The built-in application

Muhammad Yaseen

2026-11-04

Table of contents

1 Abstract **1**

2 Starting the application **1**

3 The pages **2**

4 Reproducibility **4**

5 Deploying **5**

6 References **5**

1 Abstract

`bam_app()` opens an application in the web browser that runs every analysis of the package without code. This vignette describes its pages and how it keeps its results reproducible.

2 Starting the application

```
bam_app(
    launch.browser = TRUE
  )
```

The application needs the suggested packages **shiny**, **bslib** and **DT**. **magick** with WebP support is optional: with it the plots are sent to the browser as lossless WebP images, about half the size of the same PNG.

```
isTRUE(tryCatch(requireNamespace("magick", quietly = TRUE) &&
 ↪  isTRUE(magick::magick_config()$webp),
                error = function(e) FALSE))
```

```
[1] TRUE
```

`TRUE` means WebP is available. A magick installation that no longer loads, because the ImageMagick libraries of the system were removed or upgraded, gives `FALSE`, and the application then sends PNG images; reinstalling magick restores WebP.

3 The pages

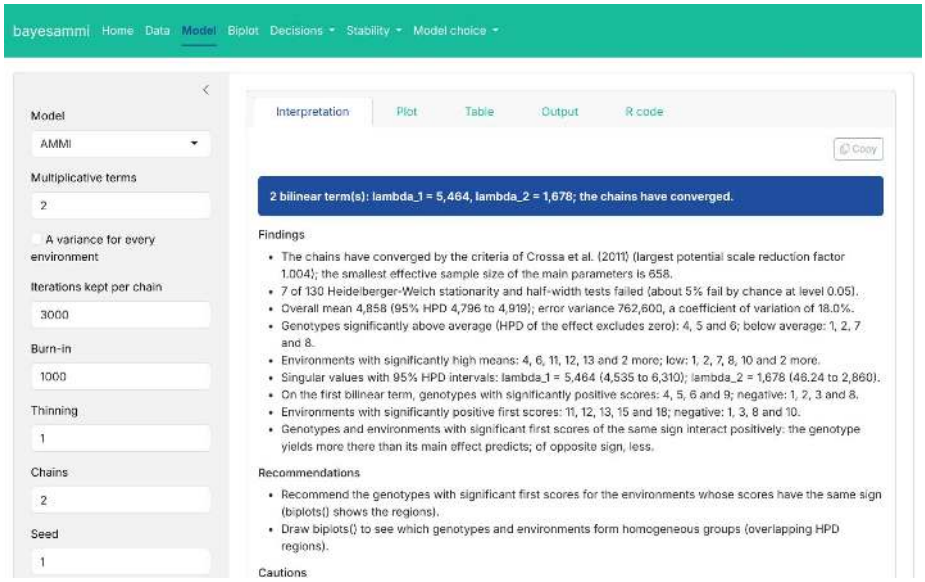
Page	What it does
Home	what the application does and the order of the pages
Data	the `Maiz` example or a CSV file with one row per plot; choice of the response, genotype, environment and replicate columns
Model	AMMI or GGE, number of terms, common or environment-specific variances, iterations, burn-in, thinning, chains, seed and priors; then the summary, traces and densities
Biplot	the biplot with HPD regions, for genotypes, environments or both
Decisions	winners, probabilities of superior performance, ranks, stability and predictions
Stability	Bayesian Eberhart-Russell, Finlay-Wilkinson and Shukla models; classical and AMMI-based stability indices
Model choice	posterior predictive checks and a comparison of the fits of the session

Every result page has the same tabs:

