R on HPC cluster:

What is R?

R is a programming language for statistical computing and graphics supported by the R Core Team and the R Foundation for Statistical Computing. It is available across widely used platforms like Windows, Linux, and macOS. Some features of R are:

- Basic Statistic
- Probability
- Data analysis
- Graphics

More documentation about R can be found in the following links:

Official Documentation

Tutorials

Versions Available:

The following versions of R are available on the cluster:

- R -v3.2
- R-v3.2.5
- R-v3.5.0
- R-v4.0.4

How to load a version of R?

To see all the available versions of R in the cluster,

```
module avail math/R
```

Now, load the desired version, R-v4.0.4 in this case, using the following command,

```
module load math/R/4.0.4
```

To verify if the module and dependencies are loaded properly, use the following command,

```
module list
```

All the dependencies for R - GCC, open MPI and Java - should be loaded.

How to manage package in R?

To see all the package installed in R,

```
echo "installed.packages()" | R --slave | less #use q to quit
```

or use the following command inside R.

```
> installed.packages()
```

To get the usage and more info about the package, use the following,

```
echo "help(package="<package name>")" | R --slave
```

or use the following command inside R.

```
help(package="<package name>")
```

To see all the libraries available,

```
echo "library()" | R --slave
```

To install most R packages, use the following inside R,

```
> install.packages(<package_name>)
```

Note: Not all packages can be installed using the following method. Some packages have special dependencies that require different methods of installation.

Another approach to set up an R environment:

Users can use Anaconda environment to set up R environment. It is much easier to customize the environment this way. Users can install specific versions of packages, remove packages and edit packages in their environment.

To install R in conda environment, please follow the instructions:

```
#Load Miniconda
module load miniconda3/base

#Create Conda environment and install R and R-package
conda create --name r-env --channel conda-forge r-dplyr r-rmapshaper
r-sf

# Activate Environment
conda activate mshpr-env

#Start R installed on user’s environment
R
```

Note: This installs R on the user’s own envionment. User needs to load miniconda and activate their environment to use this version of R.

How to use R on the cluster?

There are two main ways to use R on the cluster.

The interactive terminal:

```
#Jump into compute node
srun -p main --qos main -n 1 -c 16  --mem 16G --pty bash

# Start load the module and start R
R
#Follow command interactively on R terminal

#If user wants to run a R-script, use
Rscript myscript.r
```

The Batch Job:

To run a batch job, use the following template

```
#!/bin/bash
#SBATCH --job-name=R_job      # create a short name for your job
#SBATCH --nodes=1                # node count
#SBATCH --ntasks=1               # total number of tasks in all node
#SBATCH --cpus-per-task=1        # cpu-cores per task (>1 threaded)
#SBATCH --mem-per-cpu=4G         # memory per cpu-core
#SABTCH -p main #partition
#SBATCH --qos main  #Quality of service
#SBATCH --time=00:01:00          # total run time limit (HH:MM:SS)
#SBATCH --mail-type=all          # send email on start, end and fault
#SBATCH --mail-user=<YourNetID>@crimson.ua.edu

#Load the Module
module purge && module load math/R/

#Go to the script directory
cd $SLURM_SUBMIT_DIRECTORY

#Run the R script
Rscript myscript.R
```

Example MPI script:

```
library("datasets")
library("snow")
library("Rmpi")

mydata = iris[,-5] #dataset used to test
self.num = c(3,5,7,9,10) #centers tested
nboot.d=5

parallel.function <- function(i,data,centers) {
            kmeans(data, centers, nstart=i )
        }
cl <- makeCluster( mpi.universe.size()-1, type="MPI" )
clusterExport(cl, c('data'))

for(round.j in c(1:length(self.num))){
  para.result <- parLapply( cl, rep(1,nboot.d), fun=parallel.function,
data=mydata, centers=self.num[round.j])
```

```
  print(para.result)
}

stopCluster(cl)
mpi.exit()
```

To run a MPI job, use the following template

```
#!/bin/sh
#SBATCH --time=01:00:00
#SBATCH -N 2
#SBATCH --ntasks-per-node=4
#SBATCH --mem-per-cpu=1024
#SBATCH --error=job.%J.err
#SBATCH --output=job.%J.out
#SBATCH -p main
#SBATCH --qos main

module load math/R/3.5.0
cd $SLURM_SUBMIT_DIR

mpirun -n 1 R CMD BATCH Rmpi.R
```

To run R on multiple processors, use the following website as a guide

Parallel R

To run R on multi nodes, use the following website as a guide

R with MPI

Where to find help?

If you are stuck on some part or need help at any point, please contact OIT at the following address.

https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp