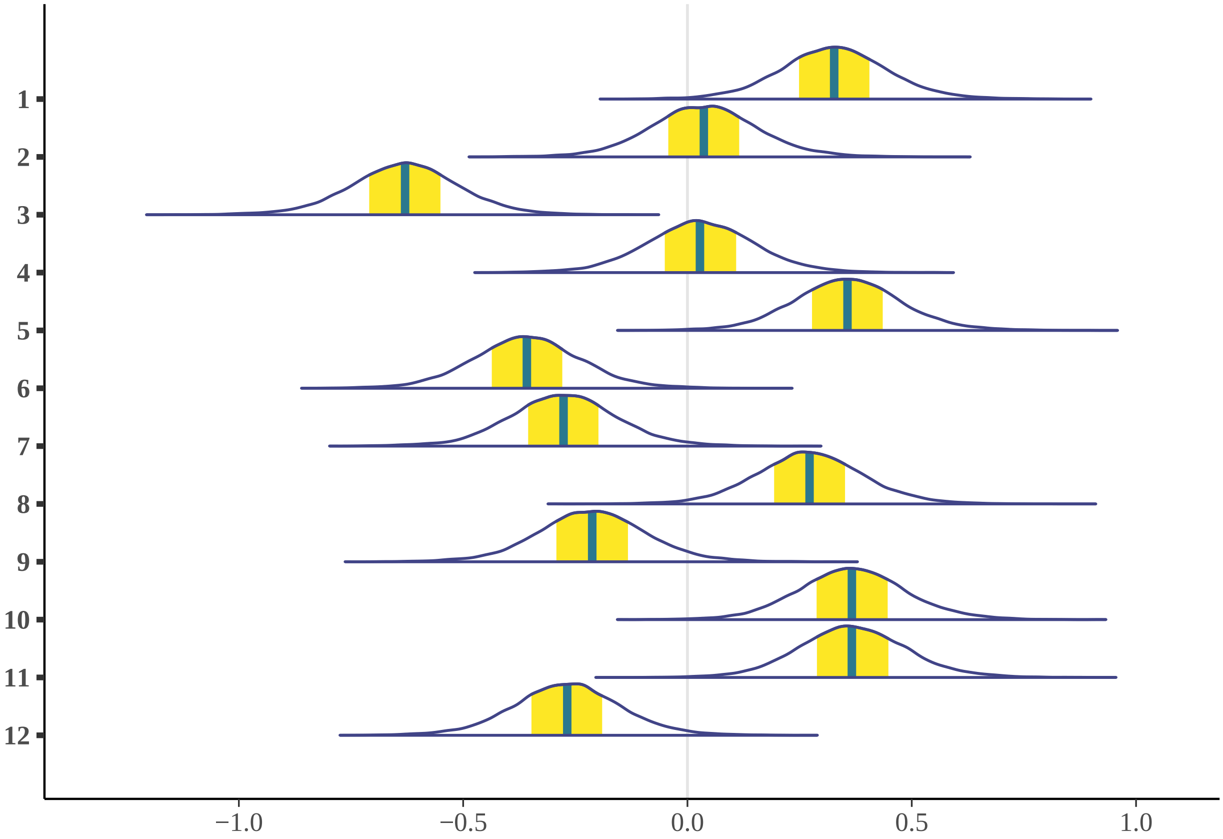
For more examples you can see: bayesplot

```
color_scheme_set("viridis")
```

```
mcmc_trace(diag.gen)
```

