Beyond R CMD check: Helping R developers to detect CRAN package conflicts

Malick Claes[⋆], Tom Mens, Philippe Grosjean

COMPLEXYS Research Institute, University of Mons, Belgium
*Contact author: maelick.claes@umons.ac.be

Keywords: CRAN, package dependencies, conflicts, maintainability, tool support

CRAN is a large software collection containing thousands of *R* packages maintained by thousands of different maintainers. The number of packages is growing very rapidly (currently there are over 5000 packages), which is considered by some as problematic [2]. Another problem is a lack of coordination between developers of dependent software components. Maintainability problems may arise and packages may cease to function correctly because of unexpected changes made to the packages they depend upon. In addition, problems with the dependency versioning system of *R* have been reported and possible directions for improvement have been proposed, such as staged package distributions (as in Debian) and versioned package management [3].

Currently, *R* package developers can use the R CMD check command to detect possible problems in CRAN contributed packages. This tool is also used to ensure conformance of accepted packages to the CRAN quality policy, and to check that packages don't break over time. However, since the number of packages is growing quickly, it becomes harder and harder to solve problems that are due to updates of packages that one directly or indirectly depends upon. We have studied the extent of this problem through an empirical analysis of CRAN's R CMD check results [1]¹. We observed that package quality and maintainability varies with the operating system considered. We also observed that a non-negligible amount of errors are caused by dependency updates and need to be fixed by the maintainers. Maintenance effort hence needs to take into account changes made to package dependencies. This may become detrimental to package maintainability in the long run if the number of CRAN packages keeps on growing at the same pace.

Therefore, there is a need for more specific tools dedicated to R package developers, that allow them to gain insight and deal with the implications and problems raised by package updates. Will changes to their packages cause potential problems to other CRAN packages? Do package updates or changes in the dependencies of other packages cause potential problems in one's own package? Being able to address such problems *a priori* during package development and maintenance, i.e., long before submitting it to CRAN, will reduce the effort of maintaining contributed CRAN packages.

We will report on a prototype tool that we have developed for the above.² It is more specific and fine-grained than the R CMD check and it considers potential conflicts with *all* CRAN packages, not only those currently tested. It aims to help *R* package maintainers to identify and avoid problems that could break their own package or those of others *before* sending it to CRAN. The tool is based on a fine-grained function-level analysis of dependencies, conflicts and clones (copy-paste reuse of code) between packages.

References

- [1] Claes, M., T. Mens, and P. Grosjean (2014). On the maintainability of CRAN packages. In *IEEE CSMR-WCRE* 2014 International Conference.
- [2] Hornik, K. (2012). Are there too many R packages? Austrian Journal of Statistics 41(1), 59-66.
- [3] Ooms, J. (2013, June). Possible directions for improving dependency versioning in R. R Journal 5(1), 197–206.

¹R package available at github.com/maelick/extractoR. CRAN historical data available at github.com/maelick/CRANData.

²R package available at github.com/maelick/maintaineR.