# Online Applications with Rpad

Tom Short[*], Philippe Grosjean[†]

Rpad (http://www.rpad.org) is an open-source program that provides interactive workbook-type web pages using R as the calculation engine. The R scripts that perform the calculations are embedded in the web page. All required code is embedded in a R package distributed on CRAN. The following topics are discussed:

Server-side functionality. The server uses a modified version of an open-source Perl application called Statistics-R by Graciliano Monteiro Passos. This tool initiates and controls multiple instances of R. A small set of Perl scripts use Statistics-R to pass commands and data from the web server to R. Each webpage gets its own R process and its own temporary directory for reading and writing files. After a definable period of inactivity from the user, the R process shuts down.

Mini-webserver in R using Tcl socket communications. In addition to running Rpad on a dedicated web server, Rpad can also be run from within R using a built-in mini web server inside R. The mini web server is about 500 lines of Tcl code adapted from tclhttpd (http://www.tcl.tk/software/tclhttpd/) and it is used to quickly run a local Rpad server (i.e., no other requirements than to install R and Rpad from CRAN).

Browser-side functionality. Rpad is a type of application based on AJAX technologies. AJAX stands for Asynchronous JavaScript plus XML, but it is used widely to mean any web-based application that dynamically updates portions of the page from the server without refreshing the whole page. When Rpad calculates a page, the browser updates only the results portions of the web page. So, a user can change inputs, hit the calculate button, and the graphs or data or other results update within the existing page. All communication between the browser and the server is plain text. The results from the server are displayed as plain text or as HTML (possibly using functionalities of the R2HTML package). Graphs are normally created as PNG files on the server and passed as an anchor tag to the browser for display within the page. HTML form elements (inputs, checkboxes, and radio buttons) are sent to the server as R variables, which makes it easy to code simple form-type user interfaces.

Browser "editing". Rpad provides different options for the user to view and change the R scripts and other portions of the webpage. Early versions used HTMLArea to provide editing, and the most recent version of Rpad uses the Dojo toolkit to provide editing and user interactivity (http://www.dojotoolkit.org).

Integration of Rpad with a wiki. A Wiki dedicated to R is currently under construction. It will be available before the workshop (probably under an address like http://wiki.r-project.org or http://www.r-project.org/wiki). We plan to integrate Rpad with the Wiki pages to allow interactivity and experimentation on these pages. Two solutions are investigated: (1) direct integration of Rpad in the R Wiki server (or in a secured dedicated Rpad server), and exportation of the wiki pages in Rpad format. These pages can then be edited and viewed locally, using the local mini-webserver in R/Rpad.

Security issues. There is no built in security in Rpad. The user has complete access to any command in R and also to the system shell. For protection, the system needs to be locked down on the server. Write protect any files and databases that are a concern, and lock out access to the server user to other parts of the system. More advanced options to protect against malicious users are to put the server and Rpad components in a chroot jail, use a virtual server to supply Rpad (Xen or VServer), or run the server from a write-protected system disk (like a Quantian DVD).

[*] EPRI Solutions, Inc.,Schenectady, NY, USA, tshort@eprisolutions.com

[†] Numerical Ecology of Aquatic Systems, Mons-Hainaut University, Belgium, phgrosjean@sciviews.org