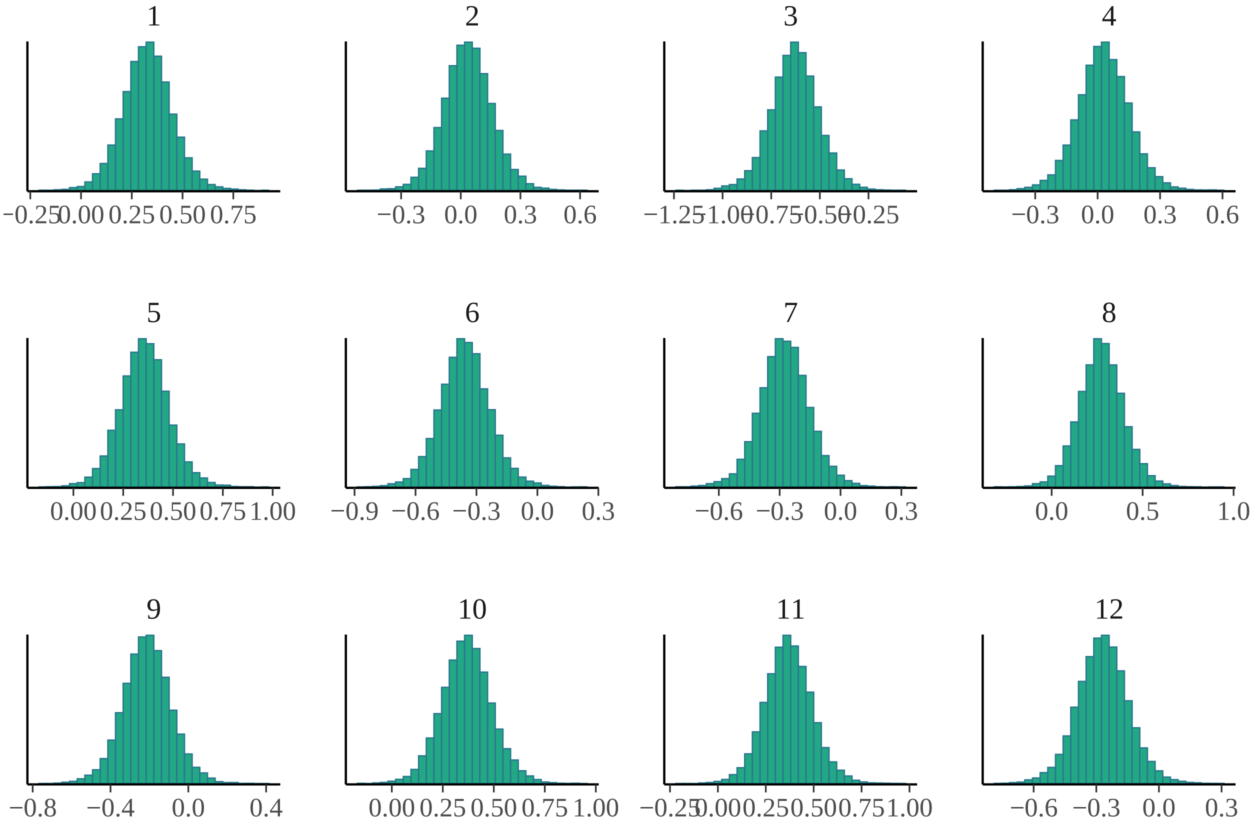


```
mcmc_areas(diag.gen)
```

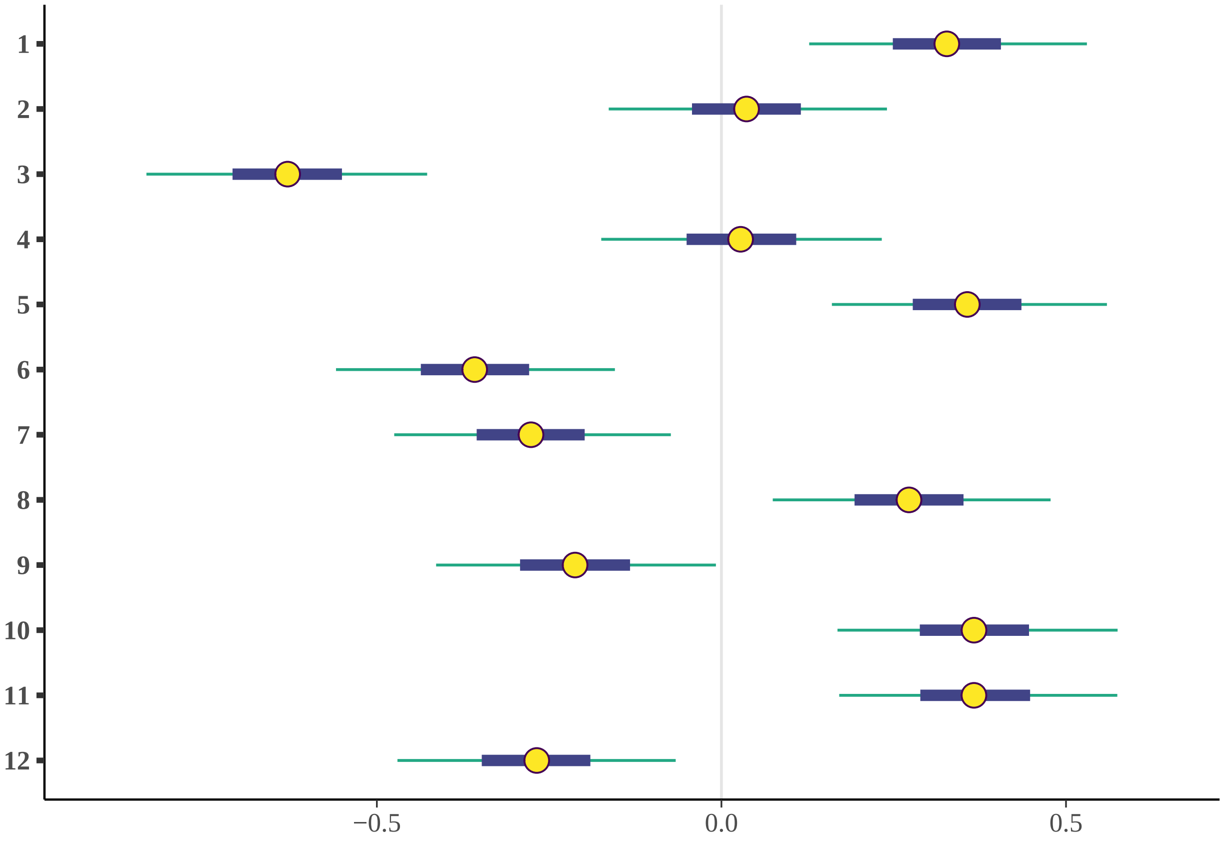


```
mcmc_hist(diag.gen)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mcmc_intervals(diag.gen)
```



