

MBatch 01 Installing on Linux
Tod Casasent
2022-08-24-1200

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 AlmaLinux 8 (via Docker) and RHEL8. These instructions with modifications should work for other distributions. Perform the installs in the order given in this document. The newest steps can be found in the Dockerfile on GitHub at: <https://github.com/MD-Anderson-Bioinformatics/BatchEffectsPackage/tree/master/docker-build/BCBbase>

Install Package Prerequisites

MBatch, or rather some of the R libraries or other applications used by MBatch, have operating system dependencies. File names and dependencies will vary by distribution. These installs are for AlmaLinux/CentOS 8. Note that often it is the "devel" (or developer) version of a package that is required.)

```
# switch to "root" user (or use sudo for each command, depending on your distribution)
```

```
sudo su -
```

```
# installs for R-related elements (like cairo)
```

```
# and build elements like Git and Maven
```

```
# set enables repo names need to be all lowercase, despite some online docs indicating otherwise
```

```
dnf -y install dnf-plugins-core
```

```
dnf config-manager --set-enabled powertools
```

```
dnf config-manager --set-enabled appstream
```

```
dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
yum install -y https://download1.rpmfusion.org/free/el/rpmfusion-free-release-8.noarch.rpm https://download1.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-8.noarch.rpm
```

```

yum install -y http://rpmfind.net/linux/centos/8-stream/PowerTools/x86_64/os/Packages/SDL2-2.0.10-2.el8.x86_64.rpm
yum install -y ffmpeg ffmpeg-devel
yum install -y unzip diffutils openssl-devel libxml2-devel cairo-devel libXt-devel
yum install -y udunits2-devel proj-devel geos-devel gdal-devel sqlite sqlite-devel
yum install -y perl-Tk git git-gui ant maven nodejs file wget
yum install -y harfbuzz fribidi harfbuzz-devel fribidi-devel ffmpeg-devel
yum install -y libcurl-devel libjpeg-turbo-devel
yum install -y tesseract tesseract-devel leptonica-devel cargo
yum install -y poppler-cpp-devel ImageMagick-c++-devel
yum install -y libwebp-devel librsvg2-devel libgit2-devel
yum update -y
yum clean all
npm install -g express-generator
# Install all TeXLive
dnf install -y texliv*
# install pandoc after texlive, since it depends on texlive
yum install -y pandoc
yum update -y
yum clean all
# exit root
exit

```

Install Python

Python installs are for AlmaLinux/CentOS 8. Similar installs work for other systems. This installs Anaconda (conda) for the needed Conda environment.

```
# switch to "root" user (or use sudo for each command, depending on your distribution)
```

```
sudo su -
```

```

wget https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh
mkdir /home/bcbuser/conda

```

```
# do not use unattended install so you can select automatic init
bash /home/bcbuser/Anaconda3-2022.05-Linux-x86_64.sh /home/bcbuser/conda
-f
source /home/bcbuser/conda/bin/activate
conda init
conda update -y conda
# exit root
exit
```

Install R 4+

MBatch uses R 4+. It may work with older or newer versions, but packages and R itself tend to change between releases, so it is best to use 4+. If your Linux distribution does not include R 4+ 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). We install the Java-supporting version below.

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.

```
# switch to "root" user (or use sudo for each command, depending on your
distribution)
```

```
sudo su -
```

```
# install R 4.x+
```

```
dnf install -y R-java-devel
```

```
echo 'LC_CTYPE="C"' >> /usr/lib64/R/.Renviron && \
```

```
echo 'LC_TIME="C"' >> /usr/lib64/R/.Renviron && \
```

```
echo 'LC_MESSAGES="C"' >> /usr/lib64/R/.Renviron && \
```

```
echo 'LC_MONETARY="C"' >> /usr/lib64/R/.Renviron && \
```

```
echo 'LC_PAPER="C"' >> /usr/lib64/R/.Renviron && \
```

```
echo 'LC_MEASUREMENT="C"' >> /usr/lib64/R/.Renviron
```

```
# register Java with R
```

R CMD javareconf

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.

NGHCM Required Packages

Within R, run the following installs.

```
# if desired, switch to "root" user (or use sudo for each command,
# depending on your distribution) to install globally
sudo 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("httr", dependencies = TRUE, repos = "http://cran.r-
project.org")
install.packages("magick", dependencies = TRUE, repos = "http://cran.r-
project.org")
install.packages("gert", dependencies = TRUE, repos = "http://cran.r-
project.org")
install.packages("htmlwidgets", dependencies = FALSE, repos = "http://cran.r-
project.org")
install.packages("shiny", dependencies = TRUE, repos = "http://cran.r-
project.org")
install.packages("usethis", dependencies = TRUE, repos = "http://cran.r-
project.org")
install.packages("covr", dependencies = TRUE, repos = "http://cran.r-
project.org")
install.packages("rversions", dependencies = TRUE, repos = "http://cran.r-
project.org")
install.packages("devtools", dependencies = TRUE, repos = "http://cran.r-
project.org")
library(devtools)
# remotes
devtools::install_github('MD-Anderson-Bioinformatics/tsvio')
devtools::install_github('MD-Anderson-Bioinformatics/NGCHMSupportFiles',
ref='main')
devtools::install_github('MD-Anderson-Bioinformatics/NGCHM-R')
# quit R
# this is at the R command
q()
# exit root
exit

