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

Ghislain VIEILLEDENT and Jeanne CLEMENT

Cirad, UMR AMAP, Montpellier, FRANCE

cirad

AMAP botAnique et Modélisation de l'Architecture des Plantes et des végétations

Outline

1. Short presentation of Rcpp* packages
   - Rcpp : extending R with C++
   - RcppGSL for fast random draws
   - RcppArmadillo for high-performance linear algebra
2. Application examples
   - Distance computation
   - Simple linear regression
3. Exercise to do yourself
4. Build of R packages using Rcpp
   - Usefull functions
   - Example of package built with Rcpp

Rcpp R package

- **Rcpp** is an R package to extend R with C++ code
- Main advantage : C++ is fast, it accelerates R (see next sections)
- 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
```

Rcpp advantages

Thanks to `Rcpp::sourceCpp()`

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

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)
- It's easy to implement additional random number distributions from the GSL base distributions (e.g. truncated normal distribution)
- For comparison, R API includes "only" 24 random number distributions (see Writing R Extensions)
- Random draws are faster with GSL than with R (eg. `gsl_ran_gamma()` vs. `R::rgamma()`)

RcppGSL example

C++ code

```
#include <Rcpp.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
using namespace Rcpp;
// [[Rcpp::depends(RcppGSL)]]
// [[Rcpp::export]]
Rcpp::NumericVector my_rnorm(int nsamp, double mu,
                             double sigma) {
    gsl_rng *s = gsl_rng_alloc(gsl_rng_mt19937); // Random seed
    Rcpp::NumericVector beta(nsamp);
    for (int i = 0; i < nsamp; i++) {
        beta[i] = mu + gsl_ran_gaussian(s, sigma); // Random draw
    }
    return beta;
}
```

RcppGSL example

R code

```
library(Rcpp)
library(RcppGSL)
beta <- my_rnorm(100, 5, 2)
par(cex=2)
hist(beta)
```

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
- Matrix operations, matrix decomposition, linear model solver, 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

RcppArmadillo example

C++ code

```
#include <RcppArmadillo.h>
// [[Rcpp::depends(RcppArmadillo)]]
// [[Rcpp::export]]
Rcpp::List fastLm(const arma::mat& X, const arma::colvec& y) {
    int n = X.n_rows, k = X.n_cols;

    arma::colvec coef = arma::solve(X, y);    // fit model y ~ X
    arma::colvec res  = y - X*coef;           // residuals
    // std.errors of coefficients
    double s2 = std::inner_product(res.begin(),
                                   res.end(),
                                   res.begin(), 0.0)/(n - k);

    arma::colvec std_err = arma::sqrt(s2 *
      arma::diagvec(arma::pinv(arma::trans(X)*X)));
    return Rcpp::List::create(Rcpp::Named("coefficients") = coef,
                              Rcpp::Named("stderr")       = std_err,
                              Rcpp::Named("df.residual")  = n - k);
}
```

RcppArmadillo example

R code

```
library(Rcpp)
library(RcppArmadillo)
# Trees data-set
y <- log(trees$Volume)
X <- cbind(1, log(trees$Girth))
# fastLm
mod <- fastLm(X, y)
mod$coef
```

```
##             [,1]
## [1,] -2.353325
## [2,]  2.199970
```

Licenses

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

Distance computation

C++ code (in file Code/arma_distmat.cpp)

```
#include <RcppArmadillo.h>
// [[Rcpp::depends(RcppArmadillo)]]
using namespace Rcpp;
using namespace arma;

// [[Rcpp::export]]
arma::mat arma_distmat(const arma::mat& x) {

  int np = x.n_rows;

  arma::mat distmat; distmat.zeros(np, np);
  for (int i = 0; i < np; i++) {
    arma::vec p0 = x.row(i).t();
    for (int j = i + 1; j < np; j++) {
      arma::vec p1 = x.row(j).t();
      arma::vec diff = p0 - p1;
      double squared_diff = as_scalar(diff.t() * diff);
      distmat(j, i) = distmat(i, j) = sqrt(squared_diff);
    }
  }
  return distmat;
}
```

Distance computation

R code

```
Rcpp::sourceCpp("Code/arma_distmat.cpp")
library(rdist)
X <- matrix(rnorm(10000),ncol=2)