5 Prediction

Extract the predict values.

```
pred <- predict(model)
head(pred)
```

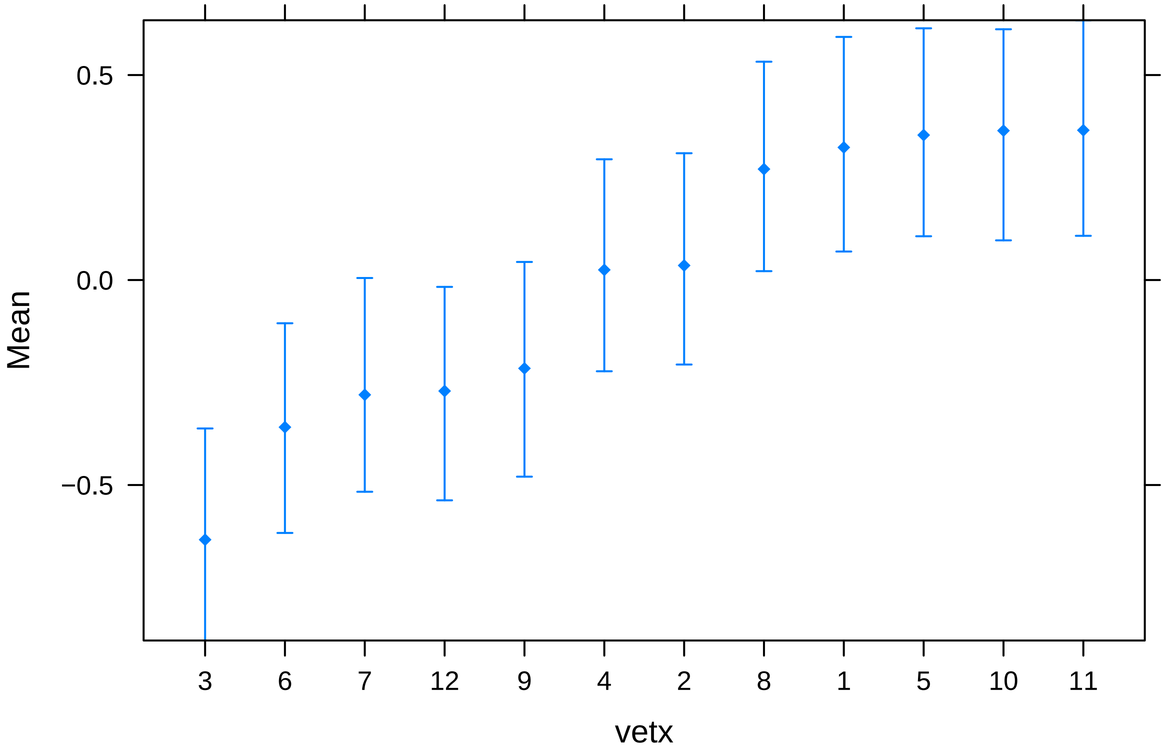
```
##        Mean    Median      2.5%     97.5%
## 1 2.0282145 2.0290942 1.8205803 2.2526869
## 2 1.4765334 1.4762873 1.2305909 1.6897226
## 3 0.6807776 0.6797665 0.4607497 0.8994324
## 4 1.6652385 1.6641988 1.4383110 1.8688034
## 5 2.1927018 2.1923872 1.9750210 2.4059077
## 6 1.0402466 1.0399317 0.8077527 1.2503860
```

6 Genotype effect

Extract the genotype effects. For more details you can see: gen.effects and ammiBayes.gen.plot.

```
gen.ef <- gen.effects(model)
```

```
ammiBayes.gen.plot(gen.ef)
```

