rClr package – low level access to .NET code from R

Jean-Michel Perraud

1. Commonwealth Scientific and Industrial Research Organisation, Australia
*Contact author: jean-michel.perraud@csiro.au

Keywords: interoperability, .NET, Mono, Common Language Runtime

rClr (http://r2clr.codeplex.com) is a package for R to access arbitrary .NET code executing on a Common Language Runtime implementation. The package can access the two main CLR implementations. On Windows® Microsoft’s implementation is supported, and on this and other operating systems the cross platform implementation Mono is supported, although as of writing only Linux has been tested. rClr is the analogue for .NET to rJava (Urbanek, 2009) for the Java runtime. rClr complements and in part re-uses the already existing R.NET library (Abe, 2013) that makes R programmatically accessible to .NET programmers. The development of rClr is a personal endeavour motivated by work-related needs with complementary use of R and .NET code. Aside from ad-hoc file formats for data exchange there are existing programmatic solutions such as rcom but the underlying COM technology is effectively limited to Windows. Web Service based approaches are also possible and more platform agnostic. Both usually require additions or modification to existing .NET code to enable access from R, and the latter has a prohibitive performance penalty in some scenarios. rClr is designed to let R users access arbitrary .NET code (C#, F#, VB.NET and any other language that targets the CLR) without inherent need for addition or modification to this code. rClr has been used by the author to interactively test the correctness of a continental-scale, gridded spatial-temporal hydrological data assimilation method ported from R to C# to improve the runtime, scalability and integration with other systems. The seamless bi-directional conversion of the most common simpler R data types such as vectors is complete. Short to medium term work will center on the distribution via CRAN, runtime performance optimizations and the interoperability of more complex data types with no obvious or single equivalent in .NET such as data frames and S4 classes.

References

