Setting Up a **pbdR** Environment

Installing MPI, R, and **pbdR**

Version 1.0

Prepared by the pbdR Core Team:

Drew Schmidt
Remote Data Analysis and Visualization Center,
University of Tennessee, Knoxville

Wei-Chen Chen
Computer Science and Mathematics Division,
Oak Ridge National Laboratory

Pragneskumar Patel
Remote Data Analysis and Visualization Center,
University of Tennessee, Knoxville

George Ostrouchov
Computer Science and Mathematics Division,
Oak Ridge National Laboratory

This publication was typeset using LaTeX.

Contents

1   Allocation

We have tried to make the installation process as simple as possible, and these instructions as thorough as possible. However, this is not an entirely labor-free procedure, and does not even get into the really difficult side of large-scale computing: managing the system.

If you affiliated with a United States institution and are engaged in research that requires large-scale computing resources, we encourage you to consider getting an allocation with us . Not only can we tailor our pbdR development to help your research, but we can manage the hardware, operating system, and software utilities for you, so that you can focus entirely on the thing that matters, your research.

2   Quick Introduction

In this guide, we will detail the necessary steps for how to set up a pbdR environment. What follows in the remaining sections is a very lengthy list of installation instructions; however, most users should find the process fairly straight-forward, and may not need (or want) all of the details we will provide unless something goes wrong. In any case, the short version for setting up a pbdR environment is to:

1. install R (and Rtools for Windows); see http://cran.r-project.org/

2. install an MPI library; http://www.open-mpi.org/, or http://www.mpich.org/ for Windows

3. install the pbdR packages; see http://r-pbd.org/

Items 1 and 2 are interchangeable, and so if you already have R (and additionally Rtools for Windows) and/or an MPI library installed, then merely skip this/these step(s); there is no need to reinstall anything.

2.1   Installing R

This should be fairly painless. R has binary packages for every operating system you have heard of (and some you haven't), and the install should go fine. Of course, since R is open source, you are free to compile it yourself, should have have reason or need to do so. You can find both the source as well as binaries at the R project's main site: http://cran.r-project.org/.

Additionally, you may wish to customize your R build by compiling from source. For example, you may wish to link R with a high performance linear algebra library, such as MKL. See the *R Installation and Administration Manual* at http://cran.r-project.org/doc/manuals/R-admin.html for full details.

2.2   Installing MPI

For Linux and Mac users, we recommend installing OpenMPI, which is available from http://www.open-mpi.org/ in both binary and source formats. Windows users should install MPICH2, available from http://www.mpich.org/ .

2.3   Installing pbdR Packages

All released pbdR packages are available from http://cran.r-project.org/ which is the Comprehensive R Archive Network (CRAN). This is similar to the CPAN for perl or CTAN for LaTeX, although with many improvements and benefits over its competitors.

It is also possible to link pbdR with high performance linear algebra libraries, such as MKL. Figure 1
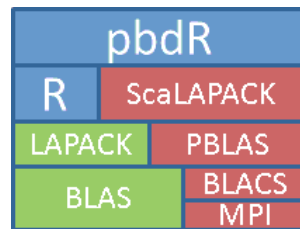
Figure 1: pbdR Relationships to Libraries

offers some insight into the package organization. See the pbdSLAP vignette for more details.

3   Windows

Officially, the pbdR team does not support gaming consoles (only kidding!). Jokes aside, it is possible to install a pbdR environment on Windows, but it is not necessarily the easiest. This guide will explain the basics of getting R, OpenMPI, and pbdR installed on your Windows system. The instructions and screenshots for this document are for version 2.15.1 of R, but later versions should be very similar, if not identical.

If you are completely new to R, then you may find the R for Windows FAQ useful. Additionally, there is also an R FAQ which may also be useful for those who know very little about R. To learn more about programming with R, then you may find the Introduction to R guide useful.

3.1   Installing R

1. Download R: http://cran.r-project.org/bin/windows/base/

2. Open the saved file from 1 above to begin the installation. At the first setup screen, click 'Next' to continue.

3. When prompted with the license, click 'Next' to continue.

4. When prompted for the location to install R, we strongly encourage you to use the default. When you have made your decision, click 'Next'.

5. When prompted with the components to install, you should select a 'User installation'. Then click 'Next'.

6. When prompted with the option to alter the startup options, we suggest selecting `No (accept defaults)`. When you have made your decision, click 'Next'.

7. When prompted with the start menu folder options, make your choice and then click 'Next'.

8. When prompted with the additional tasks options, we suggest making sure that `Save version number in registry` and `Associate R with .RData files` are both **checked**. When you have made your decisions, click 'Next'.

