

# randomForestSRC: Installing randomForestSRC Vignette

Hemant Ishwaran, Min Lu and Udaya B. Kogalur

## Creating and Installing the randomForestSRC R Package

Regular stable releases of this package are available on CRAN and on the master branch on our GitHub repository. Interim, sometimes unstable, development builds with bug fixes and/or additional functionality are available on the develop branch of our GitHub repository.

Like many other R packages, the simplest way to obtain randomForestSRC is to install it directly from CRAN via typing the following command in R console:

```
install.packages("randomForestSRC", repos = "https://cran.us.r-project.org")
```

To create the R package using the GitHub repository, you will need an installation of R ( $> v3.0$ ) that is capable of compiling source code packages containing C-code. This means that the appropriate C-code compilers need to be in place and accessible by the R packaging and installation engine. Detailed descriptions on how this is achieved are available on a number of sites online and will not be reproduced here. You will also need Apache Ant (v1.10), and Java JDK (v1.80). Once the R package development environment is in place, it is possible to build our package natively on your platform using the following steps:

After obtaining and unzipping the source code from the GitHub repository, from the top-level directory (the directory containing build.xml), the command

```
ant
```

will give you several options. The command

```
ant source-cran
```

will create the R source code package directory-tree `./target/cran/randomForestSRC/`. To install randomForestSRC in your default library, change to the directory `./target/cran/` and type

```
R CMD INSTALL --preclean --clean randomForestSRC
```

This will install an OpenMP parallel version of the package if the host system is capable of supporting this mode of execution.

Please note that on some platforms, even though an OpenMP C-compiler may have been installed, the R packaging and installation engine does not pick up the appropriate compiler. For example, on macOS, the default compiler is Clang. It is not OpenMP capable out-of-the-box. You will need to install an OpenMP version of it, or install GCC using Homebrew or another package manager. Most importantly, you will also need to direct the R packaging and installation engine to the OpenMP capable compiler. This is done by creating an `.R` directory in your `HOME` directory, and creating a `Makevars` file in that directory containing the appropriate compiler instructions. As an example, on macOS Sierra (v10.12) our installation has the following as its `Makevars` file:

```
F77 = gfortran-7
FC = gfortran-7
CC = gcc-7
CXX = g++-7
CFLAGS = -I/usr/local/Cellar/gcc/7.2.0/include
LDFLAGS = -L/usr/local/Cellar/gcc/7.2.0/lib/gcc/7
```

To set number of CPU parameters for parallel processing, see the Parallel Processing vignette [1].

## The Apache Spark Package

To create the Apache Spark package using the GitHub repository, you will need the following tools: Apache Ant (v1.10), Java JDK (v1.80), Scala (v2.12), and Apache Maven (v3.5). You must also have Apache Spark (v2.1) installed.

After obtaining and unzipping the source code from the GitHub repository, from the top-level directory (the directory containing build.xml), the command

```
ant
```

will give you several options. The command

```
ant source-spark
```

will create the Spark source code package directory-tree `./target/spark/`. To compile the the source code package, type

```
ant build-spark.
```

This will create the Spark target package directory-tree `./target/spark/target/`. A sample *helloRandomForestSRC* program can be executed by changing to the directory `./target/spark/target/` and typing `./hello.sh` or `./hello.cmd` according to your operating system. The source code for the example is located in our GitHub repository. It does little more than start a Spark session, grow a forest, and stop the Spark session. Details of raw unformatted ensemble information is presented in a log file `rfsrc-x.log` in the users `HOME` directory, though they are not available for examination by the user at this point in any coherent way.

The Java API Specification for `randomForestSRC` is available. It is purely skeletal at this point, but will be flushed out in more detail in the near future.

### Cite this vignette as

H. Ishwaran, M. Lu, and U. B. Kogalur. 2021. "randomForestSRC: installing randomForestSRC vignette." <http://randomforestsrc.org/articles/installation.html>.

```
@misc{HemantInstall,
  author = "Hemant Ishwaran and Min Lu and Udaya B. Kogalur",
  title = {{randomForestSRC}}: installing {{randomForestSRC}} vignette},
  year = {2021},
```

```
url = {http://randomforestsrc.org/articles/installation.html}  
}
```

1. Ishwaran H, Lu M, Kogalur UB. randomForestSRC: Parallel processing vignette. 2021. <http://randomforestsrc.org/articles/parallel.html>.