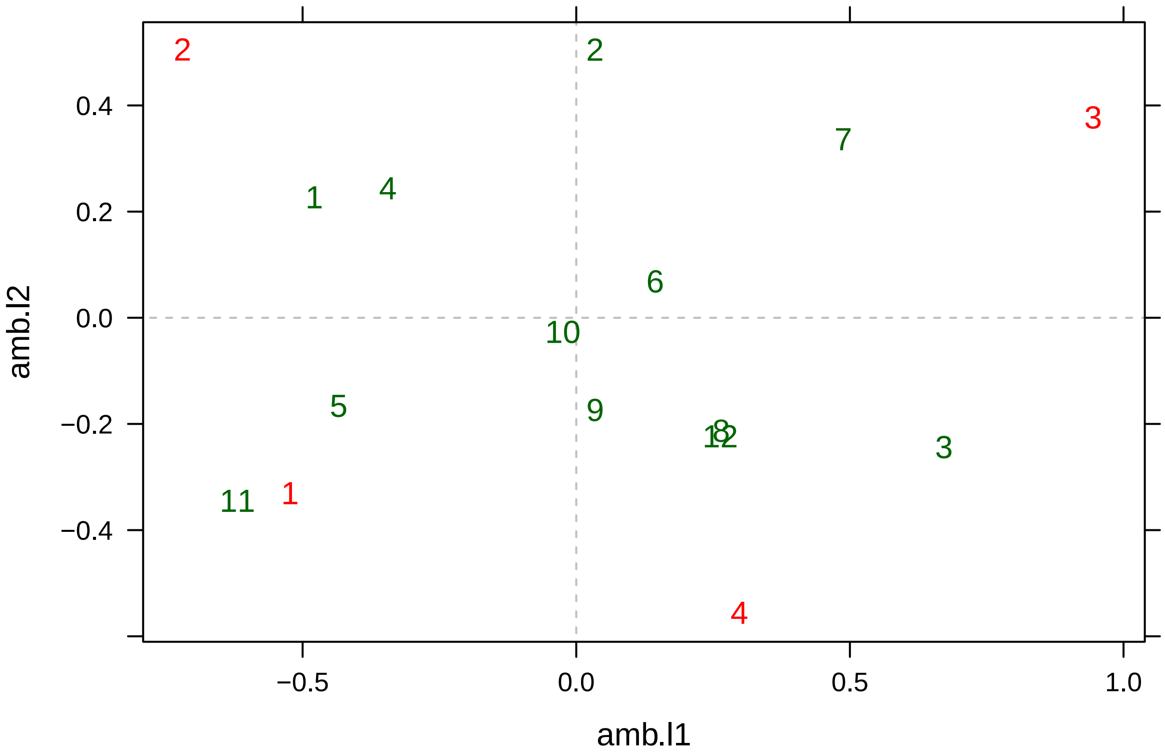


7 Plot functions

7.1 Plot means

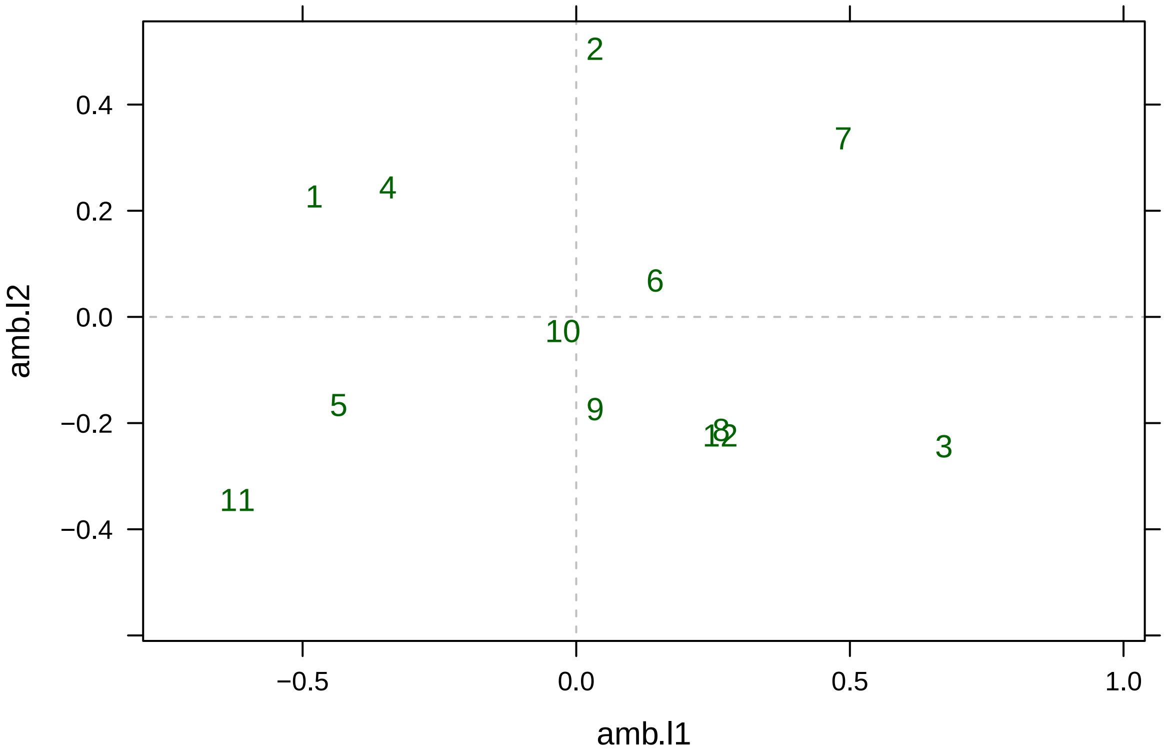
By default all genotypes and environments are plotted. For more details see `help("ammiBayes.mean.plot")`

```
ammiBayes.mean.plot(model)
```



