

MBatch 01 Installing on Linux
Tod Casasent
2017-10-30-1350

Introduction

These instructions are aimed at people familiar with Linux and R administration, with root access to their Linux environment. If using the versions discussed here, it may be possible to install MBatch with minimal administration knowledge.

Target Operating System and Installation

These instructions were tested on Debian 9.1 installed on Oracle VirtualBox. These instructions with appropriate modifications should work as a basis for installing MBatch on other distributions.

Perform the installs in the order given in this document. Some steps are dependent on previous steps.

Install Open JDK

Many Linux distributions do not install the JDK needed for compiling Java, and only install a JRE. So here, we install Open JDK.

```
# switch to "root" user
su -
# install JDK
apt-get install -f -y openjdk-8-jdk
# and we run an apt-get clean, to clean things up
apt-get clean
# exit root
exit
```

Install Package Prerequisites

MBatch, or rather some of the R libraries used by MBatch, have operating system dependencies. File names and dependencies will vary by distribution. These installs are for Debian 9.1.

```
# switch to "root" user
su -
```

```
# install packages related to other R packages
apt-get install -f -y wget libssh2-1 libcairo2-dev libxt-dev curl libcurl4-openssl-dev
libssl-dev libxml2-dev

# install packages for compiling packages and making PDFs from Rd files
apt-get install -f -y texlive

# and we run an apt-get clean, to clean things up
apt-get clean

# exit root
exit
```

Install R 3.4.x

MBatch uses R 3.6.x. It may work with older or newer versions, but packages and R itself tend to change between releases, so it is best to use 3.6.x. If your Linux distribution does not include R 3.6.x in a repository, use the instructions on a local CRAN mirror (<https://cran.r-project.org/mirrors.html>) under "Download R for Linux" (as shown in the screenshot). Also, see the notes under "Supported branches" for changes between 3.3 and 3.4 if need for your upgrade.

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

For Debian 9.1, the Debian repositories do not include R 3.6.x. So, we followed the instructions under "Download R for Linux" on CRAN to add a CRAN repository for R 3.6.x that is specific to Debian 9.1 (aka "stretch"). The instructions below add the repository (and other prerequisites) and install R 3.4.x from the CRAN repository. Other Linux distributions may work differently. Note also that we downloaded the JIRANKE key since the dynamic keyserver is not reliable.

```
# switch to "root" user
su -

# here we setup Debian to use an R 3.5 (which includes 3.6) specific repo, to
preserve reproducibility
echo 'deb http://cran.r-project.org/bin/linux/debian stretch-cran35/' >>
/etc/apt/sources.list

# grab the key for the repo from the Ubuntu keyserver
```

```

# this may very occasionally fail, but is more reliable than the network pool
#apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-key
'E19F5F87128899B192B1A2C2AD5F960A256A04AF'

# use pre-downloaded key
apt-key add /jranke.key
rm /jranke.key

# update apt-get to recognized the new repos
apt-get update

# install R3.6 and related packages
apt-get install -f -y -t stretch-cran35 littler r-cran-littler r-base r-base-dev r-recommended

# and we run an apt-get clean, to clean things up
apt-get clean
rm -rf /var/lib/apt/lists/*

# exit root
exit

# check version of R (should say something like: R version 3.6.0 (2019-04-26) --
"Planting of a Tree")
R --version

```

Cairo Prerequisites

Cairo Graphics Library for Linux is required by the R Cairo package (<http://www.rforge.net/Cairo/>). The Cairo Debian Library is installed in the instructions above. For other Linux distributions, check the repository or the Cairo website (<http://www.cairographics.org>).

Setting Up a Personal R Library

If you have not installed any packages in R, you will need to create a personal package for R libraries. Install a package, as shown below, and create a personal library directory.

```

> install.packages("devtools", repos = "http://cran.revolutionanalytics.com")
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
Warning in install.packages("devtools") :

```

'lib = "/usr/local/lib/R/site-library"' is not writable

Would you like to use a personal library instead? (y/n) y

Would you like to create a personal library

~/R/x86_64-pc-linux-gnu-library/3.4

to install packages into? (y/n) y

trying URL 'https://cran.revolutionanalytics.com/src/contrib/devtools_1.13.3.tar.gz'

Content type 'application/octet-stream' length 486402 bytes (475 KB)

=====

downloaded 475 KB

Installing Required Packages

MBatch has some required packages and setup for rJava.

Java and rJava Setup for R

R needs to know about Java. Depending on how R was built it may detect Java automatically, but for this install, we have to tell R about Java.

```
# switch to "root" user
```

```
su -
```

```
# Perform javareconf to setup Java and R
```

```
R CMD javareconf
```

```
# start R (--vanilla says don't load any environments or save any)
```

```
R --vanilla
```

```
# install rJava from CRAN (this installs to site-library so it is available for all users)
```

```
# update to preferred CRAN mirror if desired
```

```
# this is at the R command
```

```
install.packages("rJava", dependencies=TRUE, type="source", repos =  
"http://cran.revolutionanalytics.com")
```

```
# check that the Java version returns properly
```

```
# should print something like: [1] "1.8.0_144"
```

```
# this is at the R command
```

```
library(rJava)
```

```
.jinit()
J("java.lang.System")$getProperty("java.version")
# quit R
# this is at the R command
q()
# exit root
exit
```

