

R on the clusters

R is provided via the [DCSR software stack](#)

Interactive mode

To load R:

```
$> module load r-light
$> R
# Then you can use R interactively
> ...
```

By default, you get the last version available (4.4.1 when this page is written). If you need an older version, you can list the available versions as follows:

```
$> module spider r-light
```

```
-----
r-light:
-----
```

Versions:

```
r-light/3.6.3
r-light/4.0.5
r-light/4.1.3
r-light/4.2.3
r-light/4.3.3
r-light/4.4.1
```

Then you can load a specific version:

```
$> module load r-light/4.0.5
$> R --version
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
```

Batch mode

While using R in batch mode, you have to use `Rscript` to launch your script. Here is an example of sbatch script, `run_r.sh`:

```
#!/bin/bash

#SBATCH --time 00-00:20:00
#SBATCH --cpus-per-task 1
#SBATCH --mem 4G

module load r-light

Rscript my_r_script.R
```

Then, just submit the job to Slurm:

```
sbatch run_r.sh
```

Package installation

A few core packages are installed centrally - you can see what is available by using the `library()` function. Given the number of packages and multiple versions available, other packages should be installed by the user.

Library relocation

By default, when you install R packages, R will try to install them in the central installation. Since this central installation is shared among all users on the cluster, it's obviously impossible to install directly your packages there. This is why this location is not writable and you will get this kind of message:

```
$> R
> install.packages("ggplot2")
Warning in install.packages("ggplot2") :
  'lib = "/opt/R-4.4.1/lib/R/library"' is not writable
Would you like to use a personal library instead? (yes/No/cancel)
```

This is why you have to answer **yes** to this "Would you like to use a personal library instead?" question.

By default, this personal library is located in your home directory. On DCSR clusters, this home directory is pretty limited regarding the amount of data (50 GB at most) and the number of files (200'000 files at most) you can store. Installing R packages in your home directory could quickly fill all the available space. This is why your personal library should be relocated.

A good practice is to relocate your R library to a location in one of your work project. Let's consider your work project is located in `/work/FAC/Lettres/GREAT/ulambda/default`, you create a sub-directory inside, for instance `/work/FAC/Lettres/GREAT/ulambda/default/RLIB_for_ursula`. Then you have several options to tell R that you want to use this new personal library, but the easiest way is to define the `R_LIBS_USER` variable.

Thus, you can either add the following line in all your Slurm scripts (before R is invoked):

```
export R_LIBS_USER=/work/FAC/Lettres/GREAT/ulambda/default/RLIB_for_ursula
Rscript ...
```

Or you can also define it in the `~/.Renviron`. You just have to add the following line to the file:

```
R_LIBS_USER=/work/FAC/Lettres/GREAT/ulambda/default/RLIB_for_ursula
```

The second option using `~/.Renviron` is probably cleaner but the first option is more versatile, especially if you want to use several personal libraries depending on different projects and requirements.

CRAN packages

Installing R packages from CRAN is pretty straightforward thanks to [install.packages\(\)](#) function. For instance:

```
$> module load r-light
$> R
> install.packages(c("ggplot2", "dplyr"))
```

BioConductor packages

The first step is to install the BioConductor package manager, and then to install packages with `BiocManager::install()`. For instance:

```
$> module load r-light
$> R
```

```
> install.packages("BiocManager")  
> BiocManager::install("biomaRt")
```

Github/development packages

To install packages from Github/Gitlab or random websites, you can use the `devtools` library as follows:

```
$> module load r-light  
$> R  
> library(devtools)  
> install_github("N-SDM/covsel")  
> install_url("https://cran.r-project.org/src/contrib/Archive/rgdal/rgdal_1.6-7.tar.gz")
```

Missing dependencies

In some cases, it's possible that package installation fails because of missing dependencies. In such case, please send us an email to helpdesk@unil.ch with the subject starting with "DCSR R package installation". And please provide us with the name of the package that you cannot install.

Révision #8

Créé 3 juillet 2024 06:42:58 par Emmanuel Jeanvoine

Mis à jour 17 mars 2025 14:49:40 par Cristian Ruiz