Bayesian AMMI models for continuous data

Luciano A. Oliveira[*] Carlos Pereira da Silva[†] Cristian Tiago Erazo Mendes[‡]
Alessandra Querino da Silva[§] Joel Jorge Nuvunga[¶] Marcio Balestre[‖]
Julio Silvio de Sousa Bueno Filho[**] Fabio Mathias Correa[††]

12 April, 2025

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 the *bayesplot* and *coda* package.

2 Setup

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

---

[*] Universidade Federal da Grande Dourados
[†] Universidade Federal de Lavras
[‡] Universidade Federal de Lavras
[§] Universidade Federal da Grande Dourados
[¶] Eduardo Mondelane University
[‖] Universidade Federal de Lavras
[**] Universidade Federal de Lavras
[††] University of the Free State, mathiascorreaf@ufs.ac.za, Maintainer

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

3 Example

```
library(ammiBayes)
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.

```
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")`

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

```
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   0.1510 0.0857   0.0436    0.3012
## Var.error 0.0363 0.0058   0.0262    0.0479
##
## Lambda
##       Mean St.dev LI(2.5%) LS(97.5%)
## L.1 1.7704 0.1101   1.5466    1.9784
## L.2 0.8121 0.1176   0.5798    1.0375
## L.3 0.4003 0.1305   0.1336    0.6484
##
##                         L.1      L.2      L.3
## Acumulated.lambda 1.770423 2.582528 2.982792
##
## Explained
##                        D.1       D.2        D.3
## Percentage      0.7926909 0.1667916 0.04051755
## Perc.acumulated 0.7926909 0.9594825 1.00000000
```

```
##
## Elapsed time: 8.315 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       1.00
## 2           1       1.01
## 3           1       1.01
## 4           1       1.00
## 5           1       1.01
## 6           1       1.01
## 7           1       1.01
## 8           1       1.00
## 9           1       1.01
## 10          1       1.00
## 11          1       1.01
## 12          1       1.01
##
## Multivariate psrf
##
## 1
```

```
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  20       17052 3746         4.55
##  2  24       34220 3746         9.14
##  3  21       22926 3746         6.12
##  4  28       34056 3746         9.09
##  5  22       24830 3746         6.63
##  6  21       26481 3746         7.07
##  7  21       22953 3746         6.13
##  8  21       21135 3746         5.64
##  9  14       16924 3746         4.52
##  10 18       20400 3746         5.45
##  11 24       28323 3746         7.56
##  12 24       31484 3746         8.40
##
##
## [[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  18       17810 3746          4.75
##  2  24       24608 3746          6.57
##  3  18       21705 3746          5.79
##  4  18       20037 3746          5.35
##  5  20       16608 3746          4.43
##  6  30       33090 3746          8.83
##  7  36       44052 3746         11.80
##  8  24       26598 3746          7.10
##  9  32       37060 3746          9.89
##  10 21       24753 3746          6.61
##  11 21       24753 3746          6.61
##  12 20       24018 3746          6.41
##
##
## [[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  16       19320 3746         5.16
##  2  16       18488 3746         4.94
##  3  16       18520 3746         4.94
##  4  18       18201 3746         4.86
##  5  18       21705 3746         5.79
##  6  21       24753 3746         6.61
##  7  16       16176 3746         4.32
##  8  21       24876 3746         6.64
##  9  16       15286 3746         4.08
##  10 16       16108 3746         4.30
```

```
##   11 12       11922 3746         3.18
##   12 12       12510 3746         3.34
##
##
## [[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  18       23379 3746         6.24
##   2  20       26676 3746         7.12
##   3  18       23265 3746         6.21
##   4  18       20199 3746         5.39
##   5  14       16092 3746         4.30
##   6  14       14664 3746         3.91
##   7  10       11460 3746         3.06
##   8  20       21446 3746         5.73
##   9  32       36948 3746         9.86
##   10 16       16948 3746         4.52
##   11 16       16914 3746         4.52
##   12 14       18440 3746         4.92
```

