In-Person

Foundations in Genomic Analyses

Creating an R Environment for Analysis with the GCC

Northwestern | INFORMATION TECHNOLOGY
RESEARCH COMPUTING AND DATA SERVICES

COMPUTING AND SOFTWARE

Access to high performance computing, research software, and global networks for conducting computationally intense research.

DATA MANAGEMENT AND SHARING

Learn about data management planning and options for storing, securing, transferring, and sharing data.

DATA SCIENCE, STATISTICS, AND VISUALIZATION

Support for collecting, analyzing, visualizing, and programming with research data.

TRAINING AND CONSULTATION

Identify events, resources, and people to help you learn computational and data skills for your research.

Research Computing and Data Services

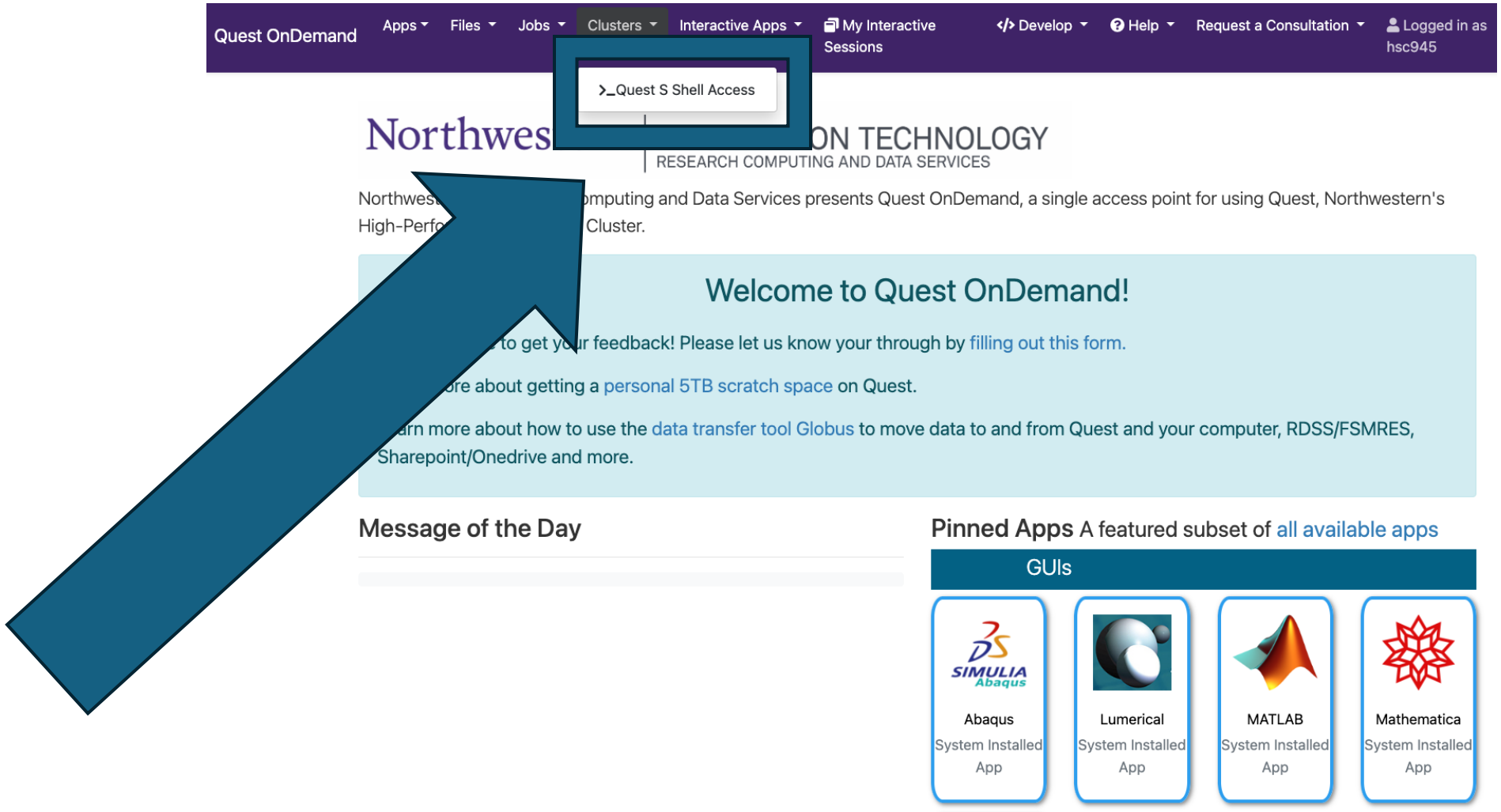
We're here to help after the workshop!

