

# Using Rcpp\* packages for easy and fast extension of R with C++



Ghislain VIEILLEDENT and Jeanne CLEMENT

**Cirad**, UMR AMAP, Montpellier, FRANCE



# Outline

## 1 Short presentation of Rcpp\* packages

- Rcpp : extending R with C++
- RcppGSL for fast random draws
- RcppArmadillo for high-performance linear algebra

## 2 Build of R package using Rcpp

- Useful functions
- Example of package using Rcpp



# Rcpp R package

- **Rcpp** is an R package to extend R with C++ code
- Main advantage : C++ is fast, it accelerates R as shown in the examples
- Written by **Dirk EDDELBUETTEL** and **Romain FRANCOIS**
- <http://www.rcpp.org/>

# Simple Rcpp example

## C++ code (in file Code/addition.cpp)

```
#include <Rcpp.h>
using namespace Rcpp;

// [[Rcpp::export]]
int addition(int a, int b) {
  return a + b;
}
```

## R code

```
Rcpp::sourceCpp("Code/addition.cpp")
addition(2, 2)

## [1] 4
```



# Useful Rcpp features

## Rcpp::export preceding the C++ function definition

- Indicate that it should be made available as an R function
- Check if the function have return and entry types compatible with Rcpp i.e. convertible into R object according to the following correspondence tables of data types :



# Useful Rcpp features

## Rcpp::export preceding the C++ function definition

- Indicate that it should be made available as an R function
- Check if the function have return and entry types compatible with Rcpp i.e. convertible into R object according to the following correspondence tables of data types :

Value	R vector	Rcpp vector	Rcpp matrix	R	Rcpp
Logical	logical	LogicalVector	LogicalMatrix		
Integer	integer	IntegerVector	IntegerMatrix		
Real	numeric	NumericVector	NumericMatrix		
Complex	complex	ComplexVector	ComplexMatrix		
String	character	CharacterVector ( StringVector )	CharacterMatrix ( StringMatrix )	data.frame	DataFrame
				list	List
				S3 class	List
				S4 class	S4



# Useful Rcpp features

## Thanks to `Rcpp:::sourceCpp()`

- Compile the C++ code
- Export the function to the R session
- Direct conversion of objects (including S3, S4) between R and C++
- For more details see `vignette("Rcpp-package")`



# Useful Rcpp features

## Thanks to `Rcpp::sourceCpp()`

- Compile the C++ code
- Export the function to the R session
- Direct conversion of objects (including S3, S4) between R and C++
- For more details see `vignette("Rcpp-package")`

## Thanks to `Rcpp::depends`

- Specify additional build dependencies on other packages for `Rcpp::sourceCpp()`
- `Rcpp::sourceCpp()` therefore configures the build environment to correctly compile and link against the package specified
- examples : `[[Rcpp::depends(RcppArmadillo)]],`  
`[[Rcpp::depends(RcppGSL)]]` ...



# GSL and RcppGSL



## GNU Scientific Library

- Numerical library for C and C++ programmers
- Reliable random number generator algorithms
- Thoroughly tested and fast random number distributions
- Linear algebra (matrices and vectors)
- <https://www.gnu.org/software/gsl/>

## RcppGSL

- Interface between R and GSL
- Using Rcpp to interface R and C
- <http://dirk.eddelbuettel.com/code/rcpp.gsl.html>



## GSL random number distributions

- GSL v2.6 includes **38 random number distributions** (see [GNU GSL](#))
- For comparison, R API includes “only” 24 random number distributions (see [Writing R Extensions](#))





# GSL random number distributions

- GSL v2.6 includes **38 random number distributions** (see [GNU GSL](#))
- For comparison, R API includes “only” 24 random number distributions (see [Writing R Extensions](#))
- It’s easy to implement additional random number distributions from the GSL base distributions (e.g. truncated normal or inverse gamma distribution)
- Random draws are faster with GSL than with R (eg. `gsl_ran_gamma()` vs. `R::rgamma()`)





# GSL random number distributions



- GSL v2.6 includes **38 random number distributions** (see [GNU GSL](#))
- For comparison, R API includes “only” 24 random number distributions (see [Writing R Extensions](#))
- It’s easy to implement additional random number distributions from the GSL base distributions (e.g. truncated normal or inverse gamma distribution)
- Random draws are faster with GSL than with R (eg. `gsl_ran_gamma()` vs. `R::rgamma()`)
- Examples of functions to generate random sample :  
`gsl_ran_multivariate_gaussian()`,  
`gsl_ran_wishart()`, `gsl_ran_dirichlet()`
- Probability density function (`_pdf`), upper or lower cumulative distribution functions (`_Q` or `_P`) and quantile functions (`_Qinv` or `_Pinv`) are also available for each distribution.