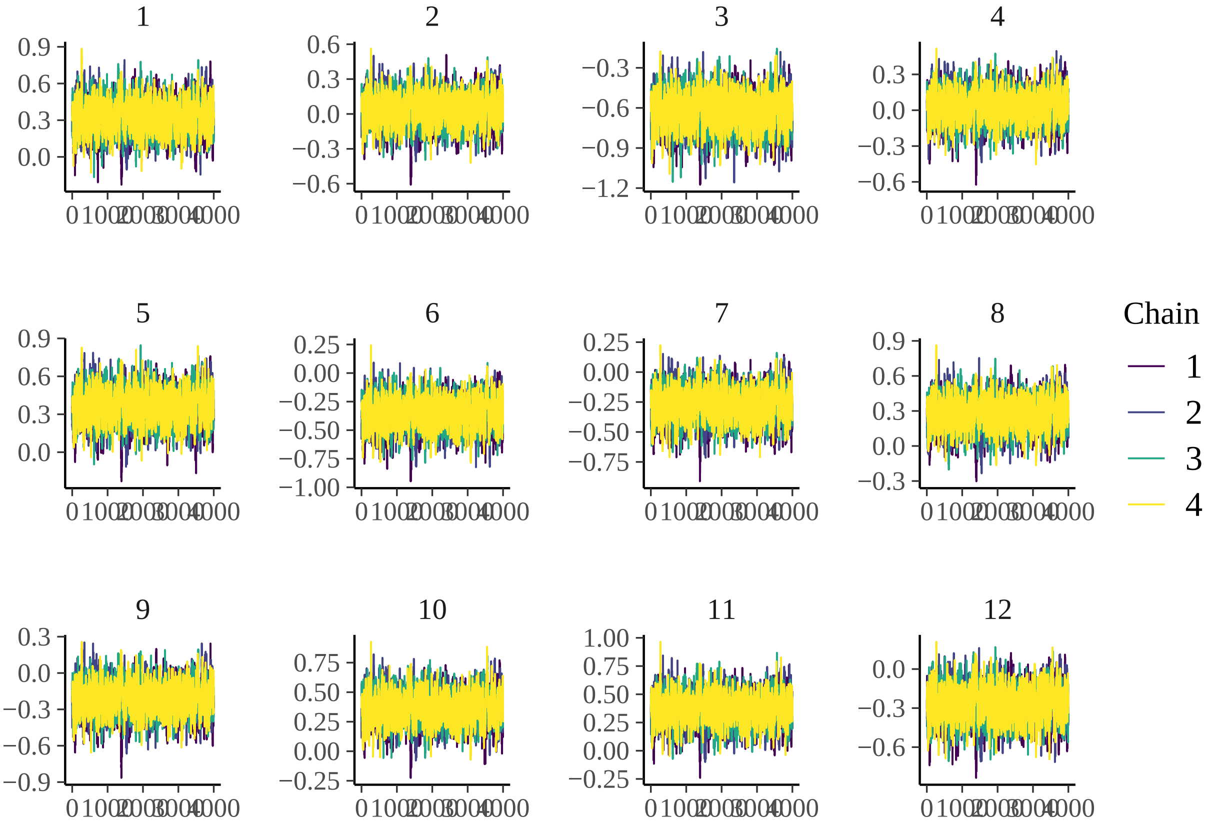
4.2 Diagnosis with bayesplot package

We will proceed only with the chain for genotypes.

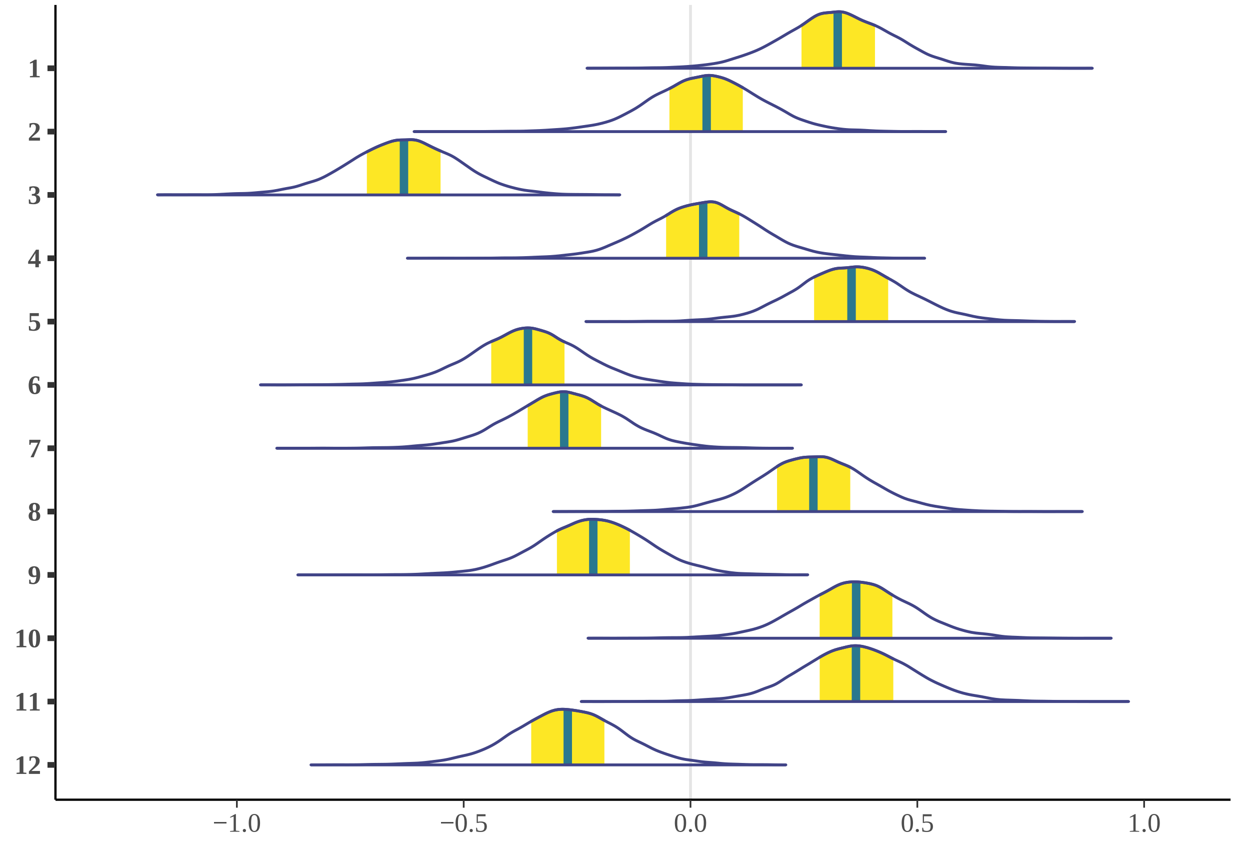
For more examples you can see: bayesplot