quest-help@northwestern.edu

bit.ly/rcdsconsult

https://sites.northwestern.edu/researchcomputing/

qondemand.ci.northwestern.edu



In the terminal tab that opens…

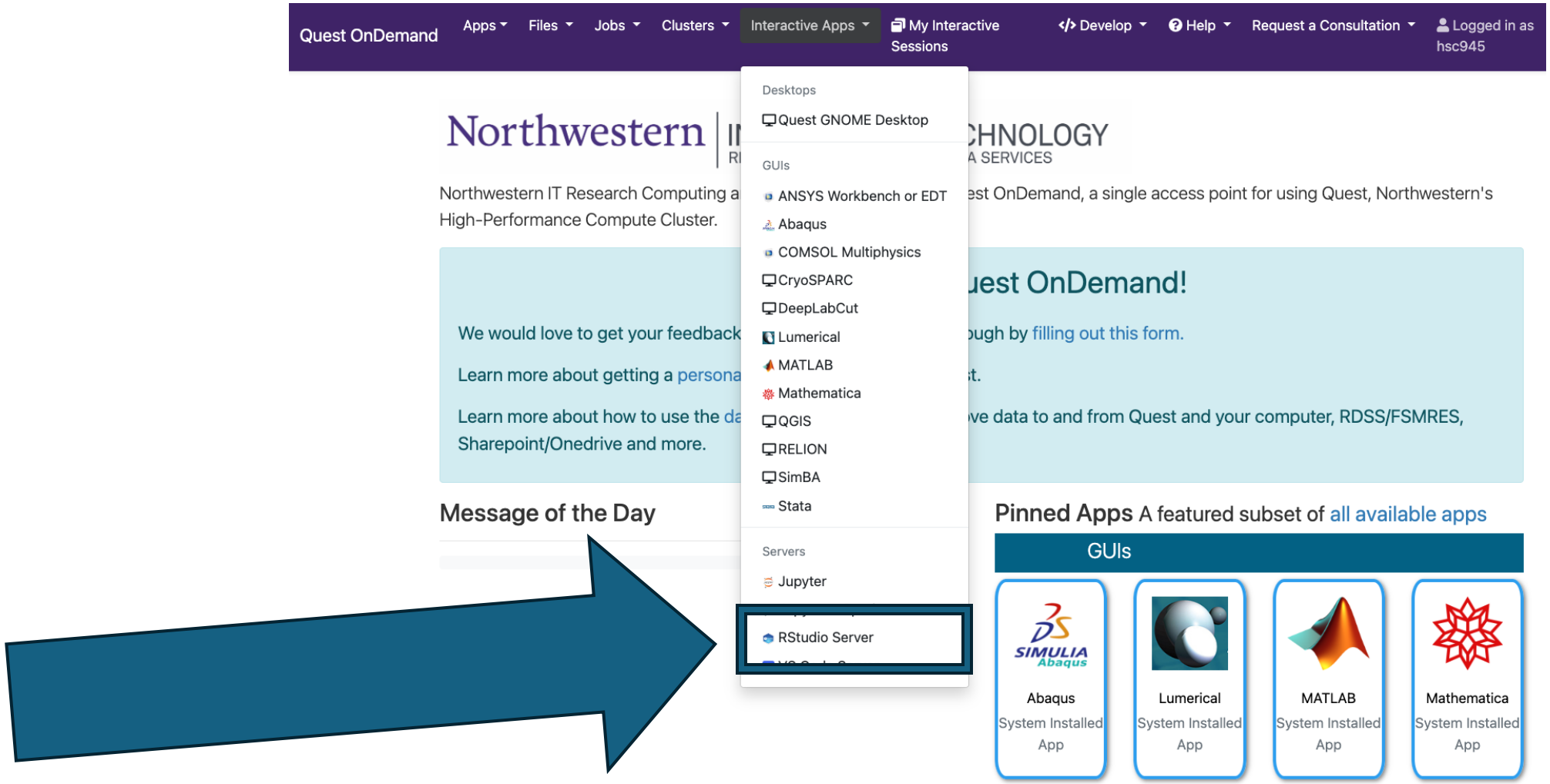
1. move to our classroom folder

   cd /projects/e32559

2. make your own subfolder if you haven't already

   mkdir <folder_name>

