The Journal of Open Source Software

shinyshval: A User-Friendly Shiny implementation of the EFSA SHVAL tool for the use in Bee risk assessment

Eduard Szöcs[*1] **and Tobias Pamminger**[1, 2]

1 BASF SE, Digitalization of R&D, Carl-Bosch-Straße 38, 67063 Ludwigshafen am Rhein, Germany
2 Current address: Bayer AG, Division CropScience, Alfred-Nobel-Straße 50, 40789 Monheim am Rhein, Germany

DOI:

Software

- Review
- Repository
- Archive

Submitted:
Published:

Summary

In 2013 the European Food Safety Authority (EFSA) published a guidance document for the proposed revision of the bee risk assessment of plant protection products in the European Union (EFSA, 2013). This revision includes the introduction of the shortcut values (SV) as an estimate for the expected oral uptake of plant protection products residues by bees. To enable the users to modify the provided default input parameters and the re-calculation of the relevant SV, EFSA has published R-scripts for the calculations: the SHVAL tool (Zancanaro, Cortiñas Abrahantes, Boesten, & Szentes, 2014).

`shinyshval` wraps these R-scripts in an R package and provides an easy to use graphical user interface to calculate SV values, using the Shiny web framework (Chang et al., 2021). `shinyshval` provides simple and convenient access to core functionalities provided by the original SHVAL tool, without the need of the users to be familiar with the R programming language.

The aim of `shinyshval` is to facilitate accessibility of the SHVAL tool to all relevant users and provide a packaged, documented, tested and open-source set of functions.

Statement of need

The calculation of SV values and modification of its input parameters, depending on the expected exposure scenario, is a core component of the envisioned future bee Risk Assessment in the European Union. However, the EFSA provided SHVAL tool required familiarity with the R programming language and specific formatting of input data files. By providing a web based graphical user interface, the `shinyshval` will facilitate the accessibility of the SHVAL tool to the wider community, enable future users without prior R knowledge to access this risk assessment calculation tool and could serve as inspiration for the development of the next generation of R based regulatory tools.

Use

The `shinyshval` app closely follows the SHVAL tool implementation outlined in the associated technical report which provides a detailed explanation of the performed calculations and required input variables (Zancanaro et al., 2014). Default parameters for

*corresponding author

The Journal of Open Source Software

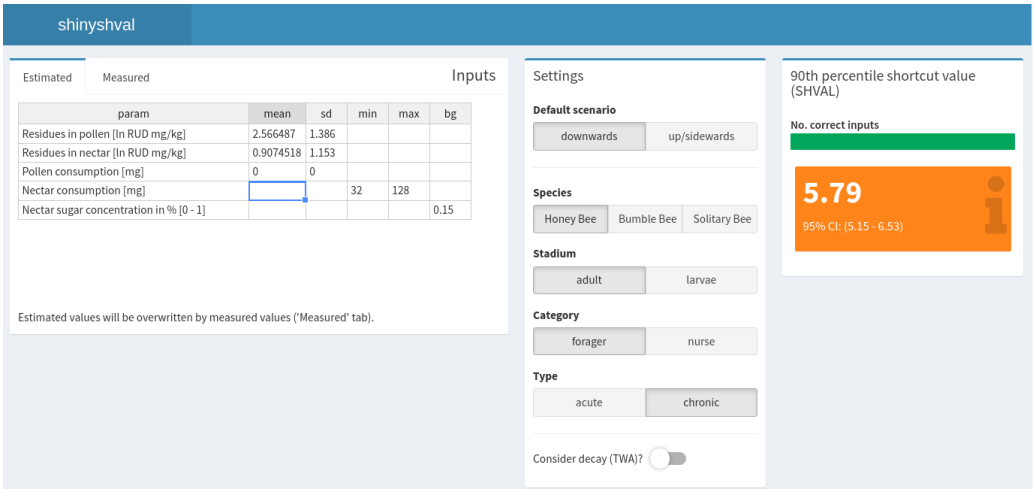

Figure 1: Screenshot of shinyshval. On right parameters can be changed. In the middle default parameters be selected. The right side shows the computed shval.

different bee species and types have been defined (according to EFSA Guideline) and can be selected. Help and description is directly included in the application.

The app can be started from R with the following code

```
# install shinyshval from github repository
install.packages("remotes")
remotes::install_github("basf/shinyshval")
# start app
shinyshval::run_app()
```

It can be also easily be deployed using docker (Boettiger & Eddelbuettel, 2017). A demo app using docker can be spin up on binder: https://mybinder.org/v2/gh/basf/shinyshval/binder?urlpath=shiny

Of course the back-end functionality to compute SHVAL values in R without app is also possible and described in the README.

Quality Assurance

The initial interactive R scripts have been split and wrapped into functions and an R package. This provides basic infrastructure for checking that the package is in full working order.

In order to ensure consistent results between the `shinyshval` and original EFSA SHVAL tool, the package is under version control and an extensive set of tests have been implemented using `testthat` (Wickham, 2011).

In short, a total number of 192 continuously integrated tests are run on each code change. These tests that cover 100% of back-end functionality.

- 156 for the internal functioning of function for the app
- 36 tests of 9 scenarios specifically aimed at comparison of results with the official EFSA script

Because of the non-deterministic nature of the computations, comparisons are made with a precision of $1x10^{-1}$ to $1x10^{-2}$.

Availability and Community Guidelines

The software is available at the GitHub repository. The GitHub repository also contains the source code for this paper and a contribution guide.

Acknowledgement

We thank EFSA for supporting the authors publishing `shinyshval` under an open-source license.

Disclaimer

`shinyshval` is released with agreement of EFSA under EUPL-1.2 open source license. According to articles 7 and 8 of the EUPL-1.2 license EFSA is not responsible for errors, omissions or deficiencies regarding `shinyshval`; `shinyshval` is being made available as such without any warranty either expressed or implied, including, but not limited to, warranties of performance, merchantability, and fitness for a particular purpose; In no event shall EFSA be liable for direct, indirect, special, incidental, or consequential damages resulting from the use, misuse, or inability to use `shinyshval`.

References

Boettiger, C., & Eddelbuettel, D. (2017). An Introduction to Rocker: Docker Containers for R. *The R Journal*, *9*(2), 527–536. doi:10.32614/RJ-2017-065

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., et al. (2021). *shiny: Web Application Framework for R*. Retrieved from https://CRAN.R-project.org/package=shiny

EFSA. (2013). Guidance on the risk assessment of plant protection products on bees (apis mellifera, bombus spp. And solitary bees). *EFSA Journal*, *11*(7), 3295. doi:10.2903/j.efsa.2013.3295

Wickham, H. (2011). testthat: Get Started with Testing. *The R Journal*, *3*(1), 5–10. doi:10.32614/RJ-2011-002

Zancanaro, G., Cortiñas Abrahantes, J., Boesten, J., & Szentes, C. (2014). *Application for the estimation of the residue intake rate for certain bee species under given conditions (SHVAL tool)*. Zenodo. doi:10.5281/zenodo.252130