

# ropenblas: Download, Compile and Link OpenBLAS Library with R

Pedro Rafael Diniz Marinho<sup>1</sup>

<sup>1</sup> Department of Statistics, Federal University of Paraíba, João Pessoa, Paraíba - PB, Brazil

DOI: [10.21105/joss.02769](https://doi.org/10.21105/joss.02769)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

---

Editor: [Tim Tröndle](#) ↗

## Reviewers:

- [@pratikvn](#)
- [@myousefi2016](#)

Submitted: 01 October 2020

Published: 22 March 2021

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

The `ropenblas` package aims to facilitate the day-to-day life of R programmers who want more performance on GNU/Linux systems, without removing the possibility that specific configurations are made, if they deem convenient, that is, more technical users will be able to pass other flags that will be considered in the compilation process. Through the package's `ropenblas()` and `rcompiler()` functions, the library user will be able to compile and link the R language in his GNU/Linux distribution with the OpenBLAS library, all within R and in a very simple fashion. All functions work without being influenced by the GNU/Linux distribution and are independent of their repositories, that is, it does not matter which GNU/Linux distribution is being used. Linking the OpenBLAS library to R will bring better computational performance to the language in the most diverse algebraic operations commonly used in areas such as statistics, data science, and machine learning.

## Introduction

The term “computational efficiency” is very common for those who program statistical methods, in which a large part of them involve algebraic operations that are often reproduced in computationally intensive simulations, such as Monte-Carlo simulations - MC and resampling methods, as is the case with bootstrap resampling. Statistics is just one example within so many other areas that need performance and uses the R language.

In addition to the adoption of good programming practices and the maximum, efficient and adequate use of available computational resources, such as code parallelization, through multicore parallelism procedures allowed by most current processors and operating systems, small adjustments and linkage of libraries can provide useful benefits.

The `ropenblas` package aims to provide useful and simple experiences to R ([R Core Team, 2016](#)) programmers who develop their activities on GNU/Linux operating systems. These experiences consist of being able to link any version of the OpenBLAS ([Xianyi, Zhang, Wang Qian, and Werner Saar, 2016](#)) library to the R language, as well as allowing the programmer to install and link various versions of R and make them available on their operating system as well as switch between these versions as they see fit.

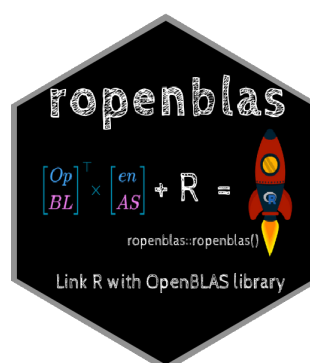
Linking the R language to the OpenBLAS library can bring several benefits to algebraic computing in R. OpenBLAS is an Open-Source implementation of the Basic Linear Algebra Subprograms - BLAS library that is often the first library option for algebraic computing to be linked in the installation of R on many GNU/Linux distributions. The OpenBLAS library is available at <https://github.com/xianyi/OpenBLAS> and adds optimized implementations of linear algebra kernels that can run optimized on various processor architectures. OpenBLAS

is based on the GotoBLAS2 project code in version 1.13 (Goto, 2010), code available under the terms of the BSD license.

The `ropenblas` is a package designed to facilitate the linking of the library OpenBLAS with the language R. The package, which works only for Linux systems, will automatically download the latest source code from the OpenBLAS library and compile the code. The package will automatically bind the language R, through the `ropenblas()` function, to use the OpenBLAS library. Everything will be done automatically regardless of the Linux distribution you are using. Enumerating some advantages of the package:

1. Everything is done within the R language;
2. The procedure (use of functions) will be the same for any Linux distribution;
3. The OpenBLAS library will be compiled and you will choose which build version to bind to R, regardless of your Linux distribution;
4. The package allows you to install  $R \geq 3.1.0$ , also allowing you to install one more version, in addition to allowing you to easily switch between those versions;
5. The linked versions of R will continue to be recognized by their Integrated Development Environment - IDE and nothing will have to be adjusted in your GNU/Linux distribution after using any function of the package;
6. Unnecessary builds will be avoided. Therefore, if you need to switch between compiled versions of the R language, the use of binaries compiled previously will be suggested;
7. If any errors occur, the functions of the package will not damage the previous installation of the language;
8. If something better can be done or if a newer version of what you want to install (R or OpenBLAS) exists, the functions will automatically suggest that you should consider installing newer versions.