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

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

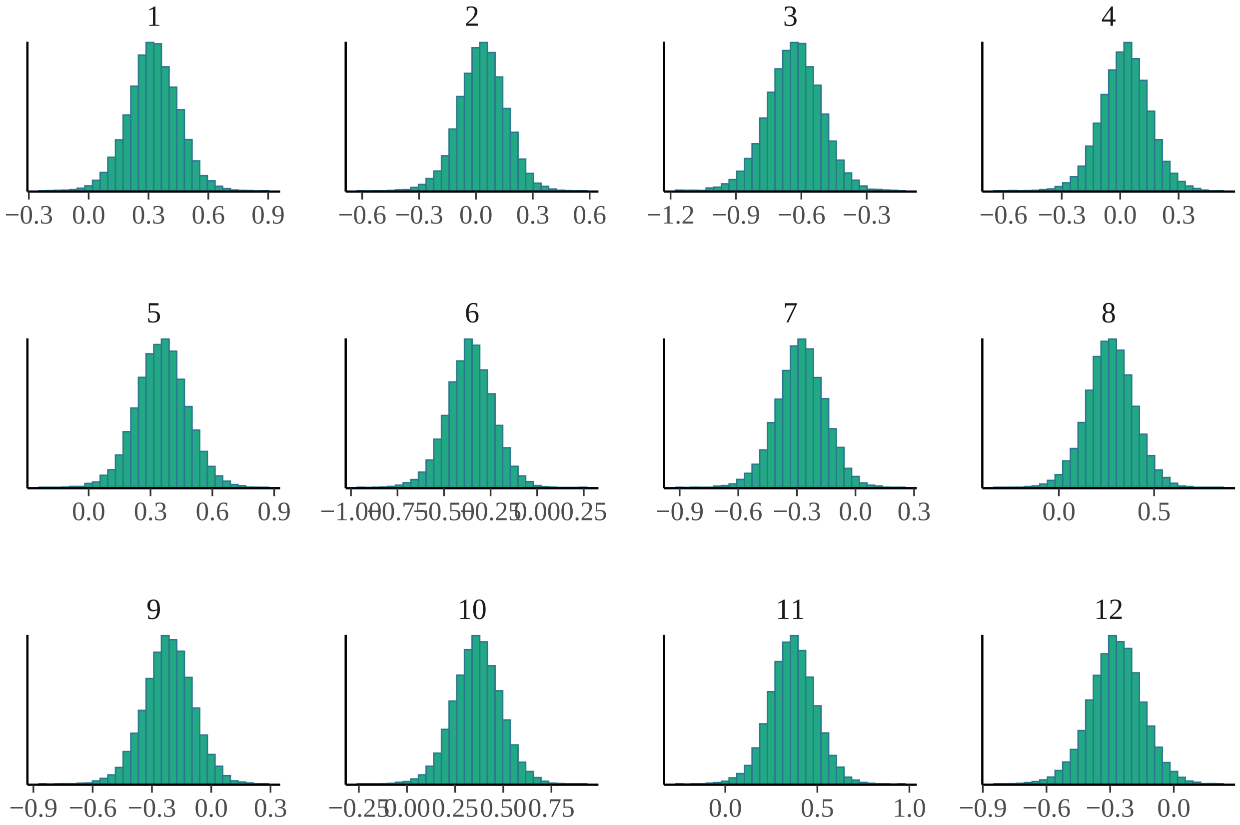


```
mcmc_areas(diag.gen)
```



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

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



5 Prediction

Extract the predict values.

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

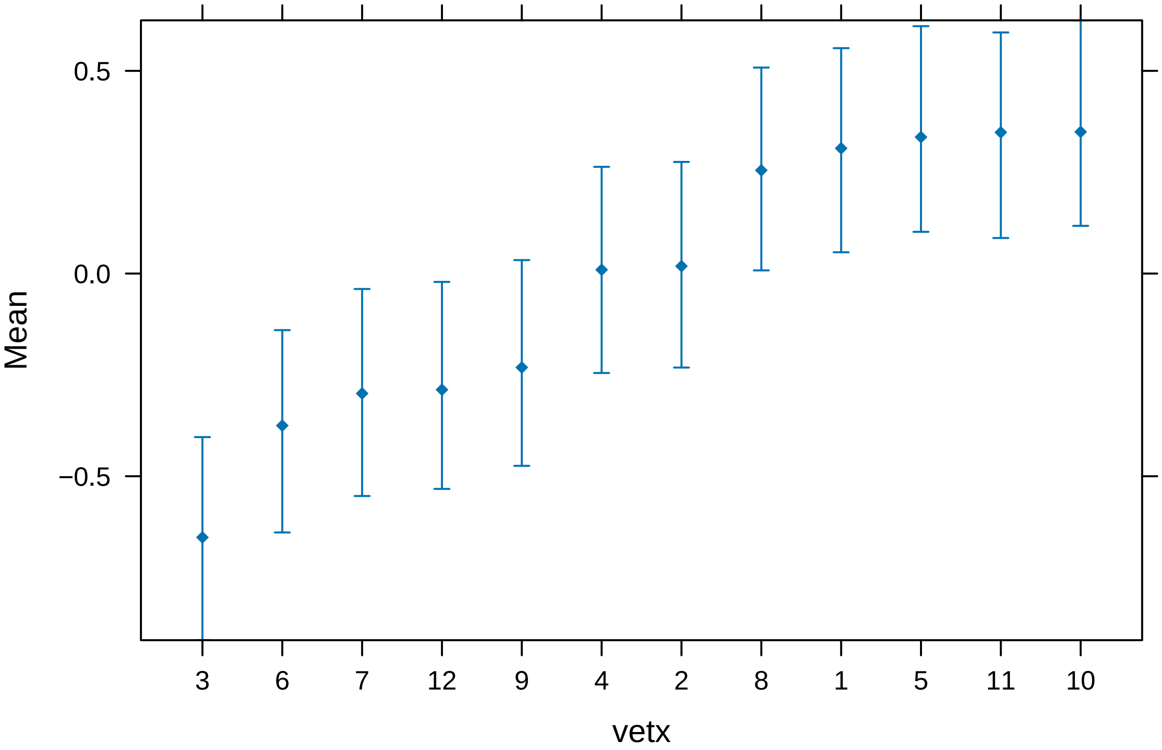
```
##       Mean    Median      2.5%     97.5%
## 1 2.030495 2.0284991 1.7954455 2.2381756
## 2 1.480747 1.4810897 1.2504086 1.7009272
## 3 0.681462 0.6806372 0.4712874 0.9057387
## 4 1.668655 1.6701669 1.4507153 1.8902522
## 5 2.194171 2.1949558 1.9718773 2.4131871
## 6 1.039253 1.0397560 0.8060134 1.2439083
```

