

Bayesian AMMI models for continuous data with or without additive and dominance effects

Fabio Mathias Correa* Larisa Carvalho Vilas Boas† Luciano A. Oliveira‡
Carlos Pereira da Silva§ Joel Jorge Nuvunga¶ Julio Silvio de Sousa Bueno Filho||

02 March, 2026

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 1 |
| 2 | Setup | 1 |
| 3 | Example | 2 |
| 4 | Diagnosis | 3 |
| 4.1 | Diagnosis with coda package | 3 |
| 4.2 | Diagnosis with bayesplot package | 5 |
| 5 | Prediction | 7 |
| 6 | Genotype effect | 7 |
| 7 | Plot functions | 9 |
| 7.1 | Plot means | 9 |
| 7.2 | Detection GxE interaction | 11 |
| 7.2.1 | By Genotype | 11 |
| 7.3 | Confidence regions | 11 |

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**

*University of the Free State, mathiascorreaf@ufs.ac.za, Maintainer

†Universidade Federal de Lavras

‡Universidade Federal da Grande Dourados

§Universidade Federal de Lavras

¶Eduardo Mondelane University

||Universidade Federal de Lavras

```
library(bayesplot)
library(ammBayes)
library(coda)
```

3 Example

```
library(ammBayes)

data(ammData) # see help("ammData")

head(ammData)
```

```
##   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(ammData$amb)
Rep <- factor(ammData$rep)
Gen <- factor(ammData$gen)
Y <- ammData$prod
```

We are using default priors. To change the priors see: `help("ammBayes")`

```
model <- ammBayes(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   0.1473 0.0791  0.0435   0.2948
## Var.error 0.0359 0.0058  0.0258   0.0476
##
## Lambda
##           Mean St.dev LI(2.5%) LS(97.5%)
## L.1 1.7712 0.1098   1.5621   1.9891
## L.2 0.8151 0.1161   0.5871   1.0468
## L.3 0.4034 0.1305   0.1392   0.6575
##
##           L.1      L.2      L.3
## Accumulated.lambda 1.771223 2.586291 2.989713
##
## Explained
##           D.1      D.2      D.3
## Percentage    0.7913673 0.1675791 0.04105362
```

```
## Perc.acumulated 0.7913673 0.9589464 1.00000000
##
## Elapsed time: 3.466 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.ammBayes")`.

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

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

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

For each object extracted with the `diagnosis.ammBayes` 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      1.04
## 2      1.01      1.04
## 3      1.01      1.04
## 4      1.01      1.04
## 5      1.01      1.03
## 6      1.01      1.03
## 7      1.01      1.04
## 8      1.01      1.03
## 9      1.01      1.03
## 10     1.01      1.04
## 11     1.01      1.03
## 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  14      16156 3746      4.31
##  2  10      10588 3746      2.83
##  3  16      16892 3746      4.51
##  4  12      14762 3746      3.94
##  5  16      16914 3746      4.52
##  6  15      16680 3746      4.45
##  7  12      14907 3746      3.98
##  8  28      24528 3746      6.55
##  9  16      20910 3746      5.58
## 10  8       8238  3746      2.20
## 11 15      17115 3746      4.57
## 12 15      17115 3746      4.57
##
##
## [[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  20      19810 3746      5.29
##  2  21      24804 3746      6.62
##  3  27      26709 3746      7.13
##  4  18      23379 3746      6.24
##  5  14      15684 3746      4.19
##  6  14      15306 3746      4.09
##  7  16      18564 3746      4.96
##  8  30      29193 3746      7.79
##  9  21      26481 3746      7.07
## 10  14      12986 3746      3.47
## 11  21      23076 3746      6.16
## 12  21      22953 3746      6.13
##
##
## [[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      16084 3746      4.29
##  2  10      12176 3746      3.25
##  3  15      13674 3746      3.65
##  4  12      13952 3746      3.72
##  5  16      16092 3746      4.30
##  6  12      12442 3746      3.32
##  7  12      14420 3746      3.85
##  8  18      18201 3746      4.86
##  9  18      20544 3746      5.48

```

```
## 10 16      15300 3746      4.08
## 11 16      17672 3746      4.72
## 12 16      17758 3746      4.74
##
##
## [[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  24      22347 3746      5.97
## 2  24      26276 3746      7.01
## 3  18      20604 3746      5.50
## 4  14      15010 3746      4.01
## 5  15      19977 3746      5.33
## 6  12      15102 3746      4.03
## 7  12      12180 3746      3.25
## 8  24      25839 3746      6.90
## 9  18      19446 3746      5.19
## 10 18      16954 3746      4.53
## 11 12      12736 3746      3.40
## 12 24      25839 3746      6.90
```