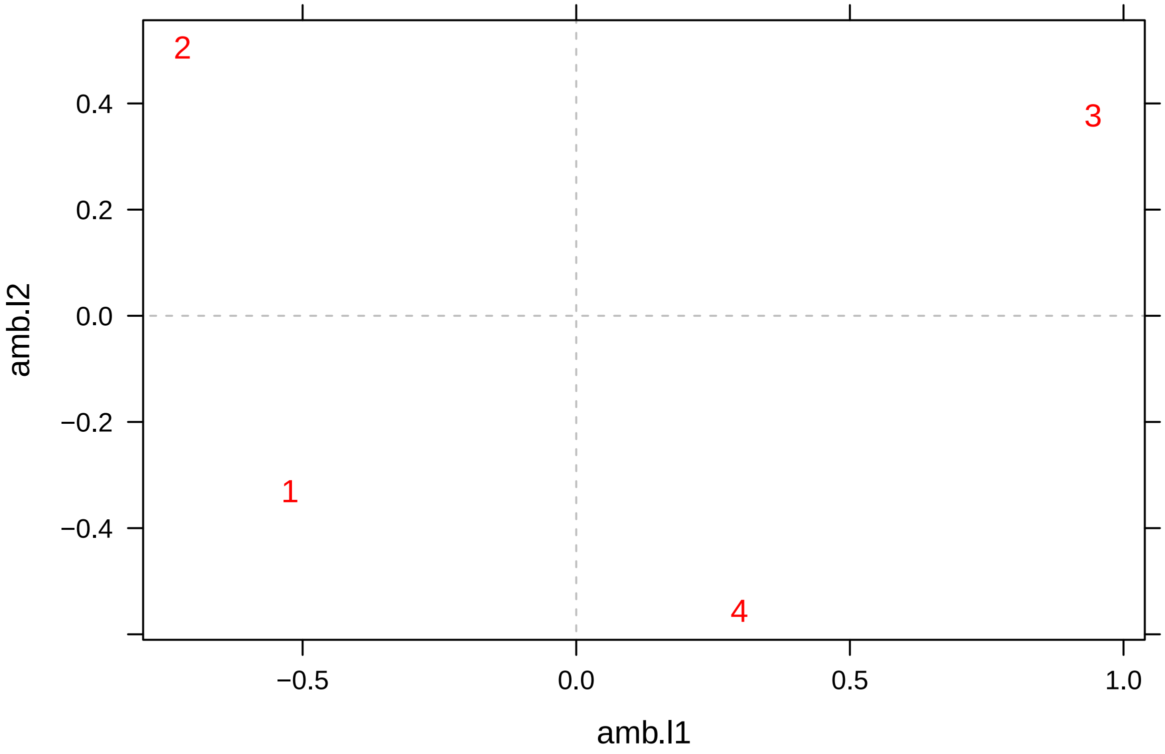
Plotting only the effects of genotypes

```
ammiBayes.mean.plot(model, col.text.env="transparent")
```



