Package 'StabilityApp'

November 10, 2026

Type Package

Title Interactive Stability Analysis of Multi-Environment Trials

Version 1.0.0

Date 2026-11-03

Description A 'shiny' application for the stability analysis of multi-environment trials of plant breeding with the package 'stability': analyses of variance, Eberhart and Russell's and Finlay and Wilkinson's regression, Wricke's, Shukla's, Kang's and Fox's statistics, Eskridge's safety-first indices, Annicchiarico's, Lin and Binns' and Huehn's statistics, AMMI analysis
and stability values, and the GGE biplot. Every analysis comes with its tables, plots, the R code that repeats it and an interpretation specific to the method, computed from the results; an optional longer written interpretation can be requested from an external text service with the user's own key.

License GPL-3

URL https://myaseen208.com/StabilityApp/,
https://github.com/myaseen208/StabilityApp

BugReports https://github.com/myaseen208/StabilityApp/issues

Encoding UTF-8

Language en-US

Depends R (>= 4.2)

Imports bslib, commonmark, data.table, DT, ggplot2, htmltools, httr2, shiny, stability (>= 1.0.0), utils

Suggests knitr, pkgload, quarto, R.rsp, svglite, testthat (>= 3.0.0), withr

VignetteBuilder quarto, R.rsp

Config/Needs/website pkgdown (>= 2.1.0)

Config/testthat/edition 3

Config/skill/prefix sa

RoxygenNote 7.3.1

NeedsCompilation no

Author Muhammad Yaseen [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-5923-1714>),
Kent Eskridge [aut, ctb]

Maintainer Muhammad Yaseen <myaseen208@gmail.com>

Repository CRAN

Date/Publication 2026-11-10 09:20:09 UTC

Contents

sa_prompt 2
sa_services 3
StabilityApp 5

Index **7**

`sa_prompt` *The request for a written interpretation of one analysis*

Description

`sa_prompt()` builds the text sent to an external text service (`sa_ask()`) for a written interpretation of one result of the package **stability**. It is specific to the method: it contains how the statistics of that method are read, what to concentrate on, the interpretation computed by `stability::stab_interpret()` (to be kept to), the facts behind it and the first rows of the main table.

Usage

```
sa_prompt(x, higher_better = TRUE, trait = NULL, rows = 15L)
```

Arguments

`x`	a result of a function of **stability** (or the `ggplot` of `stability::gge_biplot()`).
`higher_better`	`TRUE` (default) when larger values of the trait are better.
`trait`	name of the trait (default: the response of the analysis).
`rows`	number of rows of the main table included (default 15).

Details

The text has two parts: `system` (the role and the rules: use only the numbers given, keep to the computed findings, name genotypes and environments as given, say when something cannot be concluded) and `user` (the material and the task: four sections, Statistical interpretation, Key agronomic findings, Recommendations and Next steps, at most about 350 words). Because the computed findings are part of the request, the written text cannot contradict the tests and rankings of the analysis without contradicting its own material, which the rules forbid.

Value

A list with `system` and `user` (character strings) and `interpretation` (the result of `stability::stab_interpret()`).

Author(s)

Muhammad Yaseen (<myaseen208@gmail.com>)

See Also

`sa_ask`, `StabilityApp`

Examples

```
fit <-
  stability::ammi(
      .data = stability::ge_data
    , .y    = Yield
    , .rep  = Rep
    , .gen  = Gen
    , .env  = Env
    )
p <-
  sa_prompt(
      x             = fit
    , higher_better = TRUE
    , trait         = "grain yield (kg/ha)"
    , rows          = 15
    )
cat(p$user)
```

`sa_services` *External text services for a written interpretation*

Description

`sa_services()` lists the external text services the application can ask for a longer written interpretation, and `sa_ask()` sends one request to such a service and returns its text. Using a service is optional: the application always shows the interpretation of `stability::stab_interpret()`, which is computed from the results and needs no connection.