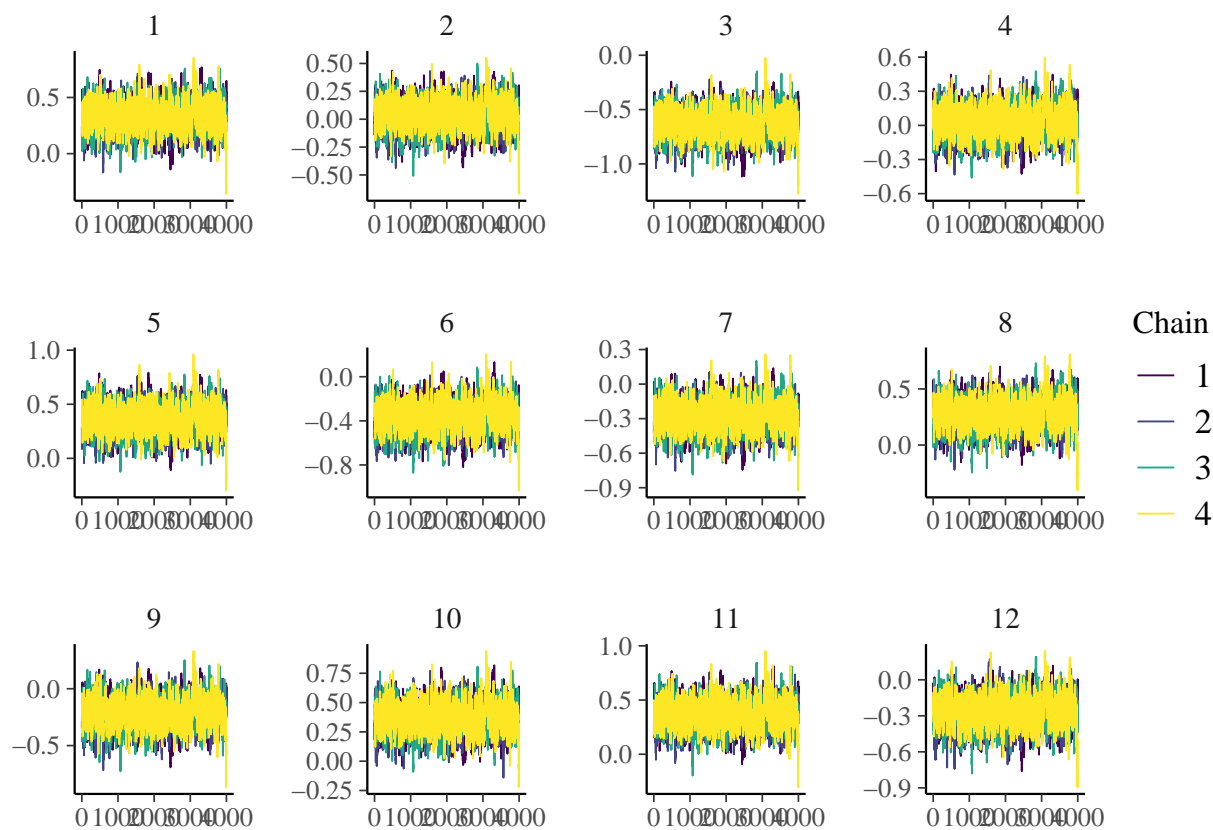
4.2 Diagnosis with bayesplot package

We will proceed only with the chain for genotypes.

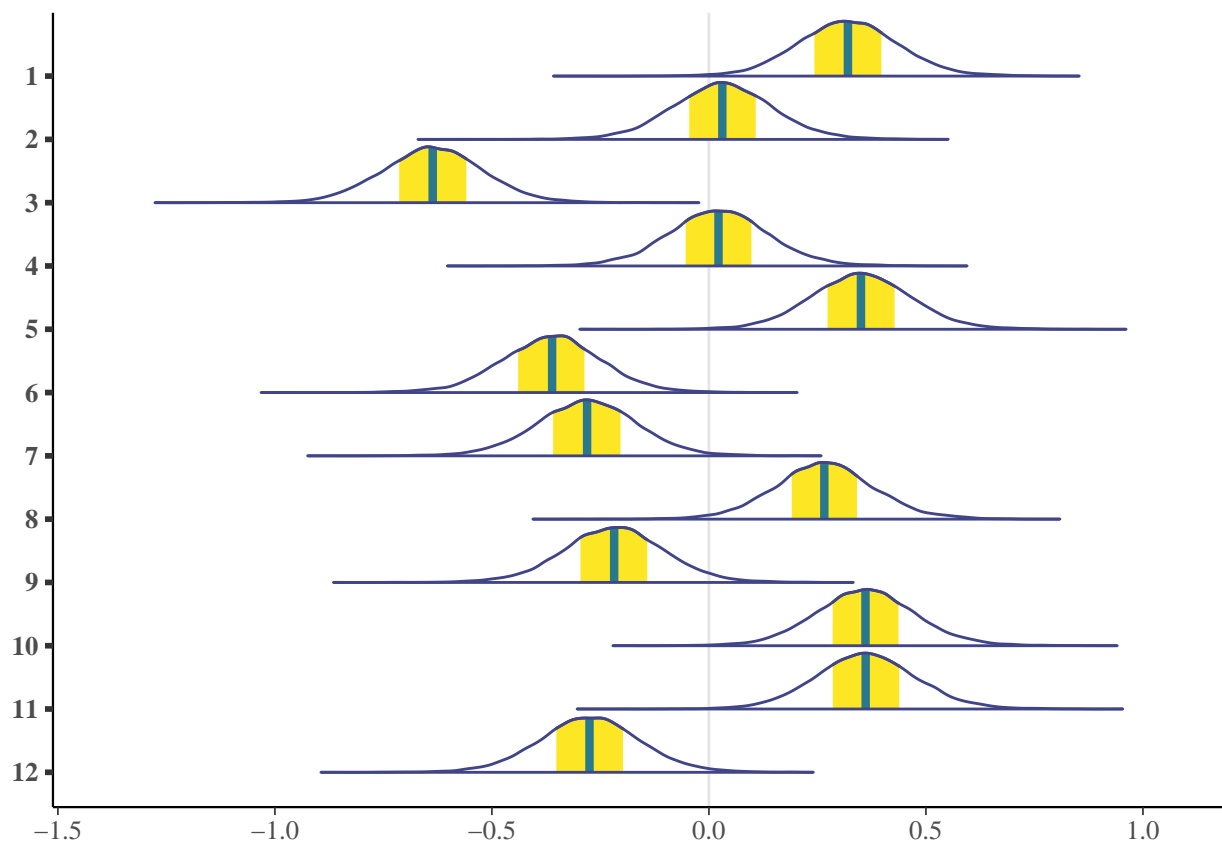
For more examples you can see: bayesplot