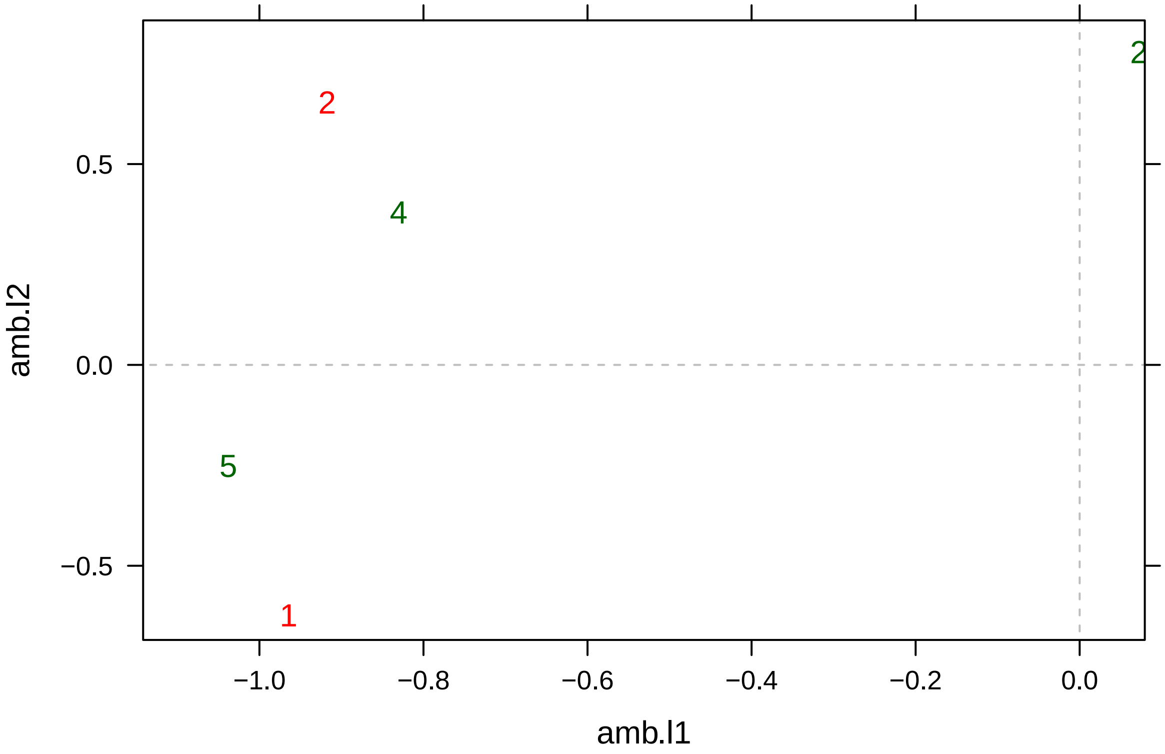
Plotting only the effects of environments

```
ammiBayes.mean.plot(model, col.text.gen="transparent")
```



Specifying the effects of genotypes and environments

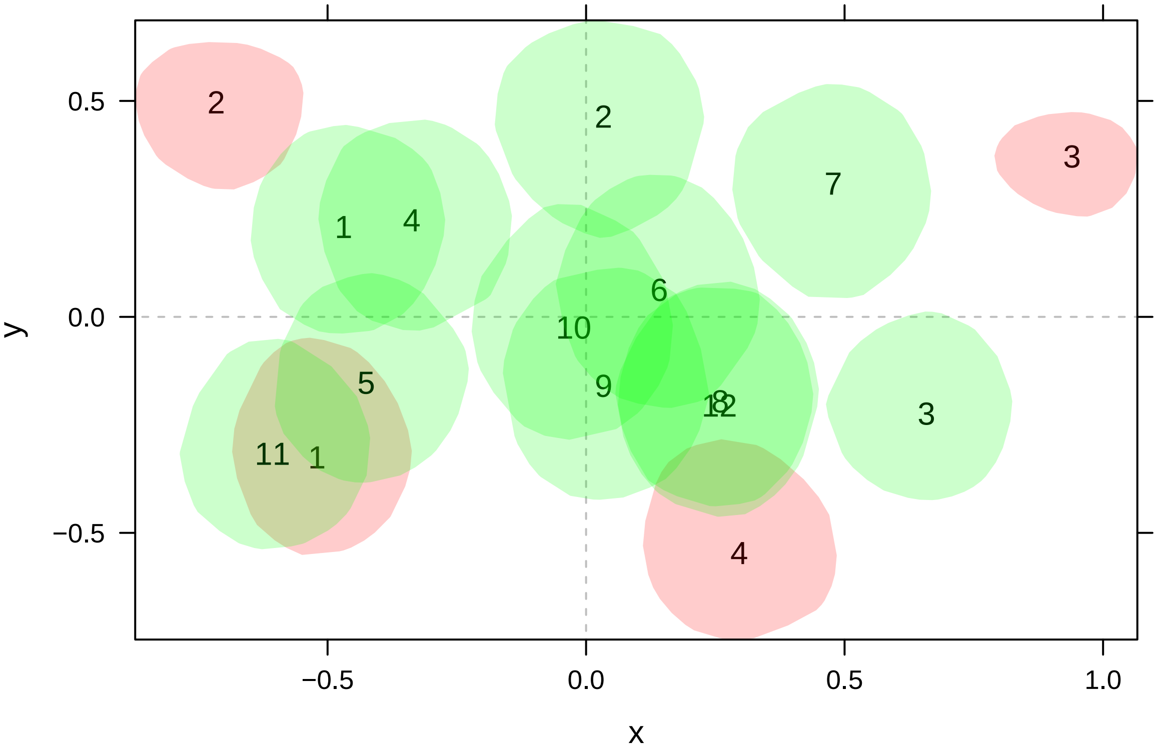
```
ammiBayes.mean.plot(model, pars.env=c("1","2"), pars.gen=c("2","4","5"))
```



