A Quick Guide for the pbdSLAP Package

Wei-Chen Chen\textsuperscript{1}, Drew Schmidt\textsuperscript{1}, George Ostrouchov\textsuperscript{2,3}, Pragneshkumar Patel\textsuperscript{2}

\textsuperscript{1}pbdR Core Team

\textsuperscript{2}National Institute for Computational Sciences, University of Tennessee, Knoxville, TN, USA

\textsuperscript{3}Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA

Contents

Acknowledgement ii

1. Quick Start 1

1.1. System Requirements ........................................... 1

1.2. Installation and Quick Start ................................... 1

2. Useful Information 2

2.1. \texttt{CDEFS} Flag of ScaLAPACK ............................. 2

2.2. Using Other Distributions of ScaLAPACK ....................... 2

3. Inside pbdSLAP 3

3.1. Linking with \texttt{pbdSLAP} ................................... 3

4. Testing BLACS ..................................................... 3

References 6

© 2012-2016 pbdR Core Team.

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

This publication was typeset using \LaTeX.
Acknowledgement

Chen was supported in part by the project “Bayesian Assessment of Safety Profiles for Pregnant Women From Animal Study to Human Clinical Trial” funded by U.S. Food and Drug Administration, Office of Women’s Health. The project was supported in part by an appointment to the Research Participation Program at the Center For Biologics Evaluation and Research administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and the U.S. Food and Drug Administration.

Chen was supported in part by the Department of Ecology and Evolutionary Biology at the University of Tennessee, Knoxville, and a grant from the National Science Foundation (MCB-1120370.)

Chen and Ostrouchov were supported in part by the project “Visual Data Exploration and Analysis of Ultra-large Climate Data” funded by U.S. DOE Office of Science under Contract No. DE-AC05-00OR22725. Ostrouchov, Schmidt, and Patel were supported in part by the project “NICS Remote Data Analysis and Visualization Center” funded by the Office of Cyberinfrastructure of the U.S. National Science Foundation under Award No. ARRA-NSF-OCI-0906324 for NICS-RDAV center.

This work used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This work also used resources of National Institute for Computational Sciences at the University of Tennessee, Knoxville, which is supported by the Office of Cyberinfrastructure of the U.S. National Science Foundation under Award No. ARRA-NSF-OCI-0906324 for NICS-RDAV center. This work used resources of the Newton HPC Program at the University of Tennessee, Knoxville.

We thank our colleague, Ed D’Azevedo from the Computational Mathematics Group, Computer Science and Mathematics Division, Oak Ridge National Laboratory (ORNL), for discussing ScaLAPACK issues and matrix computation.

We also thank Brian D. Ripley, Kurt Hornik, Uwe Ligges, and Simon Urbanek from the R Core Team for discussing package release issues and helping us solve portability problems on different platforms.
Warning: The findings and conclusions in this article have not been formally disseminated by the U.S. Department of Health & Human Services nor by the U.S. Department of Energy, and should not be construed to represent any determination or policy of University, Agency, Adminstration and National Laboratory.

This document is written to explain the main functions of **pbdSLAP** (Chen et al. 2012c), version 0.2-0. Every effort will be made to ensure future versions are consistent with these instructions, but features in later versions may not be explained in this document.

Information about the functionality of this package, and any changes in future versions can be found on website: http://r-pbd.org/.

1. Quick Start

The **pbdSLAP** package serves as a mechanism to utilize a subset of functions from the **ScaLAPACK** (Blackford et al. 1997) library from within **R** (R Core Team 2012), and in particular from the higher level **R** packages **pbdBASE** (Schmidt et al. 2012a) and **pbdDMAT** (Schmidt et al. 2012b). It allows one to merely “plug in” the necessary libraries without needing to do a complicated system installation. It is a bundling of the official ScaLAPACK distribution from the ScaLAPACK Team at netlib (http://www.netlib.org/scalapack/). However, it is possible to use other ScaLAPACK libraries instead; see Section 2.2 for details.

The **pbdSLAP** package consists of a minimum set of double precision functions from the **ScaLAPACK** library for **R**’s distributed matrix computation. **ScaLAPACK** includes many important functions for distributed linear algebra, including LU factorization, singular value decomposition, etc. We also include the necessary components of the libraries that our subset of **ScaLAPACK** relies on, namely **BLACS**, **PBLAS**, **BLAS**, and **LAPACK**.