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

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

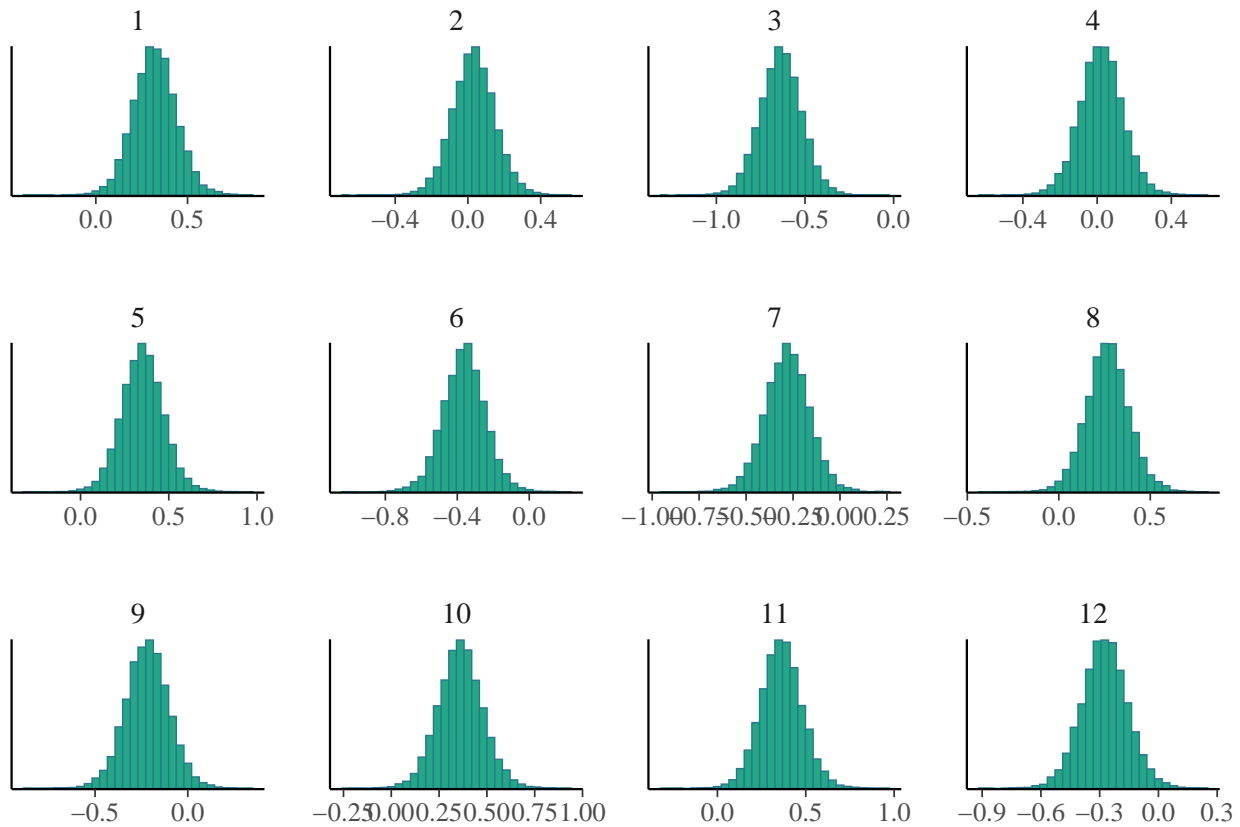


```
mcmc_areas(diag.gen)
```



```
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)
```