# Benchmark
library(rbenchmark)
Benchmark <- benchmark(
  "arma_distmat" = {arma_distmat(X)},
  "cdist" = {cdist(X,X)},
  replications=10,
  columns = c("test", "elapsed", "relative"))
knitr::kable(Benchmark, booktabs=TRUE) %>%
        kableExtra::kable_styling(latex_options=c("HOLD_position","striped"),
                                  full_width=FALSE)
```

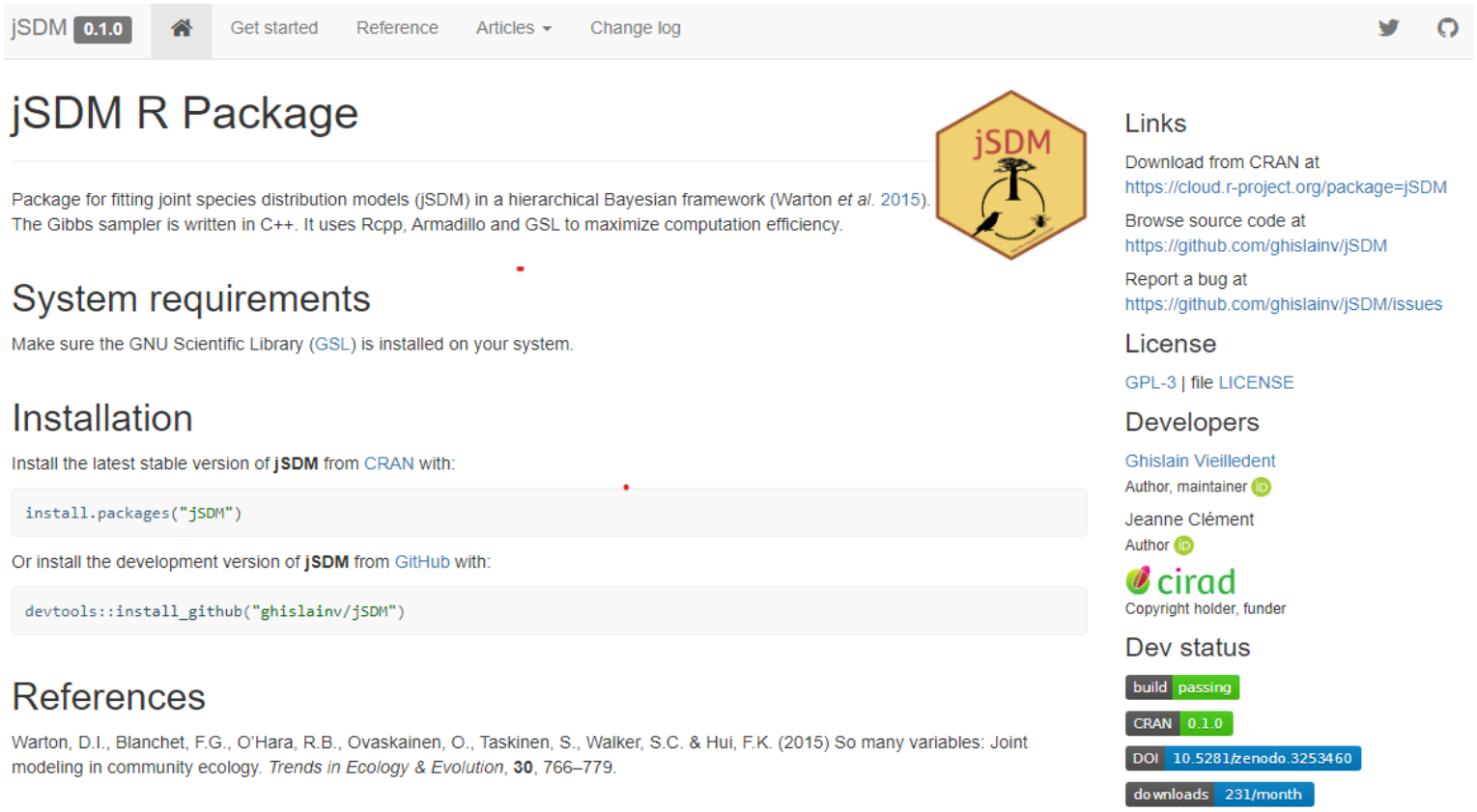
test	elapsed	relative
arma_distmat	5.788	1.00
cdist	15.104	2.61

Simple linear regression

Functions to build an R package

- `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.

jSDM R package



- https://ecology.ghislainv.fr/jSDM
- Made with Rcpp* packages

. . . Thank you for attention . . .

@ghislainv

https://ecology.ghislainv.fr

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