The system requirements and installation instructions for **pbdSLAP** are provided in the following section. A technical issue for grid information of **BLACS** is described in the Section 4.

1.1. System Requirements

**pbdSLAP** requires **pbdMPI** (Chen et al. 2012a), which itself requires a system installation of MPI (http://en.wikipedia.org/wiki/Message_Passing_Interface). **pbdSLAP** should also work with LAM/MPI (http://www.lam-mpi.org/) and MPICH2 (http://www.mcs.anl.gov/research/projects/mpich2/).

**pbdSLAP** is mainly developed and tested under **OpenMPI** (http://www.open-mpi.org/) in Xubuntu 11.04 and 12.04 systems (http://xubuntu.org/). **pbdSLAP** should also run on other operating systems, such as Mac with OpenMPI, or Windows with MPICH2 if MPI is installed and launched properly. However, we have not extensively tested installation and use of the **pbd** toolchain on other platforms. The reader is encouraged to report his/her experience with **pbdSLAP** on other platforms.

1.2. Installation and Quick Start

The remaining assumes that **pbdMPI** is installed correctly. If **pbdMPI** is not yet installed, see the **pbdMPI** vignette (Chen et al. 2012b) for installation details. Users can download **pbdSLAP** from CRAN at http://cran.r-project.org, and the installation can be done with the following commands
Users can get started quickly with \texttt{pbdSLAP} by learning from the following example.

### Under command mode, run the demo with 2 processors by
### (Use Rscript.exe for windows system)

\texttt{mpiexec -np 2 Rscript -e "demo(gridinfo,'pbdSLAP',ask=F,echo=F)"

\section*{2. Useful Information}

\subsection*{2.1. CDEFS Flag of ScaLAPACK}

The \texttt{CDEFS} is a flag for interface between C and Fortran, and is required to compile \texttt{pbdSLAP}. \texttt{CDEFS = -DAdd_} is the default for GNU C and Fortran. The correct value can be determined by BLACS test intface tool in \texttt{pbdSLAP/inst/intface/}. For other compilers, the possible values could be \texttt{-DNoChange}, \texttt{-Df77IsF2C}, or \texttt{-DUpCase}.

For other values, user can provide an external \texttt{CDEFS} flag at installation time, such as

\begin{verbatim}
tar zxvf pbdMPI_0.1-0.tar.gz
R CMD INSTALL pbdSLAP \
   --configure-vars="CDEFS='other possible flags'"
\end{verbatim}

The value of \texttt{CDEFS} (other possible flags) can be one of the possible values above or other configurations. For PGI compiler, the value could be \texttt{-DAdd_ -DNO_IEEE $(USEMPI)}. Note that the \texttt{CDEFS} will overwrite the default \texttt{CDEFS} inside \texttt{pbdSLAP/src/Makevars.in} for compiling \texttt{libslap.a}.

\subsection*{2.2. Using Other Distributions of ScaLAPACK}

Some users may have access to other, often non-free, distributions of Scalapack, such as Intel’s \texttt{MKL} (\url{http://software.intel.com/en-us/intel-mkl}) or Cray’s \texttt{LibSci} (\url{http://docs.cray.com/}). It may be possible to achieve some performance gains using these libraries over netlib ScaLAPACK, especially when using big machines, such as Nautilus and Kraken in NICS (\url{http://www.nics.tennessee.edu}).

To use these external libraries with \texttt{pbdSLAP}, you need only supply the appropriate flag to \texttt{EXT_LDFLAGS} at compile time. Specifically, the user would issue the command:

\begin{verbatim}
tar zxvf pbdMPI_0.1-0.tar.gz
R CMD INSTALL pbdSLAP \
   --configure-vars="EXT_LDFLAGS='external ldflags'"
\end{verbatim}
where "external ldflags" needs to include linking information to ScaLAPACK, BLACS, LAPACK, and BLAS. Please be aware that the order matters in the EXT_LDFLAGS. Note that EXT_LDFLAGS will be part of PKG_LDFLAGS inside pbdSLAP/src/Makevars.in.