9. To complete the R installation, select 'Finish'.

Once R is finished installing, you need to install the **rlecuyer** package. To install it from an interactive R session, simply start an R session and issue the command

```
1  install.packages("rlecuyer")
```

3.2   Installing Rtools

1. Download Rtools: http://cran.r-project.org/bin/windows/base/

2. Open the saved file from 1 above to begin the installation. At the first setup screen, click 'Next' to continue.

3. When prompted with the license, click 'Next' to continue.

4. When prompted for the location to install R, we strongly encourage you to use the default. When you have made your decision, click 'Next'.

5. When prompted with the components to install, you should select a 'User installation'. Then click 'Next'.

6. When prompted with the option to alter the startup options, we suggest selecting `No (accept defaults)`. When you have made your decision, click 'Next'.

7. When prompted with the start menu folder options, make your choice and then click 'Next'.

8. To complete the Rtools installation, select 'Finish'.

3.3   Installing MPI

Before proceeding, please be aware that this installation requires administrative privileges.

1. Download both the 32-bit mpich2-1.4.1p1-win-i32.msi and the 64-bit mpich2-1.4.1p1-win-x86-64.msi installers from: http://www.mpich.org/ .

2. Open the saved file from 1 above to begin the installation. At the first setup screen, click 'Next' to continue.

3. When prompted with the system requirements, click 'Next' to continue.

4. When prompted with the license, click "I agree" and then click 'Next' to continue.

5. When prompted with the 'Process Manager setup' screen, choose a passphrase and then click 'Next' to continue.

6. At the 'Select Installation Folder' screen, we recommend you keep the default folders. By default, the 32-bit application will be installed in `C:\Program Files (x86)\MPICH2\` and the 64bit application will be installed in `C:\Program Files\MPICH2\`.

   Additionally, you should make the selection for whether or not MPICH2 should be available to all users of this system or just yourself. When you are ready to proceed, click 'Next'.

7. When prompted to confirm the installation, click 'Next' to proceed.

8. To complete the MPICH2 installation, select 'Close'.

3.4   Installing pbdR Packages

Unfortunately, we do not distribute pbdR binary packages on the CRAN for Windows. This means that you must install our packages from source on Windows, and the process may be foreign. We will present two approaches; the short way, installing from github using the **devtools** package, and a longer way, installing from a downloaded source file. However, do be aware that each of these methods requires the installation of the Rtools package from Section 3.2 (so that step cannot be skipped).

3.4.1   Installing from Github

This is probably the simplest method, assuming that you have Rtools installed and set up correctly. If Rtools is not in your PATH, then you may need to enter something like the following:

```
rtools <- "C:\\Rtools\\bin\\"
mingw <- "C:\\Rtools\\gcc-4.6.3\\bin\\"

PATH <- Sys.getenv("PATH")
new.PATH <- paste(rtools, mingw, PATH, sep = ";")
Sys.setenv(PATH=new.PATH)
```

Where the `rtools` and `mingw` directories are as they are on your machine.

Once that is settled, installing is fairly simple. You simply load the **devtools** package and install from our github repo as follows:

```
library(devtools)

install_github(repo="pbdMPI", username="RBigData")
install_github(repo="pbdSLAP", username="RBigData")
install_github(repo="pbdNCDF4", username="RBigData")
install_github(repo="pbdBASE", username="RBigData")
install_github(repo="pbdDMAT", username="RBigData")
install_github(repo="pbdDEMO", username="RBigData")
```

You can also install *really* new package builds, which will be very current in terms of features, but also bugs (or even complete package breakage). If you're sure you want these packages, then you can install them as follows:

```
# dev repo 1
install_github(repo="pbdMPI", username="snoweye")
install_github(repo="pbdSLAP", username="snoweye")
install_github(repo="pbdNCDF4", username="snoweye")
# dev repo 2
install_github(repo="pbdBASE", username="wrathematics")
install_github(repo="pbdDMAT", username="wrathematics")
install_github(repo="pbdDEMO", username="wrathematics")
```

3.4.2   Installing from Downloaded Source

1. To help simplify things, we offer a simple install script which makes managing the `PATH` slightly simpler. By default, this is for R version 2.15.3 and **pbdMPI** version 0.1-6. If you have a different version of R or want to install a different version of **pbdMPI**, simply change lines 4 and/or 5 of the install script to match your needs.

2. Open a command prompt either from the start menu, or by entering the command `Windows key` then `R`, and then entering `cmd` in the "run" dialog box.