Other Required Packages

Within R, run the following installs.

```
# switch to "root" user
su -

# start R (--vanilla says don't load any environments or save any)
R --vanilla

# install packages from CRAN (this installs to site-library so it is available for
all users)

# update to preferred CRAN mirror if desired
# this is at the R command
install.packages("Cairo", dependencies=TRUE, type="source", repos =
"http://cran.revolutionanalytics.com")
install.packages("epiR", dependencies=TRUE, type="source", repos =
"http://cran.revolutionanalytics.com")
install.packages("gtools", dependencies=TRUE, type="source", repos =
"http://cran.revolutionanalytics.com")
install.packages("mclust", dependencies=TRUE, type="source", repos =
"http://cran.revolutionanalytics.com")
install.packages("squash", dependencies=TRUE, type="source", repos =
"http://cran.revolutionanalytics.com")
install.packages("httr", dependencies=TRUE, type="source", repos =
"http://cran.revolutionanalytics.com")

# install packages from Bioconductor (this installs to site-library so it is available
for all users)

# if asked about updating a/s/n, choose "a" (all)
# this is at the R command
```

```

source("http://bioconductor.org/biocLite.R")
biocLite("limma", ask="a")
biocLite("RBGL", ask="a")
biocLite("graph", ask="a")
biocLite("Biobase", ask="a")
install.packages(c("oompaBase", "ClassDiscovery", "PreProcess"), dependen-
cies=TRUE, repos=c("http://cran.revolutionanalytics.com", "http://silicovore.com/OOMPA/"))
# install Vennerable from github (this installs to site-library so it is available
for all users)
# update to preferred CRAN mirror if desired
# this is at the R command
install.packages("devtools", repos = "http://cran.revolutionanalytics.com")
library(devtools)
install_github("js229/Vennerable")
#Check that all packages installed properly. Should return [1] TRUE
all(c("Cairo", "epiR", "gtools", "mclust", "squash", "httr", "oompaBase", "ClassDiscovery", "PreProcess", "devtools", "limma") %in% installed.packages()[,1])
# quit R
# this is at the R command
q()
# exit root
exit

```

Install RStudio

If desired, download and install the "Ubuntu 16.04+/Debian 9+ (64-bit)" version of RStudio from (<https://www.rstudio.com/products/rstudio/download/#download>). RStudio is an integrated R environment for developers. After the install, RStudio is available under the Development menu.

```

# switch to "root" user
su -

wget -O /tmp/rstudio-xenial-1.0.153-amd64.deb https://download1.rstudio.org/rstudio-
xenial-1.0.153-amd64.deb
# cd to download directory and update for correct deb file name

```

```
cd /tmp
dpkg -i rstudio-xenial-1.0.153-amd64.deb
# exit root
exit
```

Install MBatch Package

MBatch can be installed from GitHub using the following commands. (The devtools package was installed earlier.)

```
# At Linux command line (this install is for your own user, su to root to install
for everyone)

# start R (--vanilla says don't load any environments or save any)

R --vanilla

# Within R

library(devtools)

devtools::install_git("https://github.com/MD-Anderson-Bioinformatics/MBatch",
subdir="package")

# quit R

# this is at the R command

q()
```

Note: As a side note, the "tests" directory installed via devtools is not by default included in install.packages or R CMD INSTALL. If installing MBatch via a different method, to include tests via the command line, use "--install-tests".

Create base directory for testing

MBatch uses a default directory of /bea_testing for tests. It also needs a directory named /bea_testing/output.

```
# At Linux command line

# switch to root to create top level directory

su -

# create top level directory

mkdir /bea_testing

# create output

mkdir /bea_testing/output
```

```
# update ownership (assuming linux is your user name and that user's own
package)

chown -R linux:linux /bea_testing

# leave root

exit
```

Populating Data for Tests

The files located at https://github.com/MD-Anderson-Bioinformatics/BatchEffectsPackage/tree/master/data/testing_static/MATRIX_DATA go in the /bea_testing/MATRIX_DATA folder.

The files located at https://github.com/MD-Anderson-Bioinformatics/BatchEffectsPackage/tree/master/data/testing_static/COMPARE go in the /bea_testing/COMPARE folder.

Running the Tests

Finally, run this code to perform the tests. This takes an hour or so to run. Machines with slower processors, slower hard drives, or minimal memory may take longer or be unable to complete the tests.

```
# At Linux command line (do not do this as root, as root is not needed)
```

```
# start R (--vanilla says don't load any environments or save any)
```

```
R --vanilla
```

```
# Within R
```

```
library(MBatch)
```

```
# Set these environment variable to override file locations if needed
```

```
Sys.setenv(MBATCH_TEST_OUTPUT="/bea_testing/output")
```

```
Sys.setenv(MBATCH_TEST_INPUT="/bea_testing/MATRIX_DATA")
```

```
Sys.setenv(MBATCH_TEST_COMPARE="/bea_testing/COMPARE")
```

```
baseDir <- file.path(system.file(package = "MBatch"), "tests")
```

```
message(baseDir)
```

```
testFiles <- list.files(path=baseDir)
```

```
print(testFiles)
```

```
results <- c()
```

```
for(myFile in testFiles)
```

```
{
```



```

message("*****")
message("*****")
message("**** ", file.path(baseDir, myFile))
message("*****")
message("*****")
test <- source(file.path(baseDir, myFile))
if (isTRUE(test$value))
{
  results <- c(results, paste("Test succeeded for ", myFile, sep=""))
}
else
{
  results <- c(results, paste("Test failed for ", myFile, sep=""))
}
}
print(results)

```