

Setting Up a pbdR Environment

Installing MPI, R, and pbdR

Version 2.0

Prepared by the pbdR Core Team:

Drew Schmidt

*Remote Data Analysis and Visualization Center,
University of Tennessee, Knoxville*

Wei-Chen Chen

*Department of Ecology and Evolutionary Biology,
University of Tennessee, Knoxville*

Pragneskumar Patel

*Remote Data Analysis and Visualization Center,
University of Tennessee, Knoxville*

George Ostrouchov

*Computer Science and Mathematics Division,
Oak Ridge National Laboratory*

© 2013-2014 pbdR Core Team. All rights reserved.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the pbdR Core Team.

This publication was typeset using L^AT_EX.

Contents

1	Allocation	1
2	Quick Introduction	1
2.1	Installing R	1
2.2	Installing MPI	1
2.3	Installing pbdR Packages	1
3	Linux and FreeBSD	2
3.1	Installing R	2
3.1.1	Installing from a Package Repository	2
3.1.2	Compiling from Source	3
3.2	Installing MPI	3
3.2.1	Installing from a Package Repository	3
3.2.2	Compiling from Source	3
3.3	Installing pbdR Packages	3
3.3.1	Installing from CRAN	4
3.3.2	Installing from the Shell	4
3.3.3	Installing from Github	4
4	Running pbdR Scripts	5
5	Installation Problems	5
5.1	R and MPI	5
5.2	pbdR	6

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 ; 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/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.

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 L^AT_EX, 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 offers some insight into the package organization. See the pbdSLAP vignette for more details.

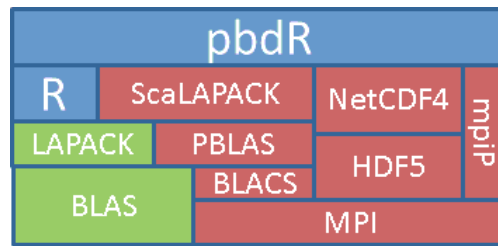


Figure 1: pbdR Relationships to Libraries

3 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.

3.1 Installing R

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

3.1.1 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
```

3.1.2 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.

3.2 Installing MPI

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

3.2.1 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
```

3.2.2 Compiling from Source

If you want to install OpenMPI from source (I don’t really recommend this unless this document is irrelevant to you in the first place), then the sources are available here: <http://www.open-mpi.org/software/mpi/v1.6/> .

3.3 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.

3.3.1 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:

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

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

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

3.3.2 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
```

3.3.3 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="pbdNCDF4", username="RBigData")
7 install_github(repo="pbdBASE", username="RBigData")
8 install_github(repo="pbdDMAT", username="RBigData")
9 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:

```
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="SEXPtools", username="wrathematics")
7 install_github(repo="pbdBASE", username="wrathematics")
8 install_github(repo="pbdDMAT", username="wrathematics")
9 install_github(repo="pbdDEMO", username="wrathematics")
```

4 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. On Linux, you should execute the command:

```
mpirun -np 2 Rscript my_script.r
```

5 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.

5.1 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.

5.2 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-mpi-type=OPENMPI'
```

or if installing from R:

```
1 install.packages("pbdMPI", configure.args='--with-mpi-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:

```
1 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.