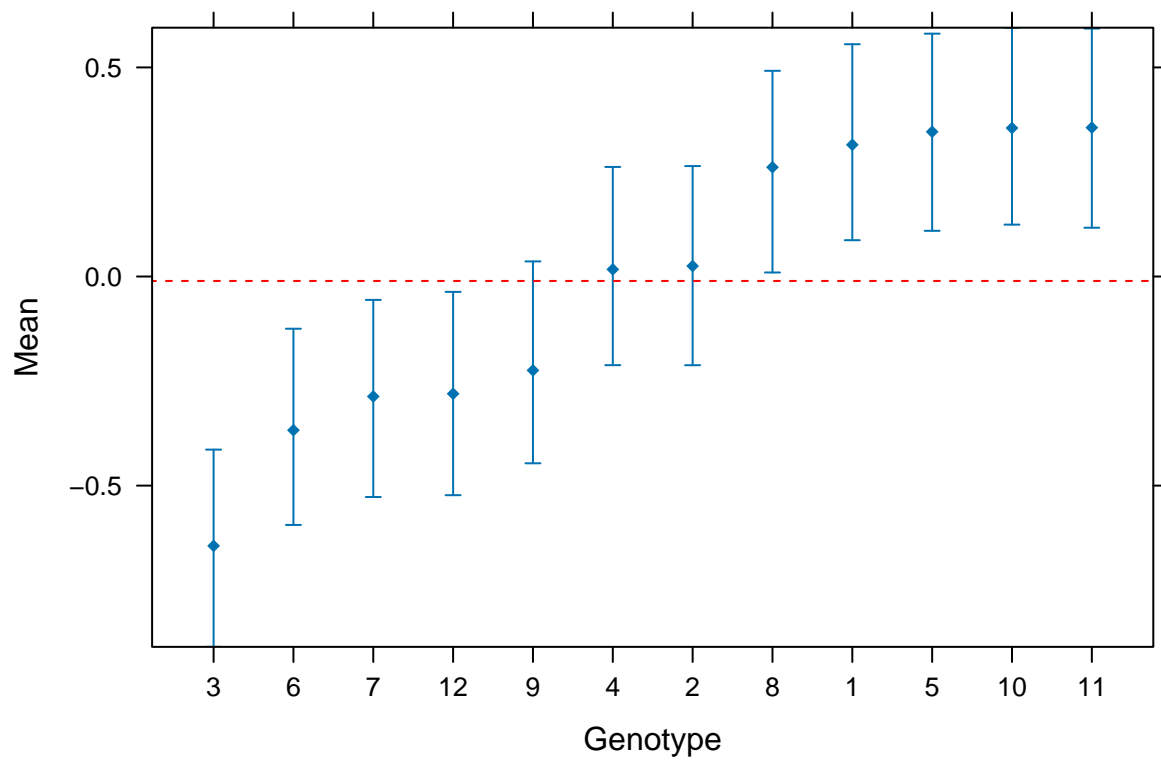
```
##      Mean   Median    2.5%    97.5%
## 1 2.0299717 2.030881 1.7964020 2.2345585
## 2 1.4746132 1.475665 1.2430896 1.7020596
## 3 0.6748327 0.673234 0.4729345 0.9049809
## 4 1.6670276 1.667619 1.4596453 1.8954583
## 5 2.1952179 2.196309 1.9828880 2.4050739
## 6 1.0420587 1.042044 0.8267854 1.2647062
```

6 Genotype effect

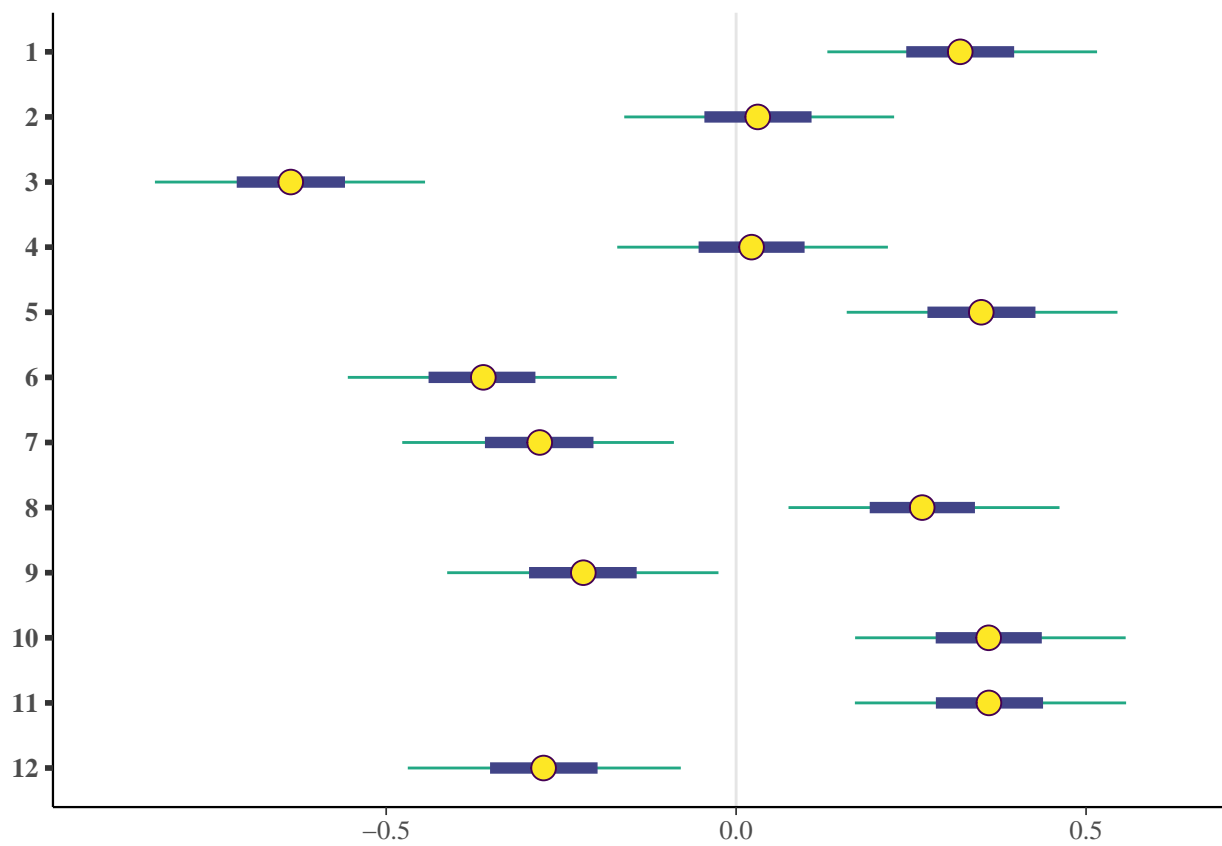
Extract the genotype effects. For more details you can see: [gen.effects](#) and [ammiBayes.gen.plot](#).