6 Genotype effect

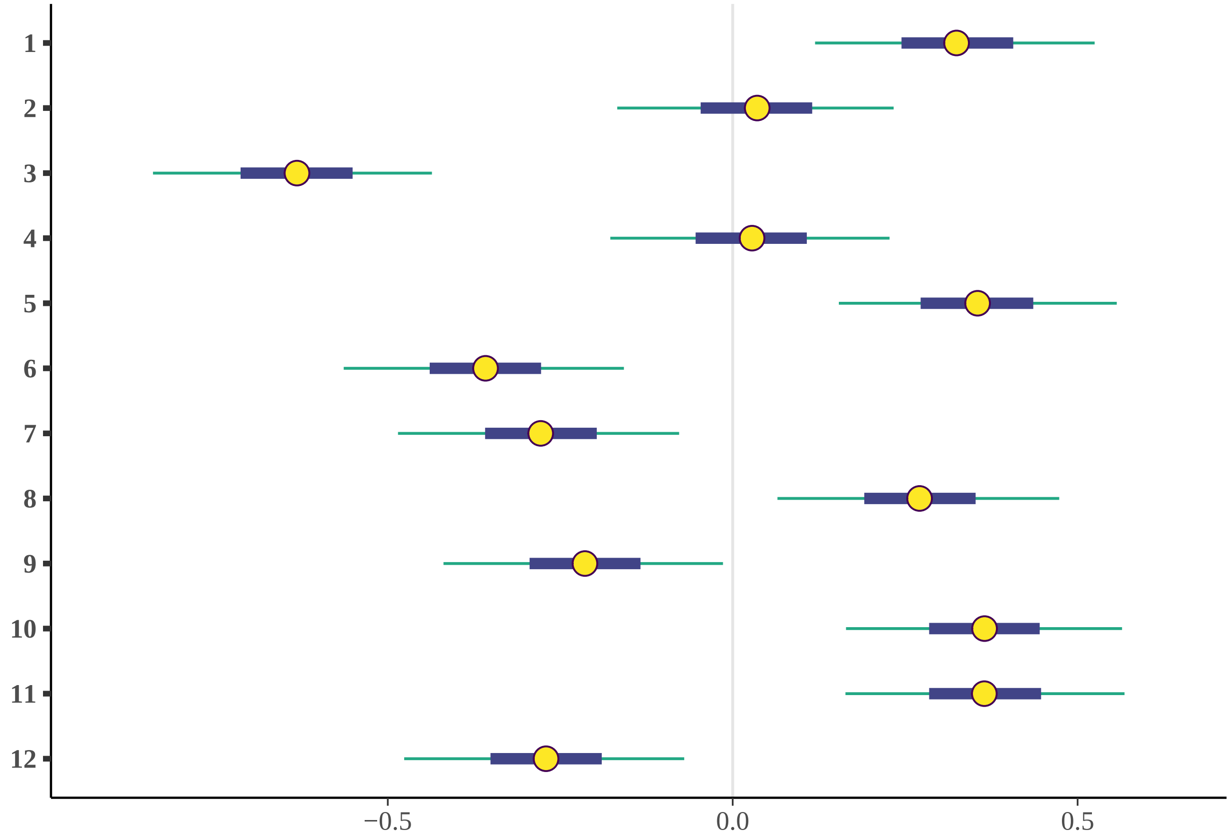
Extract the genotype effects. For more details you can see: gen.effects and ammiBayes.gen.plot.