7.2 Confidence regions

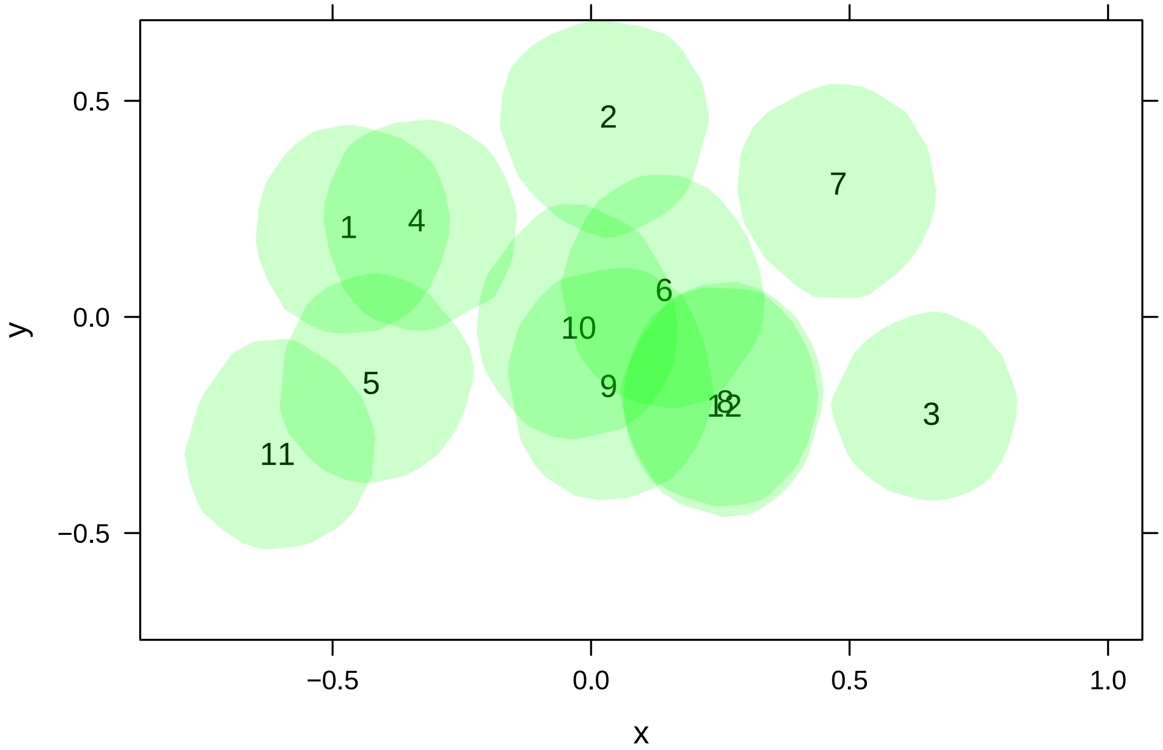
By default, all genotypes and environments are plotted with confidence regions of the 95%. For more details see: ammiBayes.conf.plot.

```
ammiBayes.conf.plot(model)
```