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

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



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

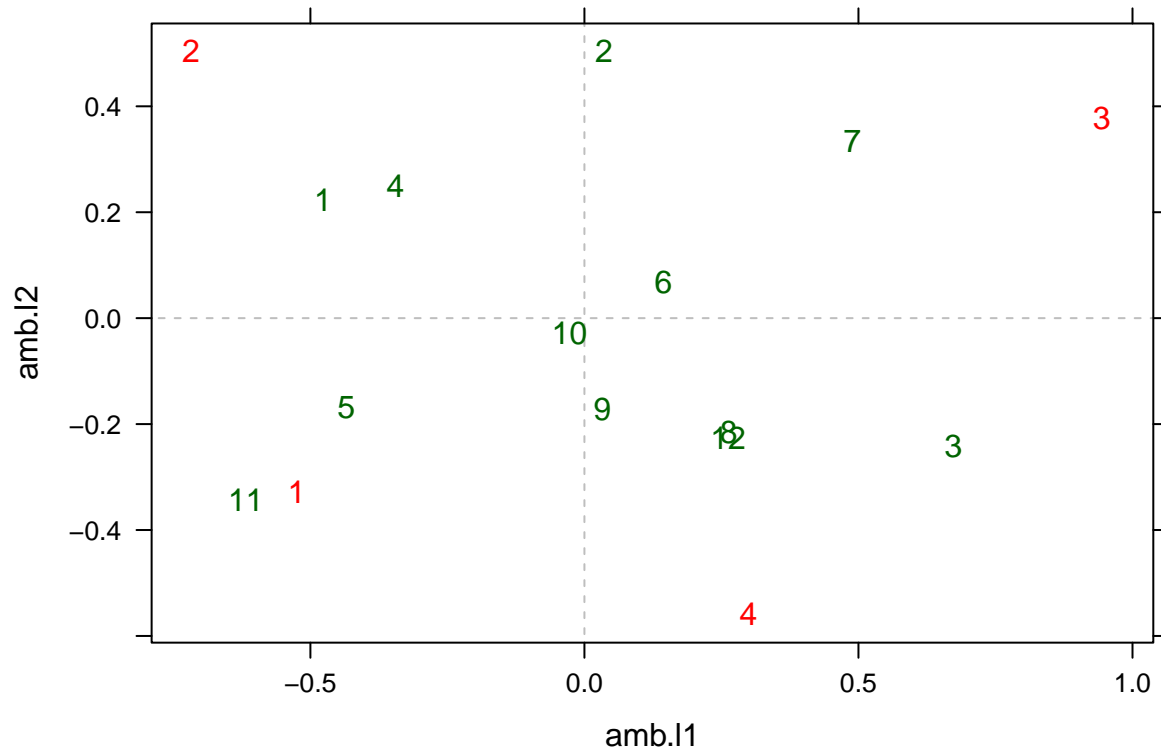


7 Plot functions

7.1 Plot means

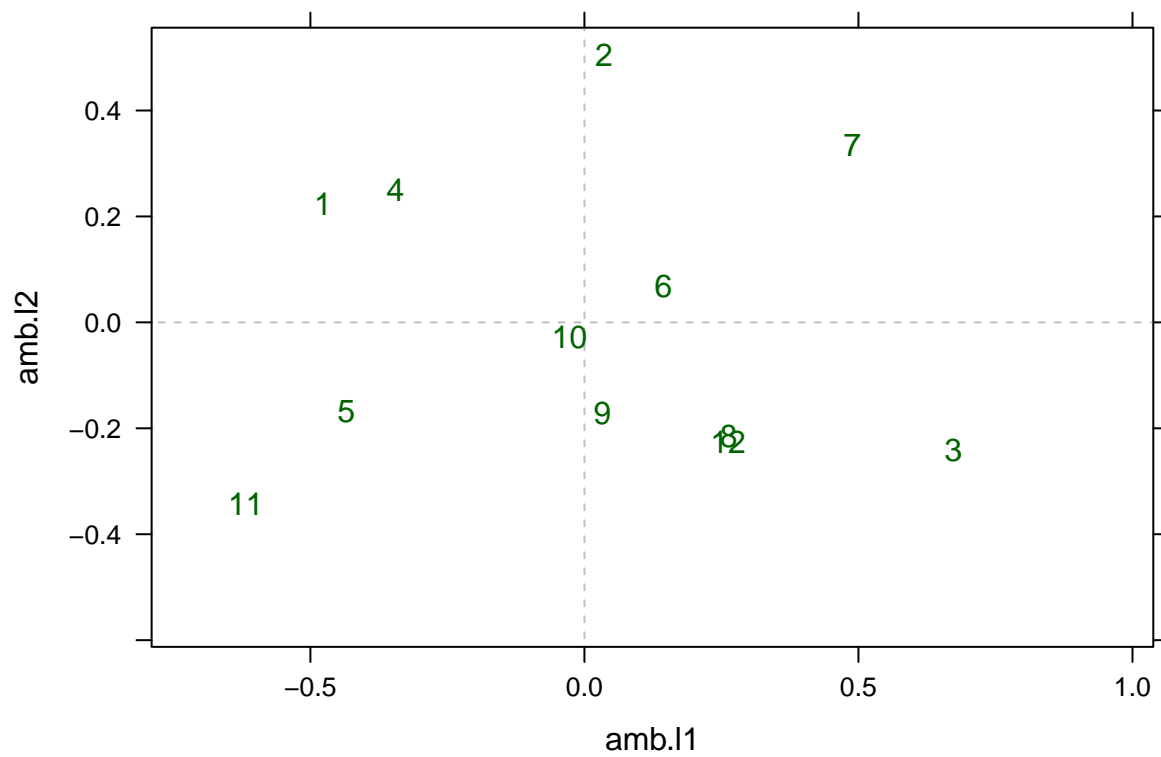
By default all genotypes and environments are plotted. For more details see `help("ammiBayes.mean.plot")`