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

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



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

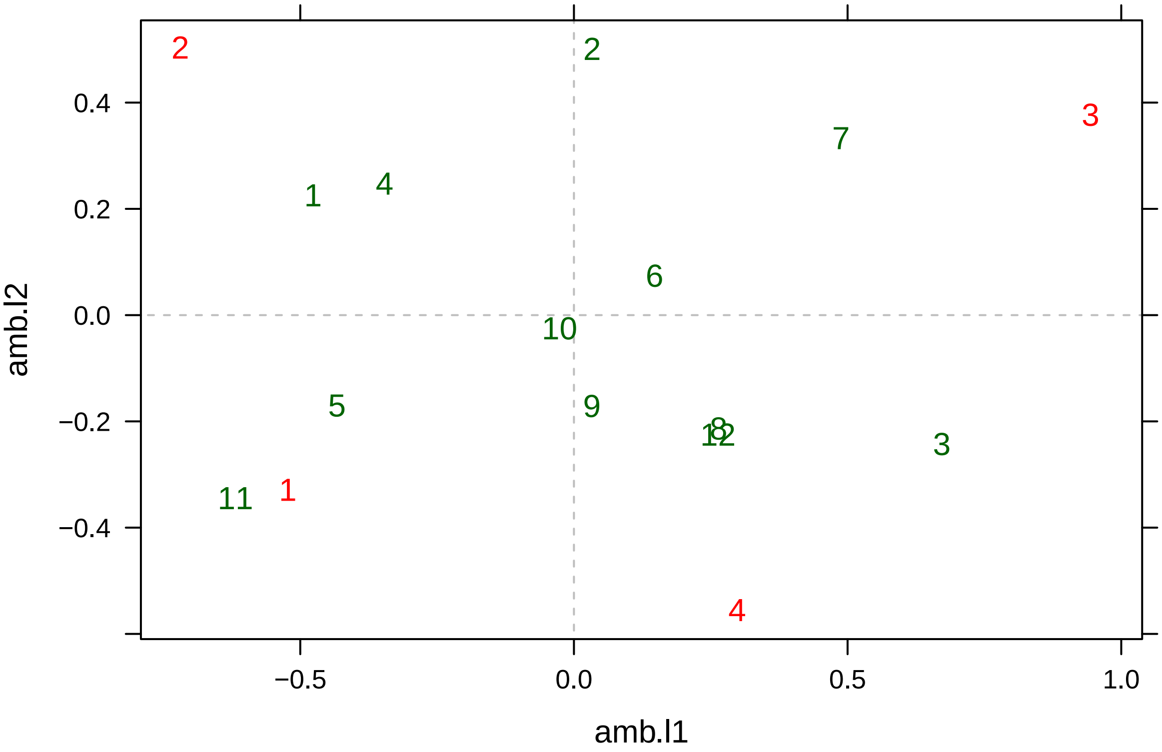


7 Plot functions

7.1 Plot means

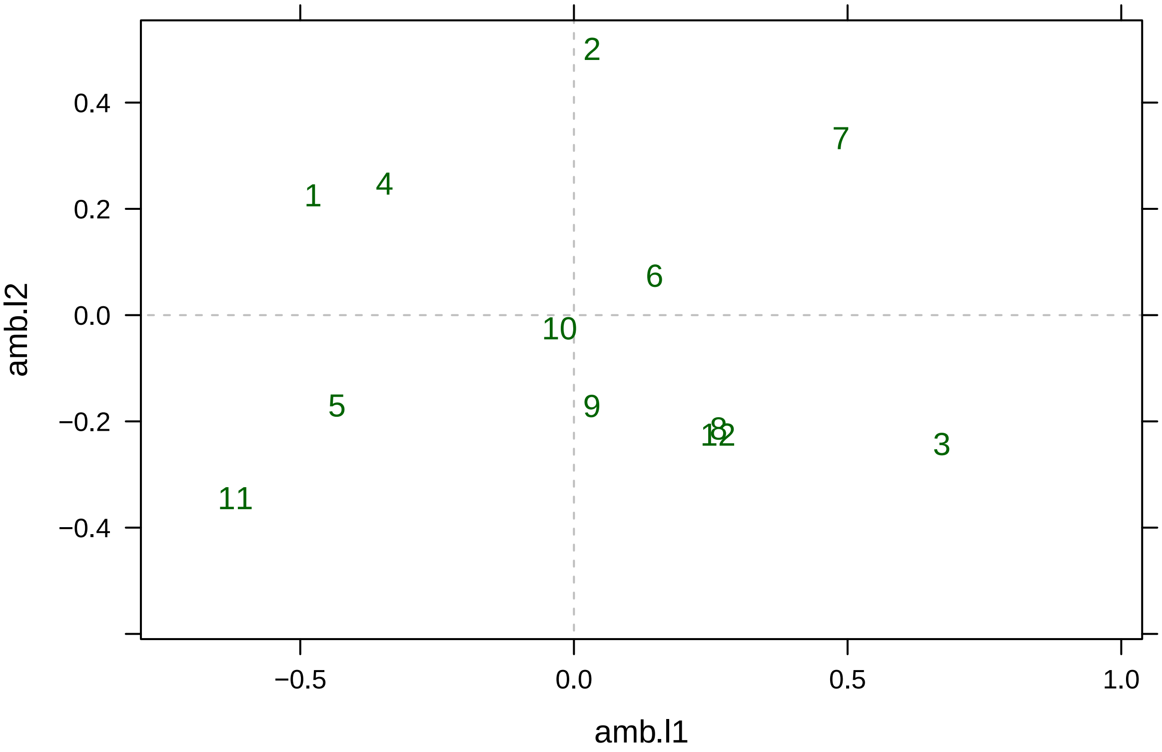
By default all genotypes and environments are plotted. For more details see `help("ammiBayes.mean.plot")`