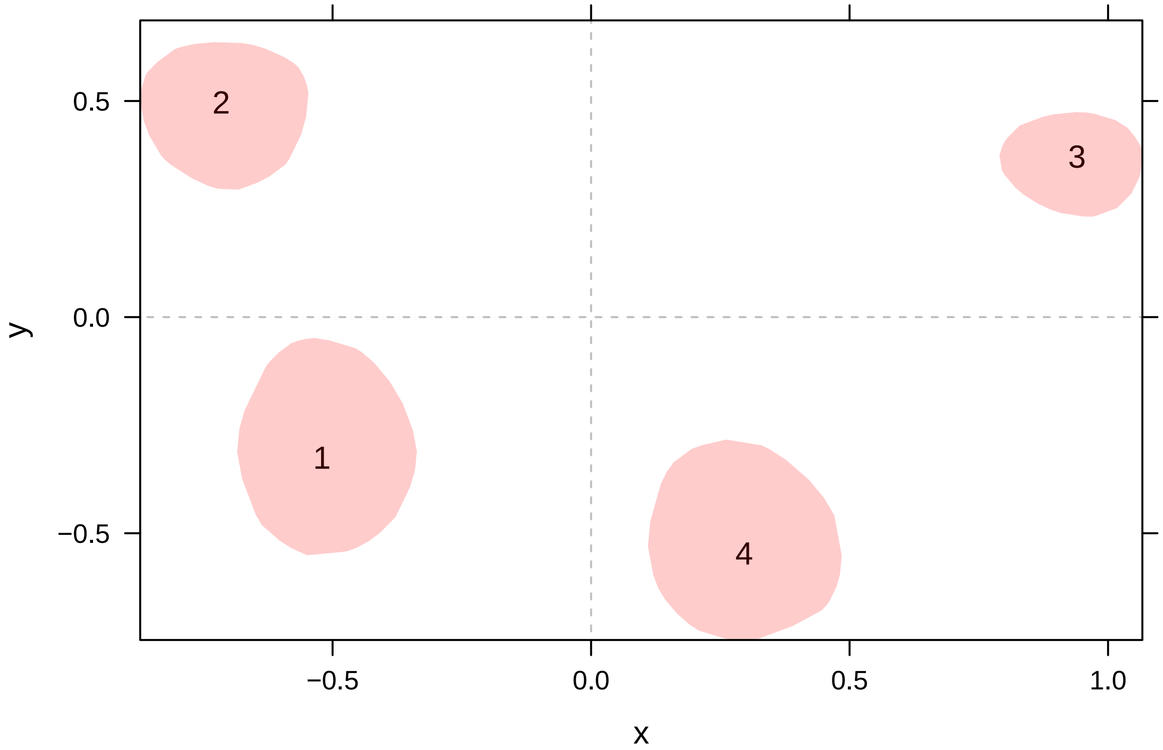
Plotting only the effects of genotypes

```
ammiBayes.conf.plot(model, plot.env=FALSE)
```



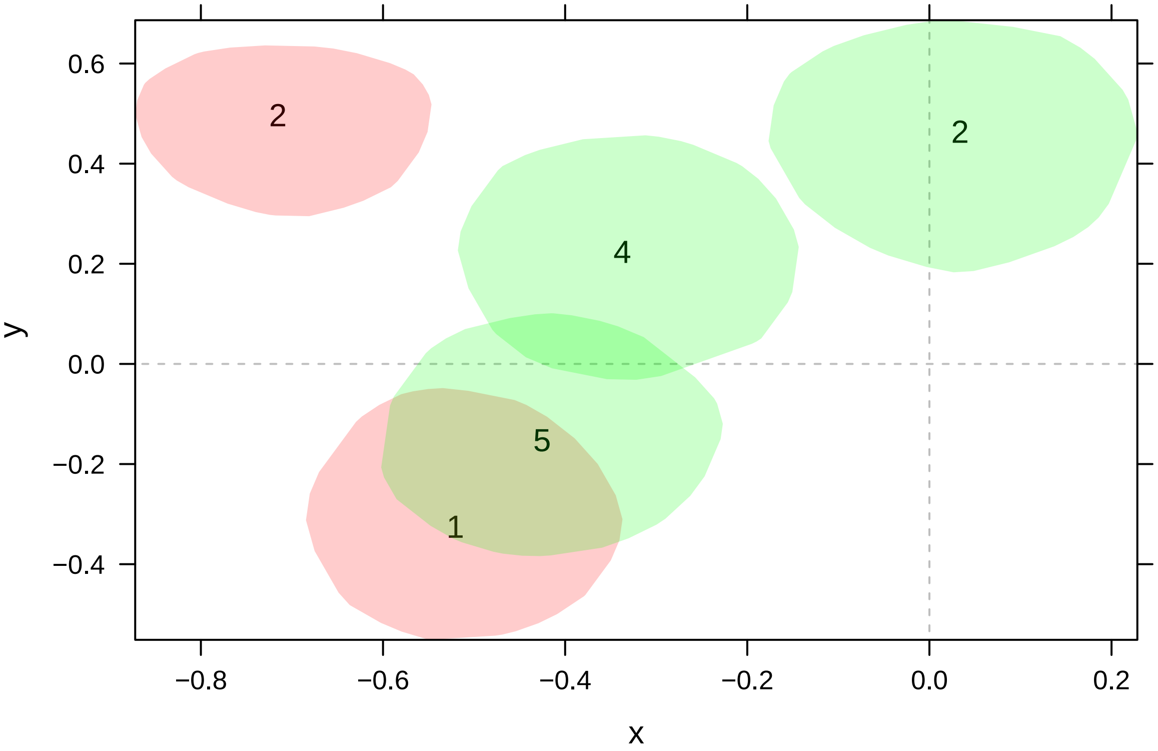
Plotting only the effects of environments

```
ammiBayes.conf.plot(model, plot.gen=FALSE)
```



Specifying the effects of genotypes and environments

```
ammiBayes.conf.plot(model, pars.env=c("1","2"), pars.gen=c("2","4","5"))
```



2
2
4
5
1
0.6
0.4
0.2
0.0
−0.2
−0.4
y
−0.8
−0.6
−0.4
−0.2
0.0
0.2
x