

HPC with R Tutorial Prerequisites

Drew Schmidt

Attendee Background

The attendee should ideally have some familiarity working with R. Service providers looking for information to better support R users need not have direct experience with R, provided they have a strong programming background, and some familiarity with a high-level language such as Python or Matlab.

Software Requirements

There is no “hands-on” portion for the tutorial, but there will be “take home” exercises. To complete these or to optionally follow along with examples presented during the workshop, you will need to install some software.

R

In order to complete the exercises, you will need to have R installed. You can find R binaries and/or source at <http://cran.r-project.org>. Installation for Windows and Mac users should be very much like what you are used to. Linux and *BSD users probably have access to R in their OS’s software repository; you can find additional information at the above download link.

If for some reason you wish to build R from source, consult the R Installation and Administration manual <https://cran.r-project.org/doc/manuals/R-admin.html>.

RStudio

R is supported in just about every text editor and multi-language IDE these days. However, much of the R community uses the RStudio Desktop IDE <http://www.rstudio.com/products/rstudio/download/>. It has binaries for Windows, Mac, and Linux. You can even install a server version running on a remote instance, and it looks/behaves much the same as the desktop version.

RStudio is probably the most popular editor used in the R community, so even if you prefer another editor/ide, it is still useful to get a sense for the tools that R users find most appealing. Installing RStudio is optional. But if you install it, you still need to install R separately as explained above.

MPI (Optional)

If you wish to experiment with using MPI with R, then you will need a system installation of MPI.

- Windows: MS-MPI [https://msdn.microsoft.com/en-us/library/bb524831\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/bb524831(v=vs.85).aspx)
- Install Open MPI; we are not Mac users so there may be a better approach, but we generally use MacPorts for this. See section 3.2 of [this document](#) for more details.
- Linux/FreeBSD: install Open MPI from your package manager.

Packages

The lab will use both in instruction and in exercises several package extensions. To install a package named `pkg`, start up an R session and enter

```
install.packages("pkg")
```

This will automatically handle dependency resolution for you.

Two packages which we will highlight during the lab which you are strongly recommended to install are: * `rbenchmark` * `microbenchmark`

If you wish to experiment with (mostly) shared-memory parallelism in R, you might want to install: * `foreach` * `doParallel` * `snow`

We will also briefly discuss using MPI with R. If you have MPI installed (see above), the main packages of interest are: * `Rmpi` * `pbDMPI` * `pbDMAT`

Compilers (Optional)

Finally, if you wish to be able to build packages from source, then you need to install some additional things. This is not strictly necessary, although there is a very active community of R developers on GitHub, who do not submit their packages to CRAN (which distributes binaries for Windows+Mac). This is optional for this lab, though encouraged if you use R regularly.

- Windows: Install Rtools <http://cran.r-project.org/bin/windows/Rtools/>
- Mac:
 - First install XCode from the Mac App Store
 - Then, open XCode and choose “Preferences”, “Downloads”, then “Install Command Line Tools”.
- Linux/FreeBSD: You need C and Fortran compilers, which you may well already have installed. Consult your distribution’s package manager.