3. Inside pbdSLAP

Currently, pbdSLAP only supports MPI systems for communication. Some packages currently utilizing pbdSLAP are pbdBASE and pbdDMAT, which use R’s S4 classes and methods for distributed matrix computations. For details, see the vignettes (Schmidt et al. 2012c,d) of pbdBASE and pbdDMAT. pbdSLAP also provides a linking mechanism to export the library libslap.a for other applications.

3.1. Linking with pbdSLAP

pbdSLAP provides a static library libslap.a which is useful to link into other applications. For further reference, all linking information is stored in the file

```
${R_HOME}/library/pbdSLAP/etc${R_ARCH}/Makeconf
```

including the header files to pbdMPI and pbdSLAP, as well as the path to libslap.a or, if utilized, the external libraries such as MKL. An example configuration can be found in pbdBASE/src/Makevars.in. The linking flags are available from

```
pbdBASE/src/Makevars.in
```

```
R_SCMD = ${R_HOME}/bin/Rscript -e
SLAP_LDFLAGS = $(shell ${R_SCMD} \\
"source('..../R/get_lib.r'); \nget.lib('R_SLP',"${R_ARCH}"))
```

It is also very likely to link with pbdMPI since MPI is a prerequisite. One can obtain the system’s MPI information from pbdMPI, and the linking flags are available from

```
pbdSLAP/src/Makevars.in
```

```
R_SCMD = ${R_HOME}/bin/Rscript -e
SPMD_CPPFLAGS = $(shell ${R_SCMD} \\
"source('..../R/get_conf.r'); \nget.conf('PKG_CPPFLAGS',"${R_ARCH}"))
SPMD_LDFLAGS = $(shell ${R_SCMD} \\
"source('..../R/get_conf.r'); \nget.conf('PKG_LIBS',"${R_ARCH}"))
```

where `{R_ARCH}` is available from the default R Makeconf file.

4. Testing BLACS

Strictly speaking, pbdSLAP does not use the original way of interacting with BLACS to deal with grid information. In BLACS, the grid information is pointers pointing to C structures.
containing MPI communicators for grid construction. The ICTXT value of the C structure is the original way for Fortran to access MPI communicators.

In R, specifically pbdBASE, we use hidden global R objects (.__grid_info_* ) to store the grid information, where * is an integer depending on the BLACS context id (ictxt) for the grid initialized by function (slap.init.grid).

When computing is finished, we need to exit all the BLACS grids. For each grid, slap.exit.grid function can free the grid via the id (ictxt). Initially, the maximum number of ICTXT is 10 in BLACS, but can be dynamically allocated if this maximum is reached. Users do not need to directly manage this.

When all grids exit completely, we need to finalize pbdSLAP by calling slap.finalize, and by default MPI is usually not finalized. slap.finalize will take care of freeing memory before Quiting the pbdSLAP.

Now save the next script in a file and run with

```
Shell Command
mpirun -np 4 Rscript gridinfo.r
```

to see the grid information. This example provides four grids ictxt = 0,1,2,3 in R, but 0, 1 are exited before initializing 2, 3. The ICTXT shows that only 0, 1 in Fortran are used twice for all grids.

### File Name: gridinfo.r

```r
library (pbdMPI, quiet = TRUE)
library (pbdSLAP, quiet = TRUE)
init()

slap.init.grid (2, 2, ictxt = 0)
slap.init.grid (1, 4, ictxt = 1)

comm.cat(">> .__grid_info_0
", quiet = TRUE)
comm.print (.__grid_info_0, all.rank = TRUE)
comm.cat(">> .__grid_info_1
", quiet = TRUE)
comm.print (.__grid_info_1, all.rank = TRUE)

slap.exit.grid (0)
slap.exit.grid (1)

slap.init.grid (4, 1, ictxt = 2)
slap.init.grid (2, 3, ictxt = 3)

comm.cat(">> .__grid_info_2
", quiet = TRUE)
comm.print (.__grid_info_2, all.rank = TRUE)
comm.cat(">> .__grid_info_3
", quiet = TRUE)
comm.print (.__grid_info_3, all.rank = TRUE)

slap.exit.grid (2)
slap.exit.grid (3)
```
slap.finalize()
finalize()
References