The `ropenblas` package is already available on the Comprehensive R Archive Network - CRAN, currently in version 0.2.9, and the project is maintained on GitHub at <https://github.com/prdm0/ropenblas> where contributors can find other details of the code, information, as well as being able to contribute with the development of the project. On the website, it is also possible to read the `NEWS.md` file with details of the versions and the focus of the current development. The site is deposited at <https://prdm0.github.io/ropenblas/>. Suggestions for improvements and bug reports can be sent via the link <https://github.com/prdm0/ropenblas/issues>. You can find out how to contribute to the package by accessing the `CONTRIBUTING.md` file at <https://github.com/prdm0/ropenblas/blob/master/CONTRIBUTING.md>.



**Figure 1:** Computer library logo.

## Brief explanation

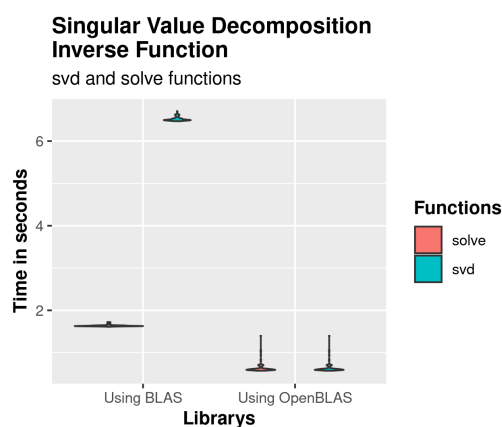
The `ropenblas` library exports six functions for use which are the `rcompiler()`, `ropenblas()`, `last_version_r()`, `last_version_openblas()`, `link_again()` and `rnews()`. All of them are very simple to use and have few arguments that are sufficient to maintain the flexibility of use. Any example that follows will consider that the installation of the `ropenblas` package has been carried out and the package has been loaded (`library(ropenblas)`). Also, functions like `rcompiler()` and `ropenblas()` do not return content or data structures that are of any practical use. What these functions do is configure the GNU/Linux system to use R, configure different versions of the language, switch between versions, and link with the OpenBLAS library. It is also possible to obtain a summary of the versions of R and the OpenBLAS library that are available.

Table 1 below presents the benefit of considering an optimized version of BLAS. Computational costs are presented in the calculation of the singular decomposition in a rectangular matrix (`svd()` function) and in the calculation of the inverse of that same matrix (`solve()` function). Some repetitions (100 repetitions) of each of the respective functions were performed. The benchmark can be better observed through the violin plots shown in Figure 2.

**Table 1:** Comparison of the computational costs of the `svd()` and `solve()` functions (average of 100 repetitions).

Functions	Library	Time (seconds)
<code>svd(x)</code>	BLAS	6.520
<code>svd(x)</code>	OpenBLAS	0.641
<code>solve(x)</code>	BLAS	1.640
<code>solve(x)</code>	OpenBLAS	0.640

Through a benchmark it is possible to better understand the performance gain that can be achieved by linking the R language to the OpenBLAS library. Figure 2 presents the benchmarks in the form of a violin plot, in which 100 reproductions of the `svd(X)` expression were considered, in the form of the code above, with the R linked to the BLAS library and linked to the OpenBLAS library, respectively, on the same hardware. It was observed that the average time of execution of the routine `svd(X)` considering the OpenBLAS library was less than 10 times the time necessary to execute it in R linking to a non-optimized version of BLAS, being the average time of 0.64 and 6.52 seconds, respectively.



**Figure 2:** Benchmarks of a decomposition of singular and inverse value of a matrix of dimension  $1000 \times 1000$ .

## Improvements

The package will continue to evolve and code reviews will always be carried out. In addition, contributions to the development of the package are always welcome, especially those that aim to allow the `rcompiler()` and `ropenblas()` functions to work on Windows systems. There is also an interest that the `ropenblas` package will allow the linking of the Intel Math Kernel Library - MKL, just as it is done with the OpenBLAS library. All of these are improvements that we would like to see in future versions of the package.

## References

- Goto, K. (2010). GotoBLAS2 1.13 BSD version. In *Texas Advanced Computing Center*. <https://www.tacc.utexas.edu/research-development/tacc-software/gotoblas2>
- R Core Team. (2016). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Xianyi, Zhang, Wang Qian, and Werner Saar. (2016). OpenBLAS: An optimized BLASLibrary. In *Texas Advanced Computing Center*. <http://www.openblas.net/>