

RSimulationHelper: Parallel Computing for R Simulations

RSimulationHelper 1.8, September 2017

Introduction

Running extensive simulations in R can be computer intensive. Many computers have multiple processors, making it possible to split a simulation task in many smaller, and hence faster, sub-simulations. In theory, it is not difficult to copy your R simulation code into as many files as necessary to run parallel processing manually, but as the number of repetitions becomes larger, this task becomes increasingly tedious.

RSimulationHelper is a Perl-driven program that facilitates running R simulations in parallel. The program prompts the user for the number of processors to be used, R Code, the typically large number of simulations to be submitted, the list of R objects to be monitored, and then runs these requested simulations in parallel computing. **RSimulationHelper** monitors the evolution of the sub-simulations launched, and begins a sub-simulation every time a processor is free. Delays between completed sub-simulations are thus minimized.

Installation

RSimulationHelper requires that R and ActivePerl be installed on your computer. ActivePerl is a free and easy-to-install package available from <http://activestate.org>

Once Active Perl is installed, you need to install the *Process* Perl library, which also installs easily. At DOS-prompt, type

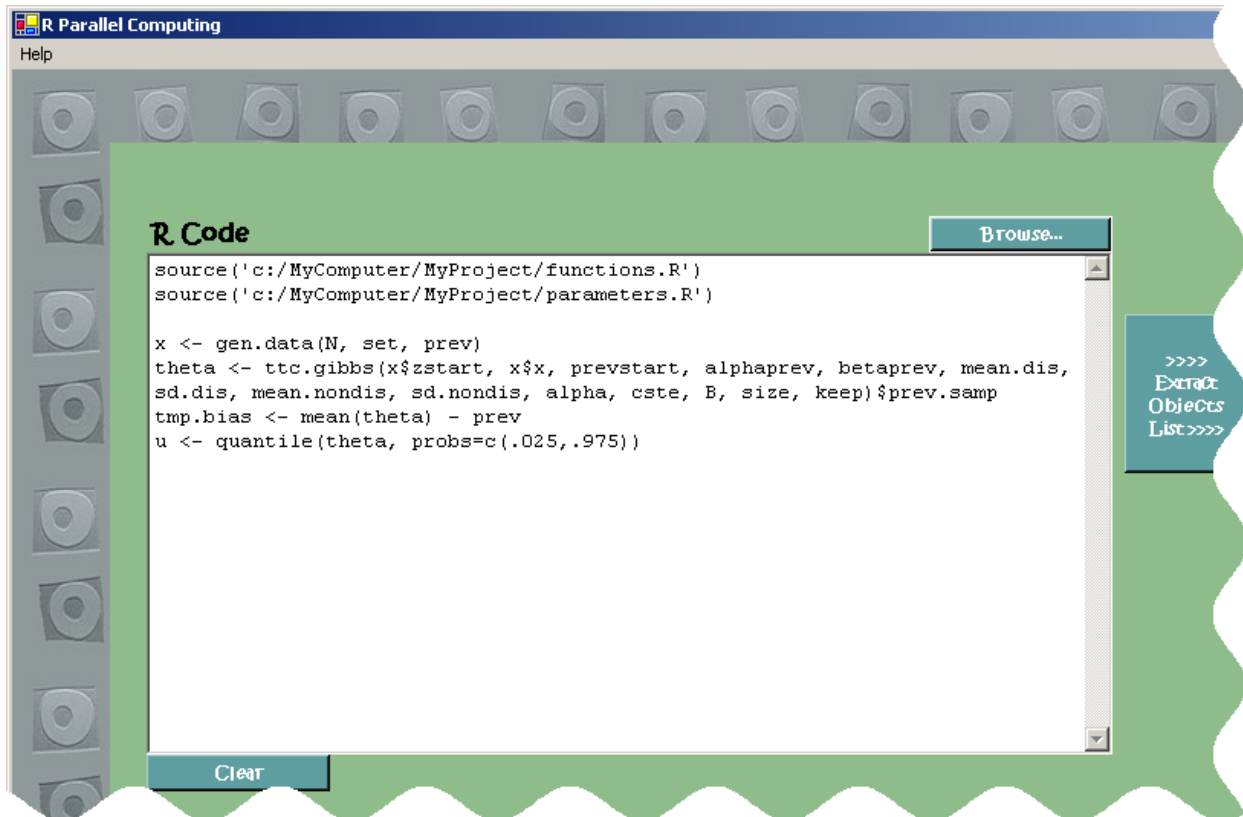
➤ ppm install process

and wait a few seconds for the installation to complete.

RSimulationHelper installs in a few clicks. Once *RSimulationHelper.zip* is downloaded, unzip it, double-click the file *RSimulationHelper.msi* and follow the steps prompted on your screen.