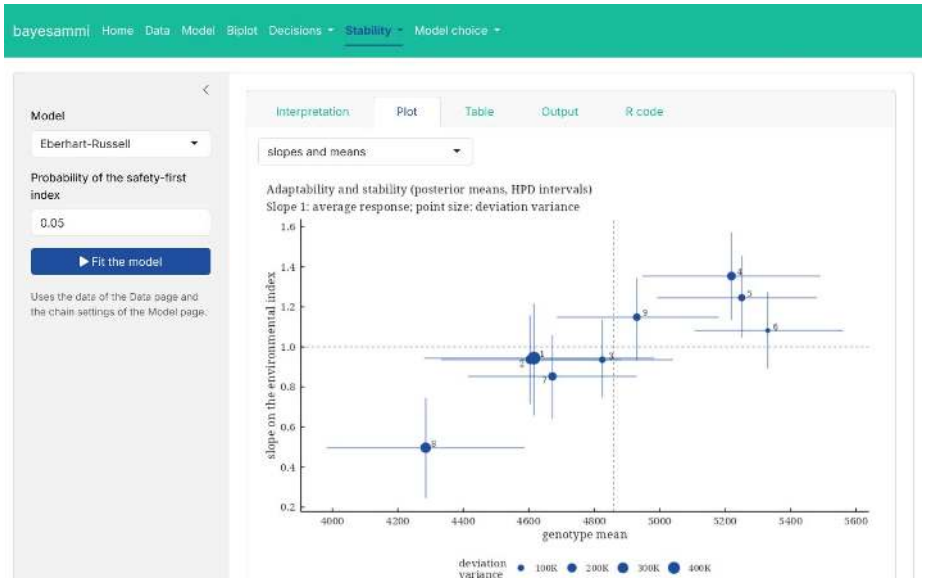
- **Interpretation** gives the plain-language reading of `bam_interpret()`, with a button that copies it, and the sources of the method.
- **Plot** shows the figure, which can be saved as PNG, PDF or WebP.
- **Table** shows the numbers, rounded, in a sortable table with buttons to copy them or save them as CSV or Excel.
- **Output** shows the printed result as text, with a button that copies it.

The model page has a further tab, **R code**, with the code that reproduces the fit outside the application.

The model page after a fit, with the interpretation tab open:



The plot tab of the Bayesian Eberhart-Russell model on the Stability menu:



The screenshots are lossless WebP in the HTML version: 62 KB for the first, against 147 KB for the lossy JPEG of the PDF version.

4 Reproducibility

For the default settings with environment-specific variances, the R code tab shows:

```
settings <-
  list(
      .y      = "y"
    , .gen    = "entry"
    , .env    = "site"
    , .rep    = "rep"
    , .nIter  = 3000
    , .k      = 2
    , .burnin = 500
    , .thin   = 1
    , .chains = 2
    , .priors = bam_priors(
          mu_mean     = 0
        , mu_var      = 1e+8
        , tau_mean    = 0
        , tau_var     = 1e+8
        , delta_mean  = 0
        , delta_var   = 1e+8
        , lambda_mean = 0
        , lambda_var  = 1e+8
        , nu          = 1
        , s2          = NULL
        )
    , .level  = 0.95
    , .seed   = 1
    , .model  = "AMMI"
    , .het    = TRUE
    )
cat(bayesammi:::.bam_app_code(a = settings, source = "maiz"))
```

```
library(bayesammi)
data(Maiz)
fit <-
  bayes_ammi(
      .data   = Maiz
    , .y      = y
    , .gen    = entry
    , .env    = site
    , .rep    = rep
    , .nIter  = 3000
    , .k      = 2
    , .burnin = 500
    , .thin   = 1
    , .chains = 2
```

```
    , .priors = bam_priors(lambda_var = 1e+08, nu = 1)
    , .level  = 0.95
    , .seed   = 1
    , .align  = TRUE
    , .model  = "AMMI"
    , .het    = TRUE
    , .waic   = TRUE
    )
fit
bam_interpret(
    x             = fit
  , higher_better = TRUE
  , top           = 5
  )
```

The seed fixes the draws, so the code returns the same numbers as the application.

5 Deploying

`bam_app_object()` returns the application object without running it, for a server:

```
# app.R on a Shiny server
bayesammi::bam_app_object()
```

6 References

Crossa, J., Perez-Elizalde, S., Jarquin, D., Cotes, J. M., Viele, K., Liu, G. and Cornelius, P. L. (2011). Bayesian estimation of the additive main effects and multiplicative interaction model. *Crop Science*, 51(4), 1458–1469. doi:10.2135/cropsci2010.06.0343