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



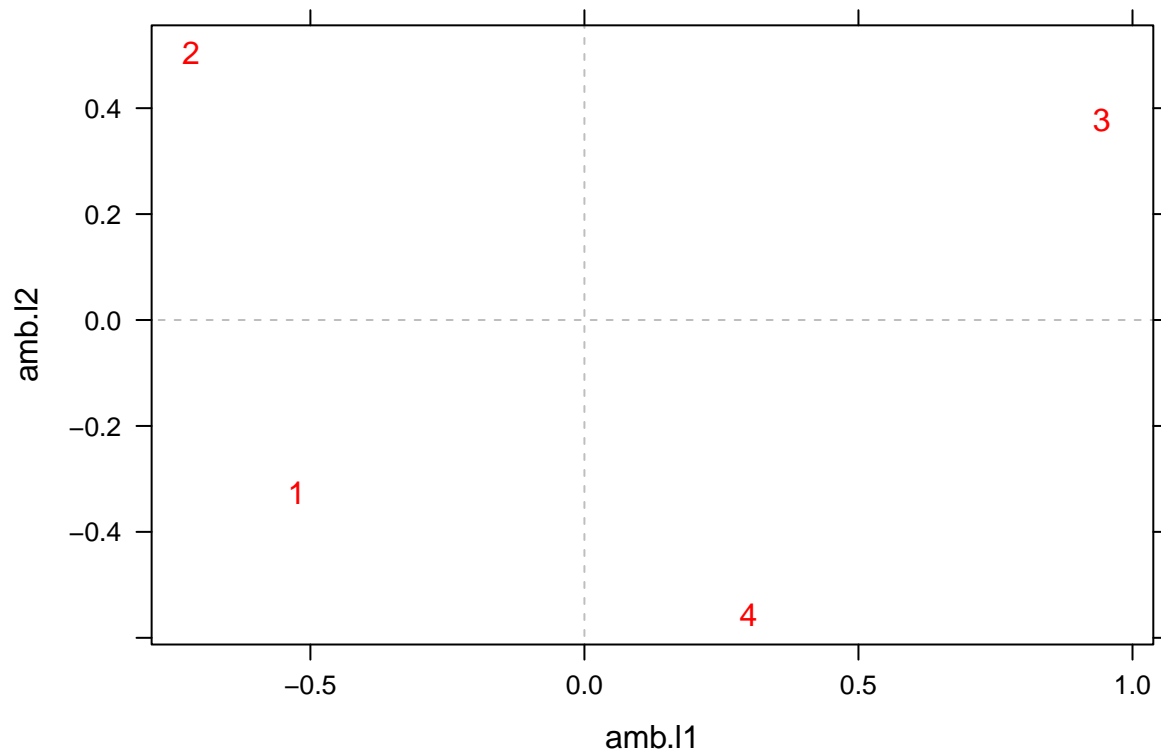
Plotting only the effects of genotypes

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



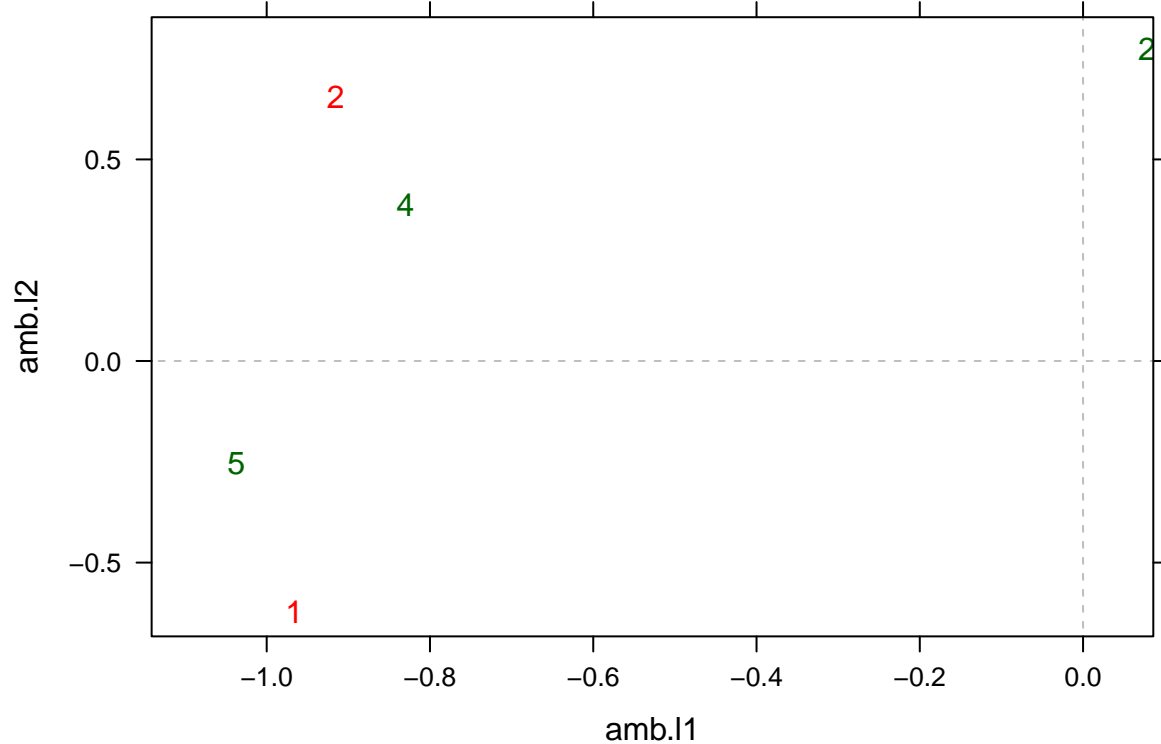
Plotting only the effects of environments

```