3. Before proceeding further, it is probably a good idea to test the installation of MPICH2. The following commands utilize MPICH2 to do some fairly trivial things (do not copy `SHELL>` if you are copying and pasting from the lines below):

```
SHELL > C:\PROGRA~1\MPICH2\bin\mpiexec -np 2 hostname.exe
SHELL > C:\PROGRA~1\R\R-2.15.1\bin\R --vanilla --slave -e "ls()"
SHELL > C:\PROGRA~1\R\R-2.15.1\bin\Rscript --vanilla --slave -e "ls
    ()"
```

4. Download the source for **pbdMPI** package. Put it in the same directory as your `build_pbdMPI.bat` file, and then execute `build_pbdMPI.bat` from the command prompt to install the package.

5. If the process is done without errors, you can see the binary package is installed.

4   Mac OS X

Before starting, make sure you have installed Apple's XCode package. You can find this in the Mac App Store. Additionally, during the course of installation, you may need to use the terminal once or twice. You can launch it from Finder by navigating to the Applications folder; it should be called Terminal.app. If you have never used the terminal before, you might consider skimming this simple guide on terminal basics.

If you are completely new to R, then you might consider reading (or at least skimming) this useful guide, R for Mac OS X FAQ. There is also the R FAQ which may also be useful for those who know very little about R. To learn more about programming with R, then you may find the Introduction to R guide useful.

4.1   Installing R

You can install R either from the binary package that CRAN builds (recommended) or from source.

4.1.1   Installing from a Binary Package

1. First you should download R from the official distribution site: http://cran.r-project.org/bin/macosx/ Download the R dmg package for Mac OS X. We recommend grabbing the latest version of R available.

2. Open the saved file from 1 above to begin the installation. At the first setup screen, click 'Continue' to begin the installation process.

3. At the 'Read Me' section, read the important information and then click 'Continue'.

4. When prompted with the software license agreement, you must click 'Agree' to proceed. R is distributed under a free, "copyleft" license (GPL v3). You can read the license by clicking 'Read License'. Once you agree to the terms, click 'Agree'.

5. From the 'Installation Type' section, we recommend you use the defaults. However, you may change the install location by selecting 'Change Install Location...'. Once you have made your choice, click 'Install'.

6. Next, you will be prompted for your computer's name and password. Enter the appropriate information and click 'Install Software' to begin the installation process.

7. Once the installation process begins, wait a few moments for the packages to validate and install.

8. Once the installation completes successfully, click 'Close' to finish the installation process.

Out of the box, you can run an interactive R session in two separate ways: from the "gui" and from a terminal. To launch the R "gui", simply go to your Applications folder in Finder and double-click R.app.

Once R is finished installing, you need to install the **rlecuyer** package. To install it from an interactive R session, simply start an R session and issue the command

```
install.packages("rlecuyer")
```

4.1.2   Compiling from Source

You can find R sources from http://cran.r-project.org/sources.html

Start by opening a terminal and navigate to the folder containing the R source package you just downloaded. You can extract the archive by executing, for example

```
tar zxvf R-3.0.0.tar.gz
```

From here, generally it should be enough to simply execute

```
./configure && make && make install
```

without problems.

4.2   Installing MPI

You have several options for installing OpenMPI on a Mac. You can install from MacPorts, which is a relatively simple way to manage compiling/installing of many packages (such as OpenMPI). You can also compile from source.

Before beginning, make sure you have some "downtime" allotted, as the compilation will take upwards of a few hours for some machines.

4.2.1   Installing from a MacPorts

Arguably the easiest way to install OpenMPI for a Mac (that I'm aware of) is using MacPorts, which is not unlike some source repositories for Linux. You can find information about installing MacPorts from MacPorts installation page, maintained by the MacPorts project.

Once you have MacPorts installed, you can install openmpi from a terminal by issuing the command:

```
sudo port install openmpi
```

4.2.2   Compiling from Source

If you want to install OpenMPI from source (I don't really recommend this unless you think you have a good reason to), then the sources are available here.

4.3   Installing pbdR Packages

Installing pbdR should go smoothly. The simplest way to install the packages is from an R terminal, which will manage dependencies for you much like the Mac App store.

4.3.1   Installing from CRAN

This is perhaps the simplest way to proceed, as R will handle any package dependency resolution for you. Simply start an R session (from the terminal[1], type R then press enter) and issue the command:

```
1  install.packages(<package>)
```

So for example, to install **pbdMPI**, you might execute:

```
1  install.packages(pbdMPI)
```

4.3.2   Installing from the Shell

If you have downloaded a pbdR (or other R) package, then installing from the shell simply amounts to issuing the command:

```
R CMD INSTALL <package>
```

So for example, to install **pbdMPI**, you might execute:

```
R CMD INSTALL pbdMPI_0.1-6.tar.gz
```

4.3.3   Installing from Github

CRAN policy is such that updates to packages can not be made too frequently. For this reason, the development versions of our packages will have bugfixes and new features much more quickly than CRAN versions.