```

MBatch Required Packages

Within R, run the following installs.

```

# if desired, switch to "root" user (or use sudo for each command,
# depending on your distribution) to install globally
sudo su -

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

message("###* reticulate used by MBatch")

install.packages("reticulate", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("###* mclust used by oompaBase, ClassDiscovery, and PreProcess (in
install 1)")

install.packages("mclust", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("###* Biobase used by oompaBase, ClassDiscovery, and PreProcess
(in install 1)")

install.packages("BiocManager", dependencies=TRUE, repos = "http://cran.r-
project.org")

BiocManager::install(c("Biobase"), update=FALSE)

message("###* oompaBase, ClassDiscovery, and PreProcess")

# used by PCA+ and other clustering algorithms in MBatch
# uses ClassDiscovery functions distanceMatrix and SamplePCA

install.packages(c("oompaBase", "ClassDiscovery", "PreProcess"), dependen-
cies=TRUE, repos=c("http://cran.r-project.org", "http://silicovore.com/OOMPA/"))

message("###* Cairo used in MBatch for images")

install.packages("Cairo", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("###* BiasedUrn needed for epiR")

install.packages("BiasedUrn", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("###* pander needed for epiR")

install.packages("pander", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("###* sf needed for epiR")

install.packages("sf", dependencies=TRUE, repos = "http://cran.r-project.org")

message("###* lubridate needed for epiR")

```

```

install.packages("lubridate", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("### epiR used in correlation density plot")

#install.packages("epiR", dependencies=TRUE, repos = "http://cran.r-
project.org")

#Install version 2.0.40, anything newer fails
install.packages("https://cran.r-project.org/src/contrib/Archive/epiR/epiR_2.0.40.tar.gz")

message("### squash used for dendromat in Supervised Clustering and Hier-
archical Clustering")

install.packages("squash", dependencies=TRUE, repos = "http://cran.r-
project.org")

# used in DAPIR
message("### httr")

install.packages("httr", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("### jsonlite")

install.packages("jsonlite", dependencies=TRUE, repos = "http://cran.r-
project.org")

# uses dunn.test in MBatchUtils
message("### dunn.test")

install.packages("dunn.test", dependencies=TRUE, repos = "http://cran.r-
project.org")

# used for UMAP
message("### uwot")

install.packages("uwot", dependencies=TRUE, repos = "http://cran.r-
project.org")

message("### devtools")

install.packages("devtools", dependencies=TRUE, repos = "http://cran.r-
project.org")

library(devtools)

message("### jlmelville/vizier")

devtools::install_github("jlmelville/vizier")

# quit R

# this is at the R command

```

```
q()
# 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_github("MD-Anderson-Bioinformatics/BatchEffectsPackage/apps/MBatch")

# 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 in <https://github.com/MD-Anderson-Bioinformatics/BatchEffectsPackageData> are the test data files.

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

The files located at https://github.com/MD-Anderson-Bioinformatics/BatchEffectsPackageData/tree/master/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)

```

Install Open JDK (no longer required)

This is no longer required, but the documentation is retained for historical purposes.

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 (or use sudo for each command, depending on your
distribution)
```

```
sudo su -
```

```
# install OpenJDK 17
```

```
dnf -y install java-17-openjdk java-17-openjdk-devel java-17-openjdk-headless
```

```
update-alternatives --install /usr/bin/java java /usr/lib/jvm/java-17-
openjdk/bin/java 1
```

```
update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/java-17-
openjdk/bin/javac 1
```

```
update-alternatives --set java /usr/lib/jvm/java-17-openjdk/bin/java
```

```

update-alternatives --set javac /usr/lib/jvm/java-17-openjdk/bin/javac
yum update -y
yum clean all

# adding these variables fixes issue where registering Java with R (keeps getting
"lost")

ENV JAR=/usr/lib/jvm/java-17-openjdk/bin/jar
ENV JAVAH=/usr/lib/jvm/java-17-openjdk/bin/javac
ENV JAVA_LIBS="-L/usr/lib/jvm/java-17-openjdk/lib/server -ljvm"
ENV JAVA_CPPFLAGS="-I/usr/lib/jvm/java-17-openjdk/include -I/usr/lib/jvm/java-17-openjdk/include/linux"
ENV JAVA_LD_LIBRARY_PATH=/usr/lib/jvm/java-17-openjdk/lib/server
ENV JAVA_HOME=/usr/lib/jvm/java-17-openjdk/
ENV PATH=/usr/lib/jvm/java-17-openjdk/bin:$PATH
ENV JAVA=/usr/lib/jvm/java-17-openjdk/bin/java
ENV JAVAC=/usr/lib/jvm/java-17-openjdk/bin/javac
ENV JAVAH=/usr/lib/jvm/java-17-openjdk/bin/javac
ENV JAR=/usr/lib/jvm/java-17-openjdk/bin/jar
ENV LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/lib/jvm/java-17-openjdk/lib/server

# exit root
exit

```