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 Detection GxE interaction

7.2.1 By Genotype

```
detect.interaction(model)
```

```
##      |
## [1] "1"  "11" "12" "2"  "3"  "4"  "5"  "7"  "8"
## attr(,"class")
## [1] "ammiBayes"
$$$ By Environment
```

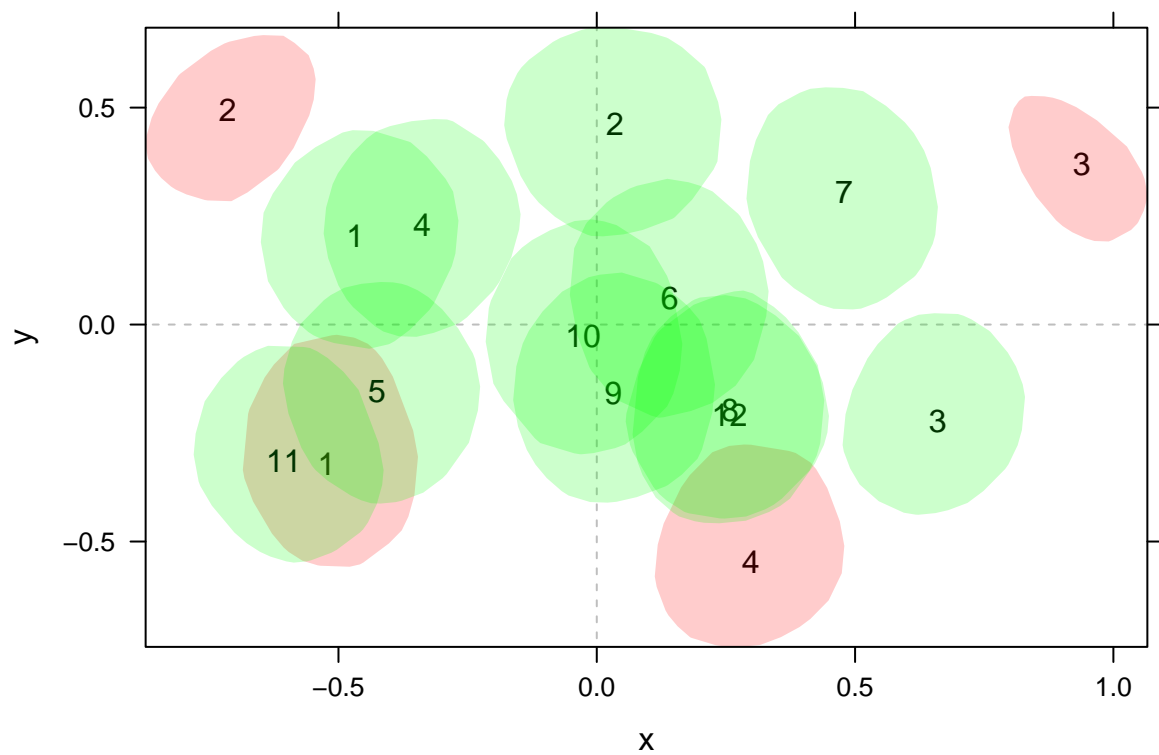
```
detect.interaction(model, type="env")
```