The easiest way to install from github is using Hadley Wichkam's **devtools** package (which can be installed via `install.packages(devtools)`). Assuming you have this package installed, then from an R session, to install a pbdR package you would issue one of the following:

```
1  library(devtools)
2
3  install_github(repo="pbdMPI", username="RBigData")
4  install_github(repo="pbdSLAP", username="RBigData")
5  install_github(repo="pbdNCDF4", username="RBigData")
6  install_github(repo="pbdBASE", username="RBigData")
7  install_github(repo="pbdDMAT", username="RBigData")
8  install_github(repo="pbdDEMO", username="RBigData")
```

---

[1]Do *not* use the gui. See section 7.2 for details

You can also install *really* new package builds, which will be very current in terms of features, but also bugs (or even complete package breakage). If you're sure you want these packages, then you can install them as follows:

```
1  # dev repo 1
2  install_github(repo="pbdMPI", username="snoweye")
3  install_github(repo="pbdSLAP", username="snoweye")
4  install_github(repo="pbdNCDF4", username="snoweye")
5  # dev repo 2
6  install_github(repo="pbdBASE", username="wrathematics")
7  install_github(repo="pbdDMAT", username="wrathematics")
8  install_github(repo="pbdDEMO", username="wrathematics")
```

5   Linux and FreeBSD

Before starting, you may need root access to your machine. Also, you will need to know how to do some simple things via the terminal. If you're using a standard Linux desktop, you probably have a terminal launcher in your applications menu somewhere. If you're using some kind of weirdo tiling thing from 1990, then I assume you know what you're doing. Additionally, if you are inexperienced with using the terminal, you should consider skimming this short introduction.

On Linux, unless you have a specific reason not to (in which case, most of this document is probably unnecessary for you), we recommend that you install R and MPI through your distribution's package repository (especially MPI). This will make the installation process *much* simpler, and generally "just works".

If instructions for your favorite distribution are not listed below, we would be happy to incorporate submissions/corrections.

Finally, if you are completely new to R, then you might consider reading the R FAQ. To learn more about programming with R, then you may find the Introduction to R guide useful.

5.1   Installing R

You can install R either from your package repo (recommended) or from source.

5.1.1   Installing from a Package Repository

If your distribution is Debian-derived, including Debian, Ubuntu, and Mint:

```
apt-get install r-base-dev
```

If your distribution is "Redhat-ish", including Redhat, Fedora, and CentOS:

```
yum install R-devel
```

If your distribution is OpenSUSE:

```
zypper install R-patched-devel
```

If you are using FreeBSD:

```
cd /usr/ports/math/R && make install clean
```

5.1.2   Compiling from Source

You can find R sources from <http://cran.r-project.org/sources.html>

Start by opening a terminal and navigate to the folder containing the R source package you just downloaded. You can extract the archive by executing, for example

```
tar zxvf R-3.0.0.tar.gz
```

From here, generally it should be enough to simply execute

```
./configure && make && make install
```

without problems.

5.2   Installing MPI

You can install R either from your package repo (recommended) or from source.

5.2.1   Installing from a Package Repository

For these systems, we recommend using OpenMPI. To install OpenMPI

If your distribution is Debian-derived, including Debian, Ubuntu, and Mint:

```
apt-get install openmpi-bin libopenmpi-dev
```

If your distribution is "Redhat-ish", including Fedora and CentOS:

```
yum install openmpi openmpi-devel
```

If your distribution is OpenSUSE:

```
zypper install openmpi-devel lam-devel
```

If you are using FreeBSD:

```
cd /usr/ports/net/openmpi && make install clean
```

5.2.2   Compiling from Source

If you want to install OpenMPI from source (I don't really recommend this unless you think you have a good reason to), then the sources are available here.

5.3   Installing pbdR Packages

Installing pbdR should go smoothly. The simplest way to install the packages is from an R terminal, which will manage dependencies for you much like your distro's package manager. Additionally, our packages are available in the Fedora repositories.