# Armadillo and RcppArmadillo

## Armadillo

- C++ library for linear algebra and scientific computing
- Provides high-level syntax and functionality : speed and ease of use
- Classes for vectors, matrices and cubes convertible in R vector, matrix and array.
- Matrix operations (+, -, \*, %), identify the elements of a matrix that meet a condition (`arma::find(A>0)`), matrix decomposition (`arma::chol()`), linear model solver (`arma::solve(A,B)`), etc.
- <http://arma.sourceforge.net/>





# Armadillo and RcppArmadillo



## Armadillo

- C++ library for linear algebra and scientific computing
- Provides high-level syntax and functionality : speed and ease of use
- Classes for vectors, matrices and cubes convertible in R vector, matrix and array.
- Matrix operations (+, -, \*, %), identify the elements of a matrix that meet a condition (`arma::find(A>0)`), matrix decomposition (`arma::chol()`), linear model solver (`arma::solve(A,B)`), etc.
- <http://arma.sourceforge.net/>

## RcppArmadillo

- Interface between R and Armadillo
- Using Rcpp to interface R and C++
- <http://dirk.eddelbuettel.com/code/rcpp.armadillo.html>



# Licenses

- Licenses : GNU General Public License, Apache License 2.0 for Armadillo
- Free software licenses : we can use, modify and redistribute these softwares



# How to build an R package around C++ functions

- `Rcpp.package.skeleton()` to generate a new Rcpp package (modifying `DESCRIPTION` and `NAMESPACE`)
- `Rcpp::compileAttributes()` scans the C++ files and generates the `RcppExports.cpp` file to make the functions preceded by `[[Rcpp::export]]` available in R.
- Implement R function that checks the conformity of user-defined parameters, calls functions in C++ and returns the results in an easy-to-use format.

# jSDM R package

jSDM 0.1.0  Get started Reference Articles ▾ Change log  

## jSDM R Package

Package for fitting joint species distribution models (jSDM) in a hierarchical Bayesian framework (Warton *et al.* 2015). The Gibbs sampler is written in C++. It uses Rcpp, Armadillo and GSL to maximize computation efficiency.

### System requirements

Make sure the GNU Scientific Library (GSL) is installed on your system.

### Installation

Install the latest stable version of jSDM from CRAN with:

```
install.packages("jSDM")
```

Or install the development version of jSDM from GitHub with:

```
devtools::install_github("ghislainv/jSDM")
```

### References

Warton, D.I., Blanchet, F.G., O'Hara, R.B., Ovaskainen, O., Taskinen, S., Walker, S.C. & Hui, F.K. (2015) So many variables: Joint modeling in community ecology. *Trends in Ecology & Evolution*, **30**, 766–779.

### Links

Download from CRAN at <https://cloud.r-project.org/package=jSDM>  
Browse source code at <https://github.com/ghislainv/jSDM>  
Report a bug at <https://github.com/ghislainv/jSDM/issues>

### License

GPL-3 | file LICENSE

### Developers

Ghislain Vieilledent  
Author, maintainer   
Jeanne Clément  
Author 

 cirad  
Copyright holder, funder

### Dev status

build  CRAN 0.1.0  DOI 10.5281/zenodo.3253460  downloads 231/month 

- <https://ecology.ghislainv.fr/jSDM>
- Made with Rcpp, RcppGSL and RcppArmadillo packages



... Thank you for attention ...



@ghislainv

<https://ecology.ghislainv.fr>

ghislain.vieilledent@cirad.fr | jeanne.clement16@laposte.net