```
##      |
## [1] "1"  "2"  "3"  "4"
## attr(,"class")
## [1] "ammiBayes"
```

7.3 Confidence regions

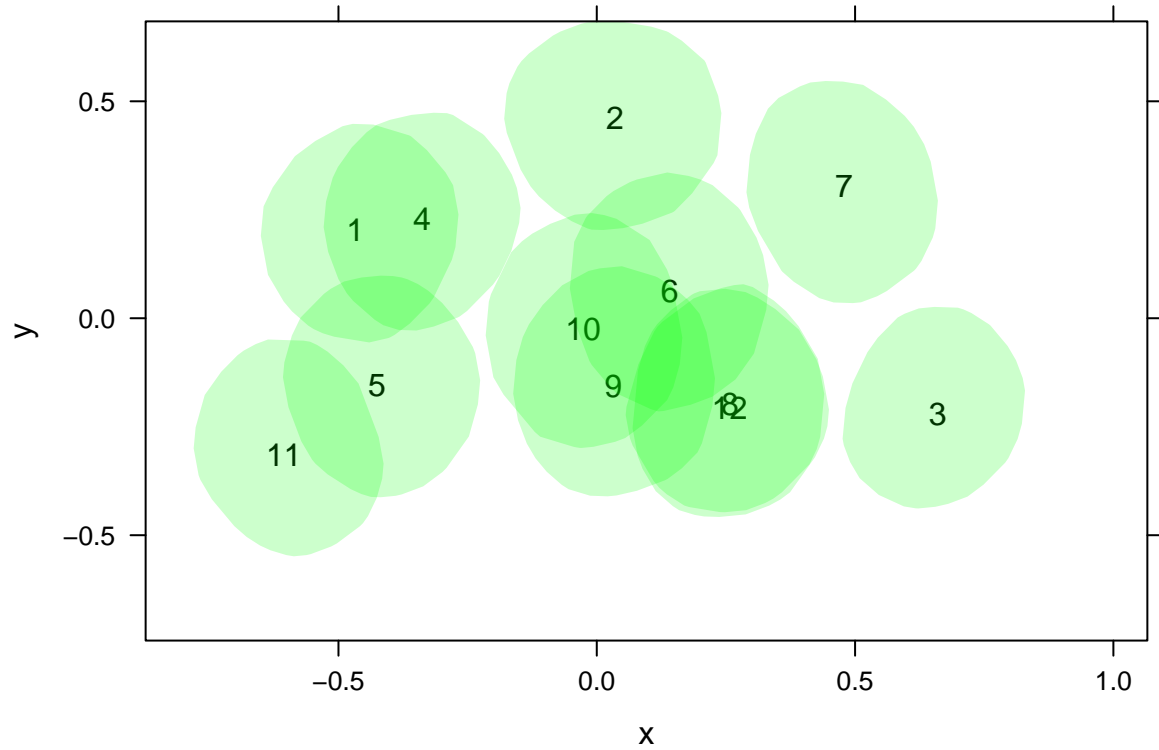
By default, all genotypes and environments are plotted with confidence regions of the 95%. For more details see: [ammiBayes.conf.plot](#).

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



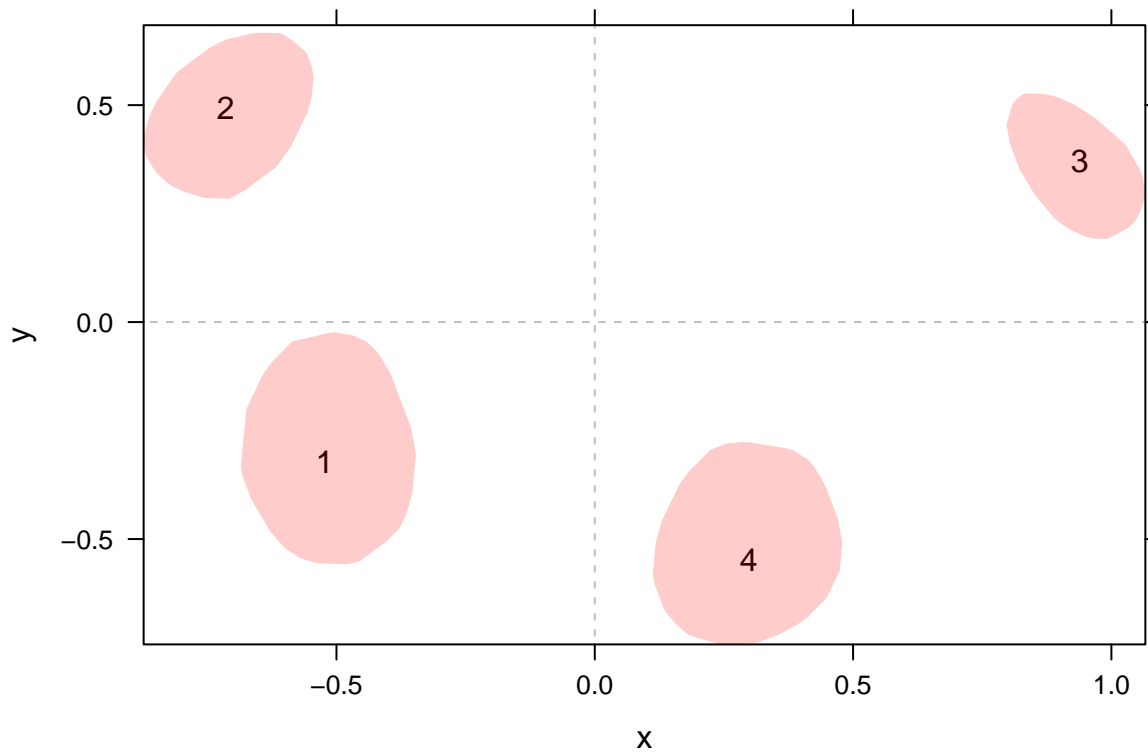
Plotting only the effects of genotypes

```
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"))
```