5.3.1   Installing from CRAN

This is perhaps the simplest way to proceed, as R will handle any package dependency resolution for you. Simply start an R session (from the terminal, type R then press enter) and issue the command:

```
install.packages(<package>)
```

So for example, to install **pbdMPI**, you might execute:

```
install.packages(pbdMPI)
```

5.3.2   Installing from the Shell

If you have downloaded a pbdR (or other R) package, then installing from the shell simply amounts to issuing the command:

```
R CMD INSTALL <package>
```

So for example, to install **pbdMPI**, you might execute:

```
R CMD INSTALL pbdMPI_0.1-6.tar.gz
```

5.3.3   Installing from Github

CRAN policy is such that updates to packages can not be made too frequently. For this reason, the development versions of our packages will have bugfixes and new features much more quickly than CRAN versions.

The easiest way to install from github is using Hadley Wichkam's **devtools** package (which can be installed via `install.packages(devtools)`). Assuming you have this package installed, then from an R session, to install a pbdR package you would issue one of the following:

```
library(devtools)

install_github(repo="pbdMPI", username="RBigData")
install_github(repo="pbdSLAP", username="RBigData")
install_github(repo="pbdNCDF4", username="RBigData")
install_github(repo="pbdBASE", username="RBigData")
install_github(repo="pbdDMAT", username="RBigData")
install_github(repo="pbdDEMO", username="RBigData")
```

You can also install *really* new package builds, which will be very current in terms of features, but also bugs (or even complete package breakage). If you're sure you want these packages, then you can install them as follows:

```
# dev repo 1
install_github(repo="pbdMPI", username="snoweye")
install_github(repo="pbdSLAP", username="snoweye")
```

```
4  install_github(repo="pbdNCDF4", username="snoweye")
5  # dev repo 2
6  install_github(repo="pbdBASE", username="wrathematics")
7  install_github(repo="pbdDMAT", username="wrathematics")
8  install_github(repo="pbdDEMO", username="wrathematics")
```

6   Running pbdR Scripts

This information is covered in *much* more detail in the pbdDEMO vignette, and should not be considered a substitute. However, there are two key points one needs to understand in order to use pbdR tools. Namely,

- pbdR codes are written in Single Program/Multiple Data style

- pbdR codes are executed in batch

For full details, see the pbdDEMO package vignette.

Below is a simple pbdR script. This will help you know if things are installed properly or not. To understand what the script is doing, or to learn how to do much more substantial things, you should see the pbdDEMO package vignette.

```
1  library(pbdMPI, quiet = TRUE)
2  init()
3
4  x <- comm.rank()
5
6  comm.print(x, all.rank = TRUE)
7
8  finalize()
```

To run the script, you must do so in batch (i.e., non-interactively). First save its contents to the file `my_script.r`, and then open a terminal/command prompt. On Linux and Mac, you should execute the command:

mpirun −np 2 Rscript my_script.r

On Windows, you should execute the command:

mpiexec.exe −np 2 Rscript my_script.r

7   Installation Problems

During the course of installation, you may run into unrecoverable issues. The pbdR team does not support MPI libraries or R core, so if you have problems during that portion of the installation phase, we probably can not directly help you. However, there are still many great resources at your disposal, maintained by those individual projects.

7.1   R and MPI

If you have problems installing or customizing R, see the *R Installation and Administration Manual* at http://cran.r-project.org/doc/manuals/R-admin.html for help.

If you are having trouble installing an MPI library, you should see that library's official documentation. For OpenMPI, see http://www.open-mpi.org/community/help/ and for MPICH, see http://www.mpich.org/documentation/guides/ .

For the remainder, we will be addressing installation issues with pbdR packages.

7.2   pbdR

This is a quick list of potential problems you could encounter when installing pbdR packages. For additional troubleshooting or installation options, each package has a vignette which may offer additional useful information.

- **When compiling pbdMPI from source**, you may be required to pass a configure argument at compile time. So for example, if you have OpenMPI installed and were installing from the command line, then you would issue the command:

  R CMD INSTALL pbdMPI_0.1−6.tar.gz \
          −−configure−args='−−with−Rmpi−type=OPENMPI'

  or if installing from R:

```r
install.packages("pbdMPI", configure.args='--with-Rmpi-type=
    OPENMPI')
```

  See the **pbdMPI** vignette for more details.

- **If you are installing on a cluster** where you must install on the login node which can not execute `mpirun`, then pass the install option `--no-test-load`. So for example, if installing from the command line, then you would issue the command:

  R CMD INSTALL pbdMPI_0.1−6.tar.gz −−no−test−load

  or if installing from R:

```r
install.packages("pbdMPI", INSTALL_opts='--no-test-load')
```

- **If you are installing binaries on MAC OS X**, do not use the gui. You can install from source using the gui, or you can install binaries (or from source) using the terminal. But you can not install binaries using the gui. So if you want to install binaries, you should open Finder, then navigate to `Applications/Utilities/` and select `Terminal`. Next, type `R` and press enter. Now try to install the packages.