Instructions via an example

Once your simulation R Code is tested and saved to a file, launch **RSimulationHelper**¹ to start your parallel simulations. From the first form, load your R simulation code by clicking the **Browse** button. Your R simulation code will show in the R code window in the left part of the form as pictured below.



Your R code may include a series of R assignments, such as

```
X <- gen.data(N, set, prev)
```

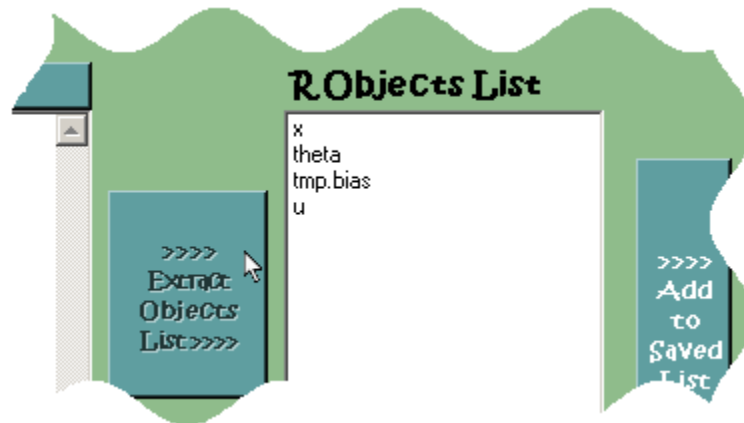
or

```
u <- quantile(theta, probs=c(.025, .975))
```

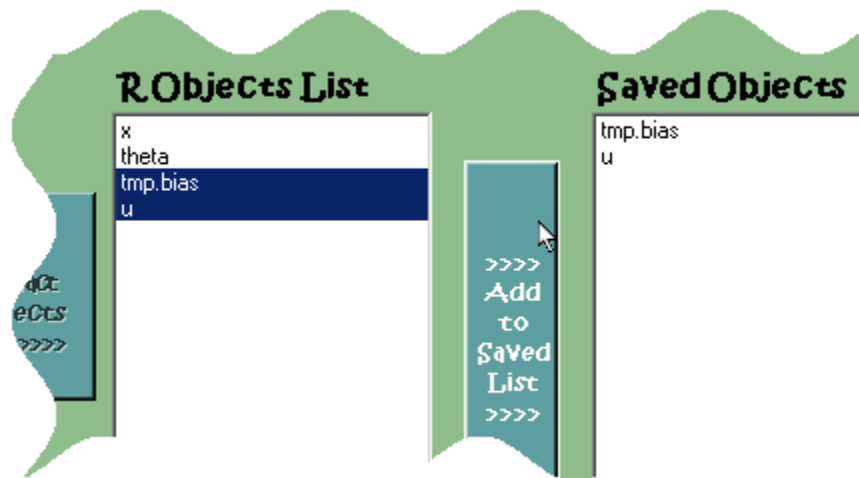
as seen in the above example. Each R object may or may not be of interest to your simulation results. You can therefore choose which objects should be monitored, as shown below.

¹ The easiest way to open this program is to use the shortcut found in Programs list from the Start menu: click the Start button, select **All programs** and pick **RSimulationHelper** from the list. You can also start **RSimulationHelper** by opening Windows Explorer, browsing to this package's location (C:\Users\user name\Documents\RSimulationHelper or C:\Documents and Settings\user name\My Documents\RSimulationHelper by default, depending on your platform) and clicking on *RSimulationHelper.vbs*.

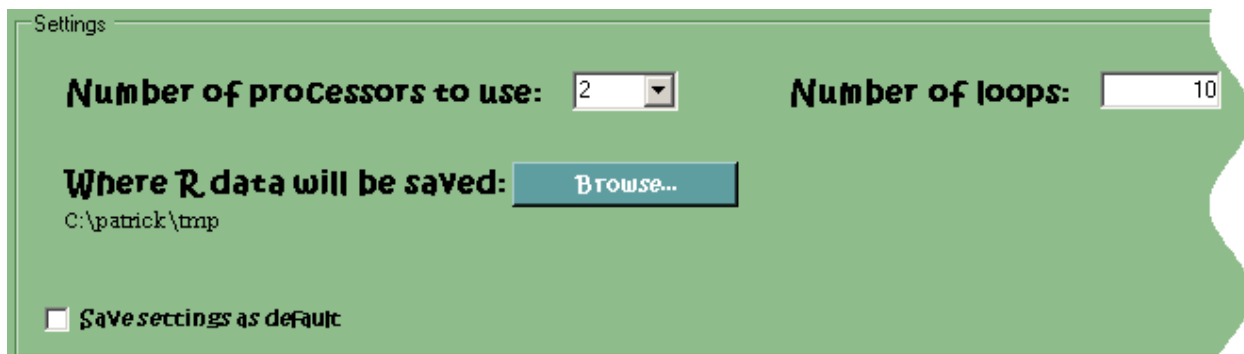
Click the *Extract Objects List* button to the right of the R Code window to create the list of R objects found in your R Code². The list will then show in the next window, labeled *R Objects List*.