qondemand.ci.northwestern.edu

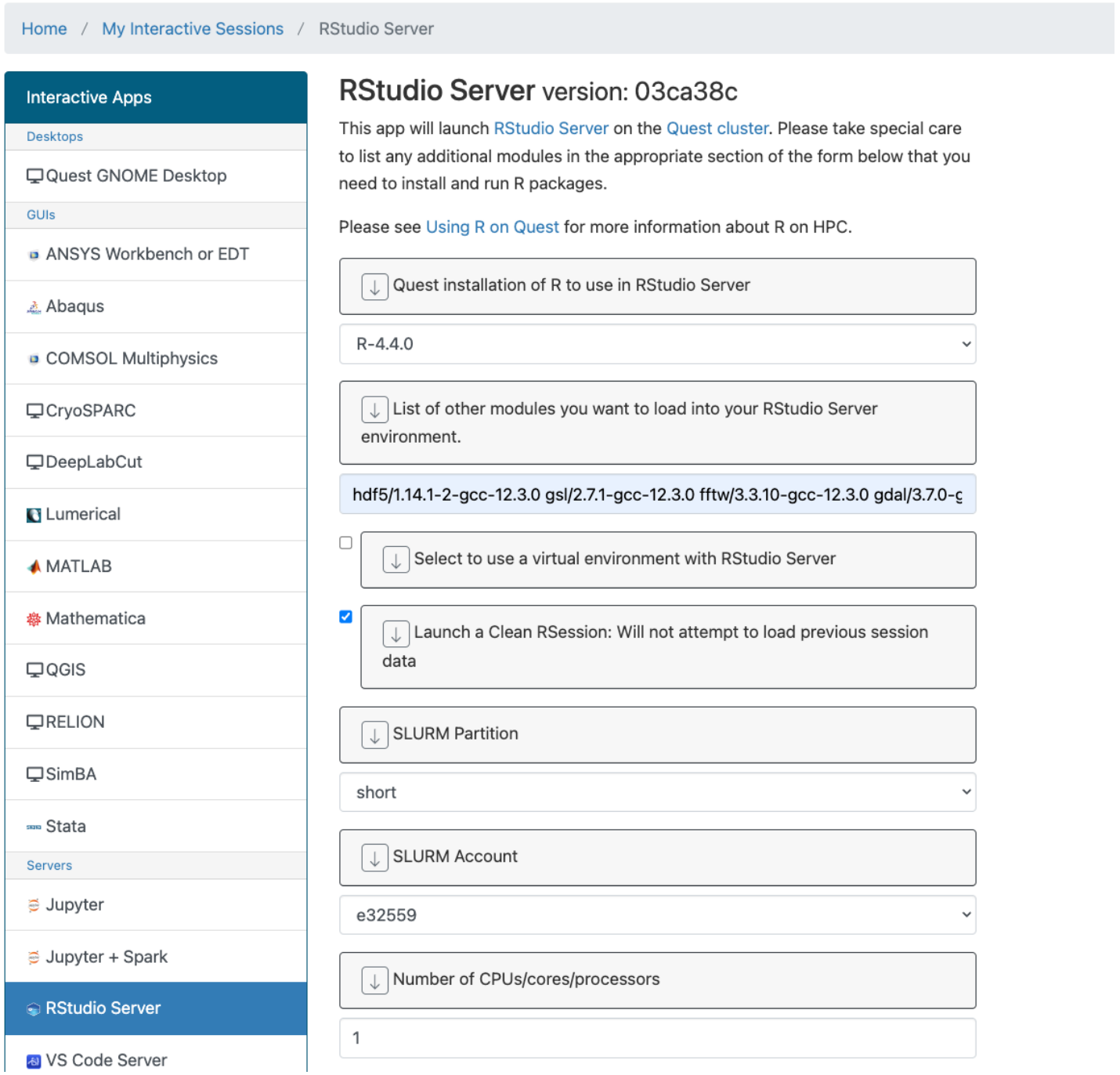


Quest OnDemand link: qondemand.ci.northwestern.edu

R version: 4.4.0

Additional modules: hdf5/1.14.1-2-gcc-12.3.0 gsl/2.7.1-gcc-12.3.0 fftw/3.3.10-gcc-12.3.0 gdal/3.7.0-gcc-12.3.0 nlopt/2.7.1-gcc-12.3.0

Accout: e32559
CPU: 1
Time: 1 hour
Memory: 3



Why these additional modules?