Usage

```
sa_services()

sa_ask(
  service,
  prompt,
  model = NULL,
```

```
  key = "",
  max_tokens = 1500,
  timeout = 90
)
```

Arguments

service	identifier of a service (column id of sa_services()).
prompt	a list with elements system and user (as returned by sa_prompt()), or a single character string.
model	model name (default: the one of sa_services()).
key	the user's key for the service; "" (default) uses the environment variable of the service.
max_tokens	largest length of the answer, in tokens (default 1500).
timeout	seconds to wait (default 90).

Details

The services are read from the file services.csv of the package (columns id, label, format, url, model, key_env, and header, an extra header name=value). Three request formats are supported: "messages", "chat" and "generate", the formats of the three services in the file. The key is taken from the argument key or, when it is empty, from the environment variable named in key_env, so that it need not be typed into the application. The request is sent over HTTPS to the service's own address; the key is never stored by the package.

What is sent is the text built by sa_prompt(): the name of the analysis, the trait, the size of the trial, the interpretation of stab_interpret(), its facts and the first rows of the main table. No data file is sent.

Value

sa_services(): a data frame of the services. sa_ask(): the text of the answer (character), or an error with the message of the service.

Author(s)

Muhammad Yaseen (<myaseen208@gmail.com>)

See Also

sa_prompt, StabilityApp

Examples

```
sa_services()[, c("id", "format", "model")]
## Not run:
fit <-
  stability::stab_reg(
      .data = stability::ge_data
```

```
    , .y    = Yield
    , .rep  = Rep
    , .gen  = Gen
    , .env  = Env
    )
p <-
  sa_prompt(
      x             = fit
    , higher_better = TRUE
    , trait         = NULL
    , rows          = 15L
    )
sa_ask(
    service    = sa_services()$id[1]
  , prompt     = p
  , model      = sa_services()$model[1]
  , key        = ""
  , max_tokens = 1500
  , timeout    = 90
  )

## End(Not run)
```

`StabilityApp` *Stability analysis of multi-environment trials in a web browser*

Description

`StabilityApp()` starts the application: load a trial (the example `ge_data` of **stability** or a CSV file with one row per plot), choose the columns, and run any of the twenty analyses of **stability**. Each analysis shows an interpretation written for its method and computed from its results (`stability::stab_interpret()`), its tables (with CSV and Excel export), its plots (PNG and PDF), and the R code that repeats it. A longer written interpretation can be requested from an external text service with your own key (`sa_services()`; optional).

Usage

```
StabilityApp(launch.browser = interactive(), ...)

sa_app()
```

Arguments

`launch.browser`	passed to `shiny::runApp()`: `TRUE` (default in an interactive session) opens the application in the web browser.
`...`	further arguments to `shiny::runApp()`, such as `port`.

Details

The analyses: means and rankings, interaction effects; analysis of variance in each environment, combined, and Eberhart and Russell's; Finlay and Wilkinson's regression, variance measures, Shukla's stability variance with Kang's yield-stability statistic, Kang's rank-sum, Fox's top third; AMMI analysis, AMMI biplots, AMMI stability value, modified AMMI stability value, AMMI stability distance; GGE biplot (which won where, mean against stability, discrimination).

The key of a text service is typed on the Settings page (it stays in the session and is not saved) or read from the environment variable named in `sa_services()`.

Value

Called for its side effect; returns the value of `shiny::runApp()` when the application is closed. `sa_app()` returns the application object (for `shiny::runApp()`, `shiny::testServer()` or deployment).

Author(s)

Muhammad Yaseen (<myaseen208@gmail.com>), Kent M. Eskridge (<keskridge1@unl.edu>)

See Also

`sa_prompt`, `sa_services`

Examples

```
if (interactive()) {
  StabilityApp(
      launch.browser = TRUE
    )
}
```

Index

sa_app *(StabilityApp)*, 5
sa_ask, *2, 3*
sa_ask *(sa_services)*, 3
sa_prompt, 2, *4*, *6*
sa_services, 3, *5, 6*
StabilityApp, *3, 4*, 5