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



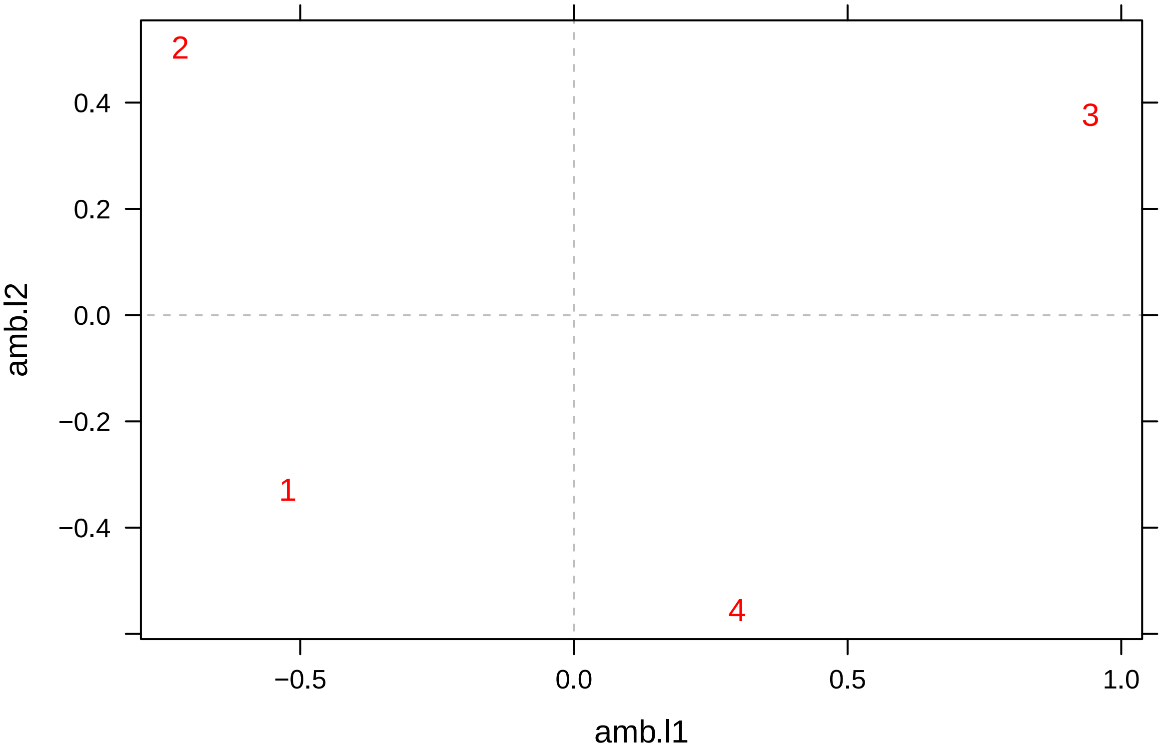
Plotting only the effects of genotypes

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



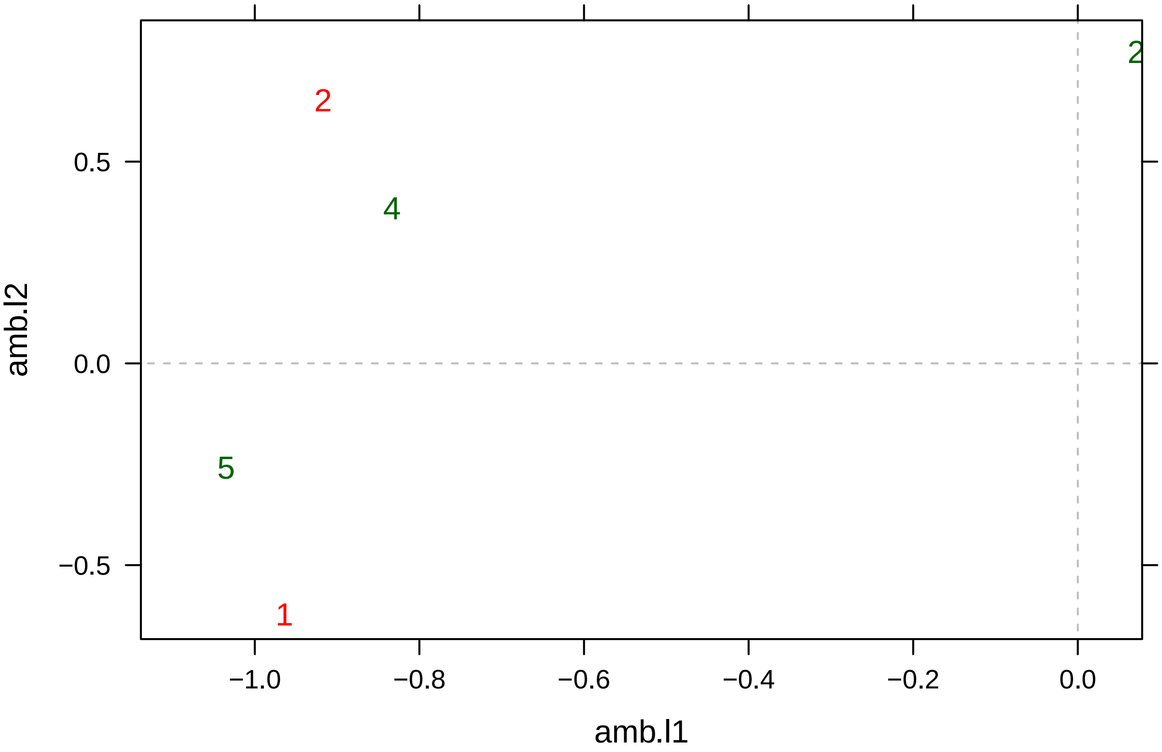
Plotting only the effects of environments