Next, select the R Objects from the list that you want to monitor. These objects will be saved to a .RData image file at the end of each sub-simulation run. Select each R Object to be monitored from the list and click the button to the right, labeled *Add to Saved List*. Selected R Objects will then show in *Saved Objects* window.



There are a few last bits of information necessary before simulation can start. In the *Settings* box at the bottom of the form, choose the directory where each sub-simulation result will be saved by clicking on the *Browse...* button to the right of *Where R data will be saved*.



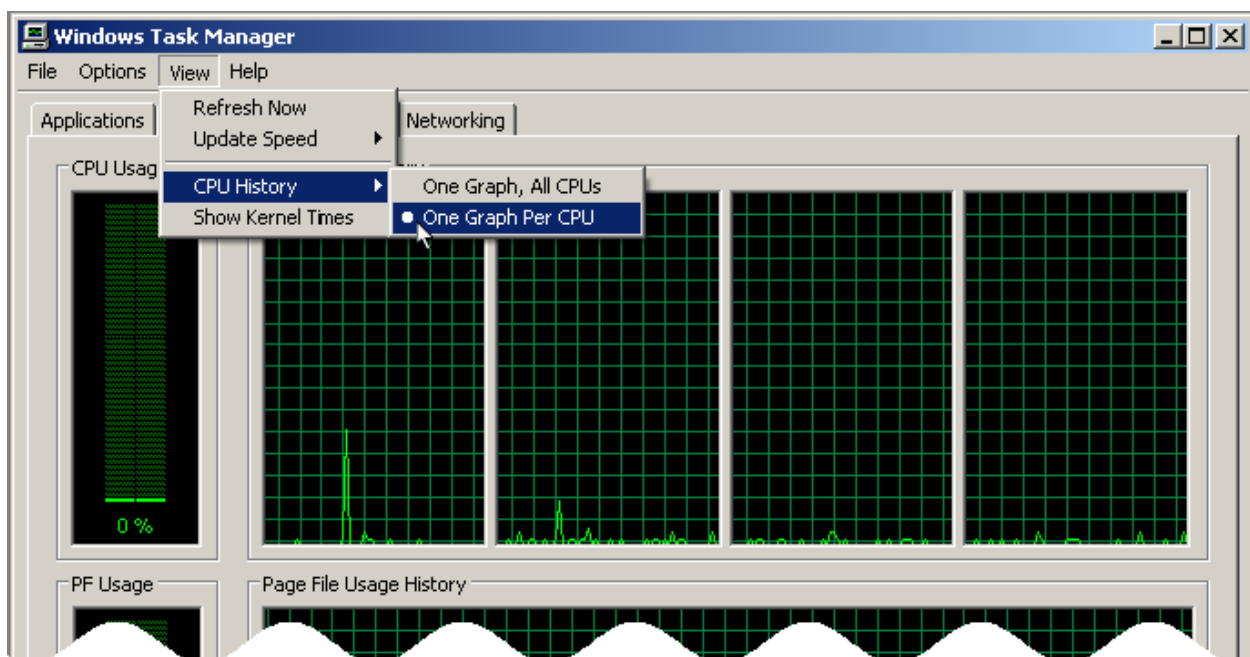
² Only the tokens appearing at the beginning of coding lines are recognized as *monitorable* R Objects by **RSimulationHelper**, provided they are followed by an equal or the R assignment arrow sign (<-).

Select the number of times you want your R Code to be run (*Number of loops* text box) and the number of processors on which you want to run individual sub-simulations. If you are not sure about the number of processors you have, please refer to the next section. We strongly recommend that you keep at least one processor free for your other work while **RSimulationHelper** runs. Otherwise, you may find your computer lacks free resources for your other needs until **RSimulationHelper** finishes.

These settings can be saved as the default for next time you use **RSimulationHelper** by clicking the tick box in the lower left corner of the *Settings* section. Click the *Next* button when you are done.

How many processors do I have?

If you are unsure of how many processors are available for parallel processing on your computer, open the **Windows Task Manager** and click the **Performance** tab: the number of isolated blocks in the **CPU Usage History** section corresponds to the number of processors available. In the example illustrated below, the **Windows Task Manager** shows 4 processors.



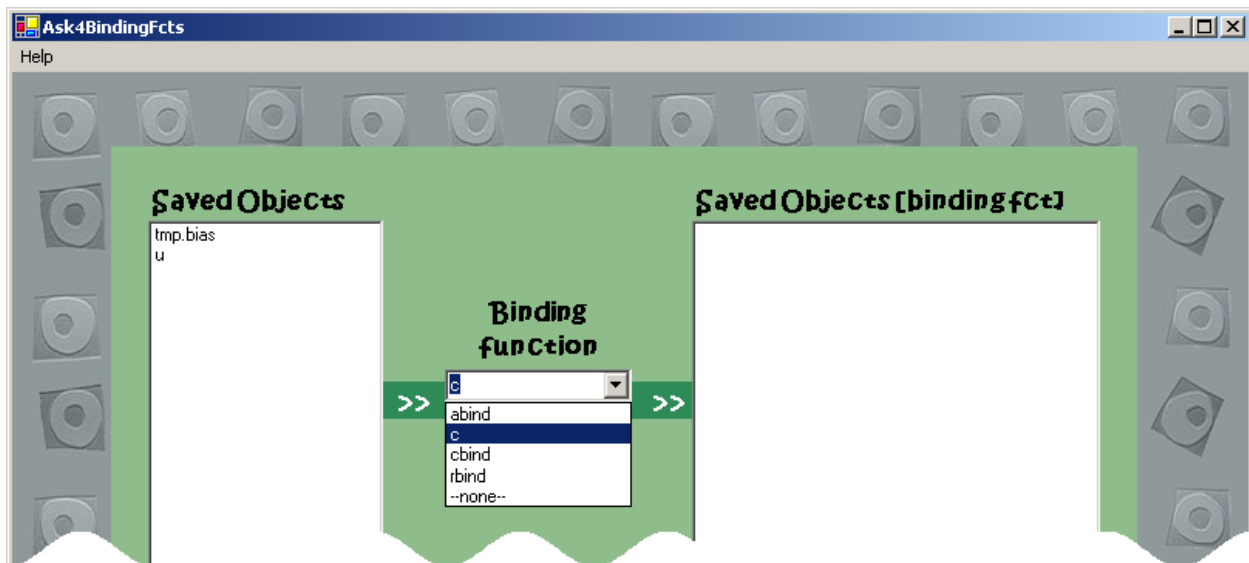
Binding simulation results

Upon completion, **RSimulationHelper** saves the monitored R Objects values to separate .RData files. Although you can write your own R code to gather the simulation results, **RSimulationHelper**'s second form offers to pick from a series of binding functions that you can use to merge the individual sub-simulation results. Each monitored R Object can be assigned a different binding function.

Select a series of R Objects from the *Save Objects* window and select the appropriate binding function from the *Binding function* list box. You can type alternative binding function names in the corresponding text box if necessary. Click the double arrows (in the dark green bar underneath the *Binding function* box) to validate your new function.

If you assign the *--none--* item (from the list of binding functions) to one or more R Objects, these objects will still be monitored — that is, saved to .RData files — but they will not appear in the R binding code.

If you do not need **RSimulationHelper** help in writing the binding R Code, simply click the *Ignore (I'll write the binding code myself)* button in the lower left part of the form.



Once each node in the *Save Objects* list is assigned a binding function, a *Next>>* button will appear in the right lower corner of the form. Click it when you are done.

If you choose to let **RSimulationHelper** write your binding R code, a last form will provide you with this R code. Save it in a safe place for future use. The actual simulations will start immediately after that form is closed.

Upon completion of all simulations, **RSimulationHelper** will open a final form. This form provides the total run time, a link to the directory where you have asked the .RData files to be saved, and, if you have saved **RSimulationHelper** suggested R code for results recovery, a link to the file where you have saved that code.



Change log

Versions 1.1 - 1.1.2 (December 2011) The previous default application folder --- c:\Program Files --- caused write permission problems for some Windows 7 and Vista users. Default application folder now changed to C:\Users\user name\Documents.

Versions 1.2 -- 1.3.1 (February 2012) Minor technical problem solved from previous version.

Version 1.4 (June 2012) Saved R objects are forced to have .RData extension.

Version 1.5 (July 2012) Minor update: cmd.exe now closes automatically when program terminates.

Version 1.6 (August 2012) The path to the sub-directory where temporary files are stored was added to the Help menu of the initial form. While you can usually ignore these files, they can sometimes be helpful in troubleshooting when there are problems.

Version 1.7 (January 2015) Minor update.

Versions 1.7.1 and 1.7.2 (April 2015) Minor bug fix update: a potential installation problem was fixed.

Version 1.8 (September 2017) **RSimulationHelper** now works on Windows 8 & 10. Windows 7 users do not need to reinstall or upgrade.

Questions? Comments?

Any questions or comments relating to **RSimulationHelper** can be sent to Patrick Bélisle, patrick.belisle@rimuhc.ca

<http://www.medicine.mcgill.ca/epidemiology/Joseph/PBelisle/>