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



Specifying the effects of genotypes and environments

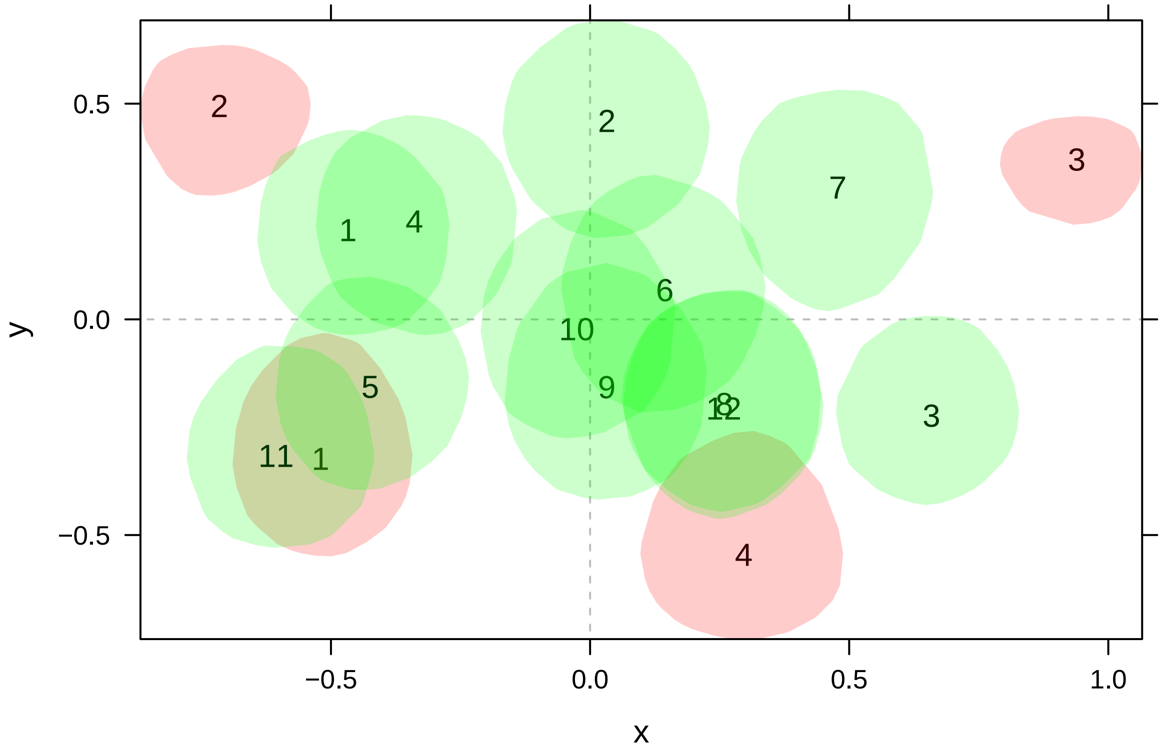
```
ammiBayes.mean.plot(model, pars.env=c("1","2"), pars.gen=c("2","4","5"))
```



7.2 Confidence regions

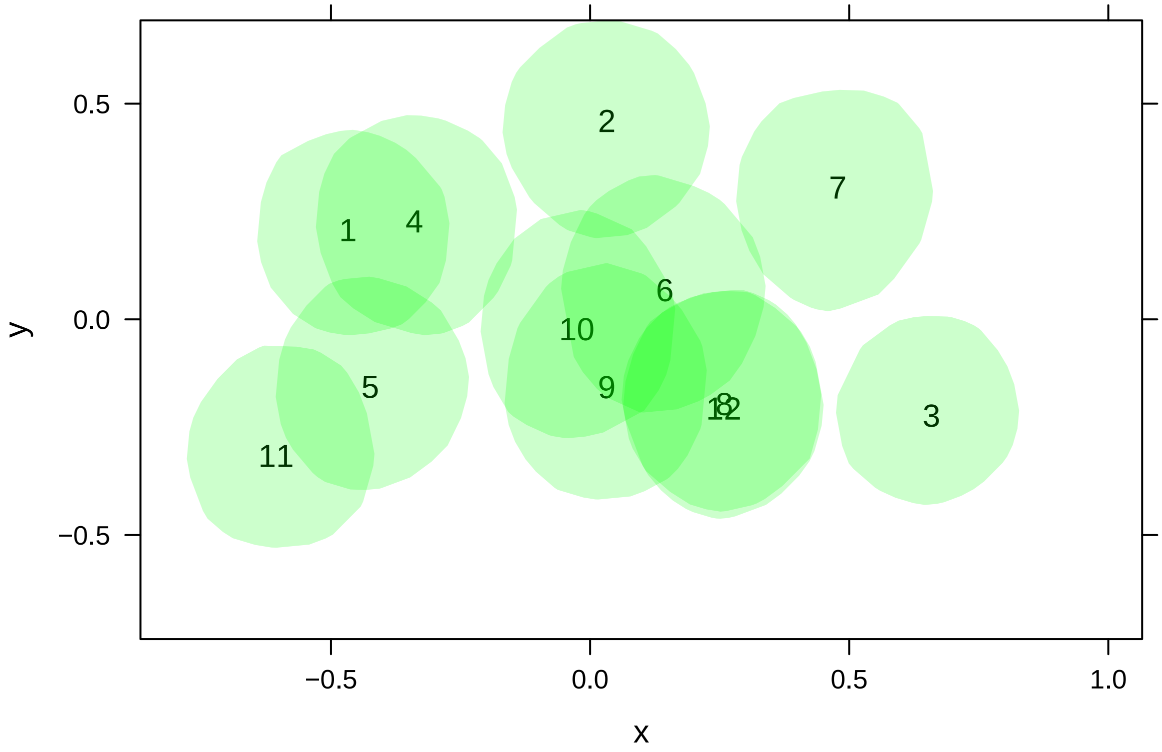
By default, all genotypes and environments are plotted with confidence regions of the 95%. For more details see: ammiBayes.conf.plot.

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



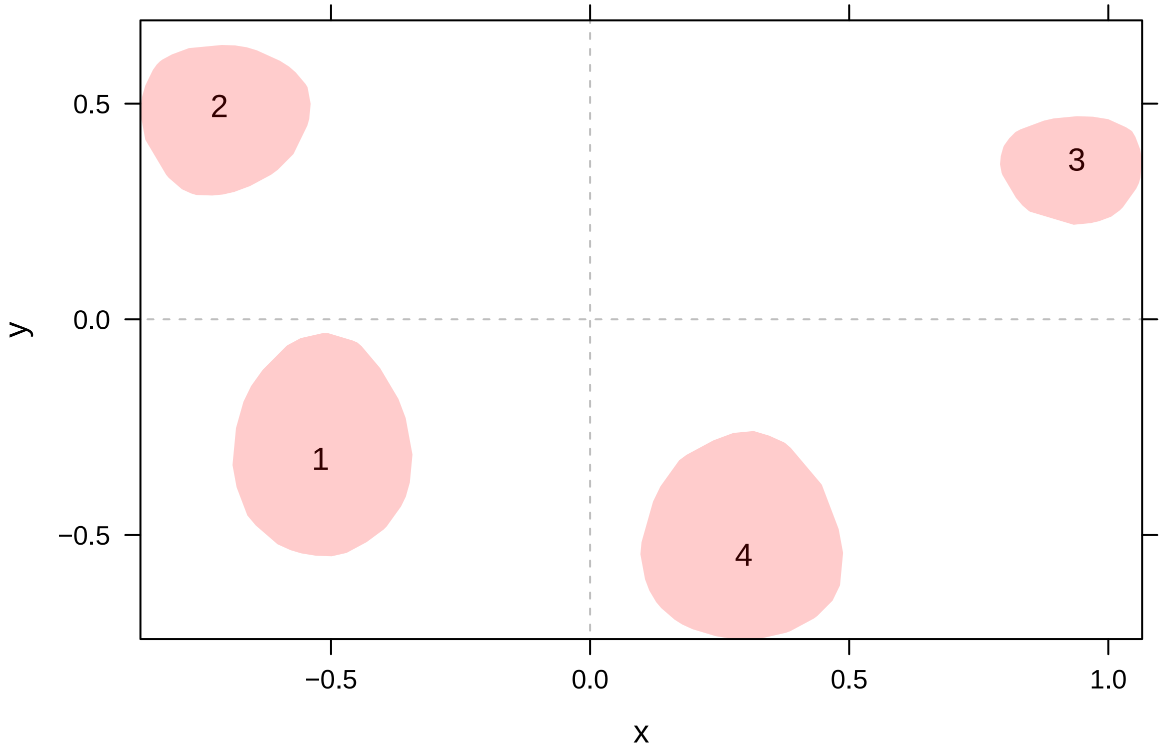
Plotting only the effects of genotypes

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



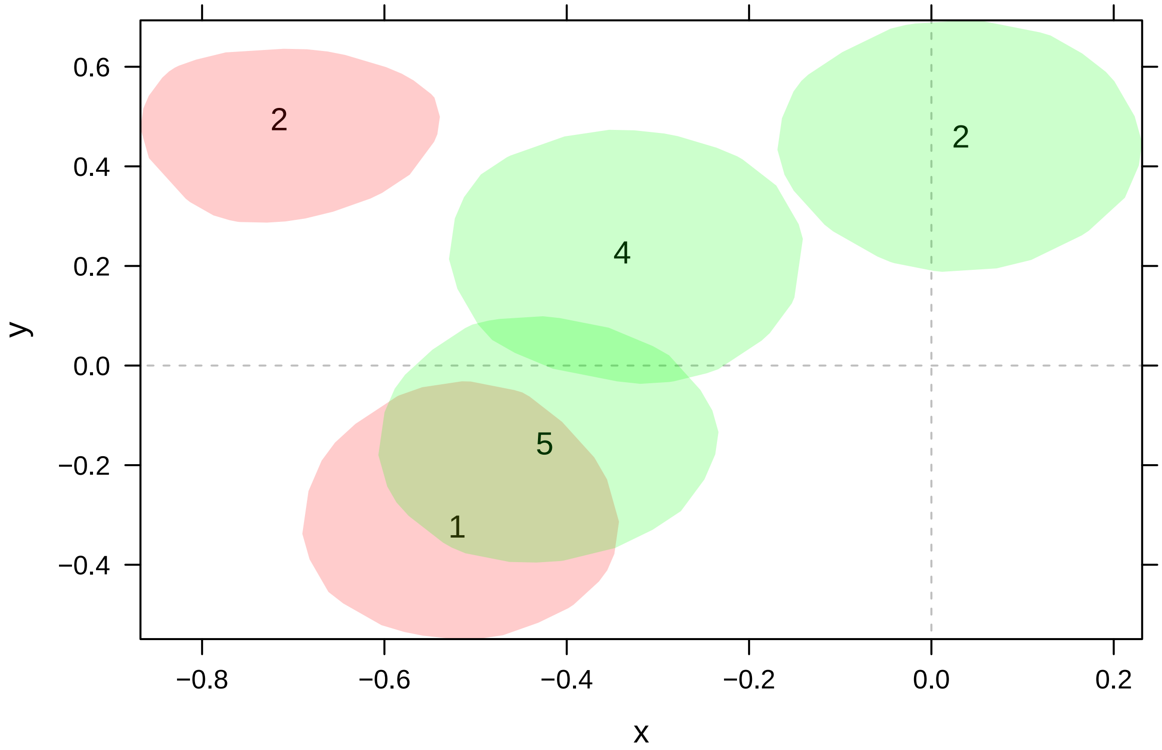
Plotting only the effects of environments

```
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"))
```



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
2
2
4
5
1