- sometimes R packages use software in addition to R as dependencies
- for example:
  - hdf5r is an R package for linking to HDF5 binary data format so it also needs the software for HDF5 data formats
  - nloptr is an R package that interfaces with NLopt software for non-linear optimization, so it also needs the NLopt software available

If you install an R package and get an error about missing files that are named something like libssl.so, or being unable to link to something that usually means you're missing some dependent software.

R packages

- base R has a lot of built in and useful functions
- many people write packages with additional functions for specific analyses/tasks
- for example, tidyverse
- first step is often to load necessary packages into your environment (or to install them)

```
# load packages
library(Seurat)
library(SeuratData)
library(SeuratWrappers)
library(Azimuth)
library(ggplot2)
library(patchwork)

# now I can do analyses!
```

install.packages() vs library()

- install.packages(), or other install commands, downloads and installs the desired package from the internet to your computer
- library() loads a package that is already installed on your computer to your active software environment

You can only use the functions in your active software environment!

```
select(tibble, column1, column2) # error
library(dplyr)
select(tibble, column1, column2) # works!
```

Where do R packages go when you install?

- install.packages() will use the output of .libPaths() to determine where to install packages
- by default, it knows about
  - where the version of R you're using was installed
  - and your home directory

Run .libPaths() in your console to see these paths.

.libPaths()

[1] "/home/hsc945/R/x86_64-pc-linux-gnu-library/4.4”

[2] "/hpc/software/spack_v20d1/spack/opt/spack/linux-rhel7-x86_64/gcc-12.3.0/r-4.4.0-aaqsqjfnzw5p5c4twtddj4sxqi23pyfw/rlib/R/library"

Let’s take a look!

cd /home/<netid>/R/x86_64-pc-linux-gnu-library/4.4

ls

```
(base) [hsc945@quser34 4.4]$ ls
abind              Deriv              gtools             modelr             RcppArmadillo      spatstat.random
affy               desc               hardhat            MsCoreUtils        RcppEigen          spatstat.sparse
affyio             devtools           haven              MultiAssayExperiment RcppGSL          spatstat.utils
AnnotationFilter   diagram            hdf5r              munsell            RcppHNSW           SQUAREM
askpass            diffobj            here               mzID               RcppParallel       StanHeaders
backports          digest             highr              nbpMatching        RcppProgress       statmod
base64enc          distributional     Hmisc              nlme               RcppTOML           stringfish
bayesplot          doBy               hms                nloptr             RcppZiggurat       stringi
BH                 doParallel         htmlTable          nnet               readr              stringr
Biobase            dotCall64          htmltools          numDeriv           readxl             SummarizedExperiment
BiocBaseUtils      downlit            htmlwidgets        openssl            recipes            survival
BiocGenerics       dplyr              httpuv             pacman             rematch            svglite
BiocManager        dqrng              httr               parallelly         rematch2           sys
BiocParallel       DT                 httr2              patchwork          remotes            systemfonts
BiocVersion        dtplyr             ica                pbapply            reprex             tensor
bit                dygraphs           ids                pbkrtest           reshape2           tensorA
bit64              e1071              igraph             pcaMethods         reticulate         terra
bitops             ellipsis           impute             pillar             Rhdf5lib           testthat
blob               evaluate           ini                pkgbuild           rlang              textshaping
boot               fansi              inline             pkgconfig          rmarkdown          threejs
```

You can also look in x86_64-pc-linux-gpu-library for other R versions and the packages you’ve installed for them.

Why might having one 4.4 folder for packages be a problem?

- different versions of packages may not be compatible
- different version of the same package might have different syntax
- you may work on projects that need different packages with the same version of R
- you may upgrade from R/4.4.0 to R/4.4.1 and find compatibility issues

What is the solution?

All package manager systems are basically **making separate folders for each environment**.

1. You can manually make different folders for different projects.
2. You can use Renv to control R package environments.
3. You can use mamba to create software environments for R or many other pieces of software.

What is the solution?

All package manager systems are basically **making separate folders for each environment**.

1. You can manually make different folders for different projects.
2. You can use Renv to control R package environments.
3. You can use mamba to create software environments for R or many other pieces of software.

Note: We will cover 1 and 3 today, not because 2 isn't a good option but because of what we have experience with and because 3 can be useful beyond R for software environments on Quest.

Specify a custom location for packages.

In terminal:

cd <your folder in our classroom allocation>

mkdir R_packages

In R:

.libPaths("/projects/e32559/<your folder>/R_packages")

.libPaths() # should see new folder on top

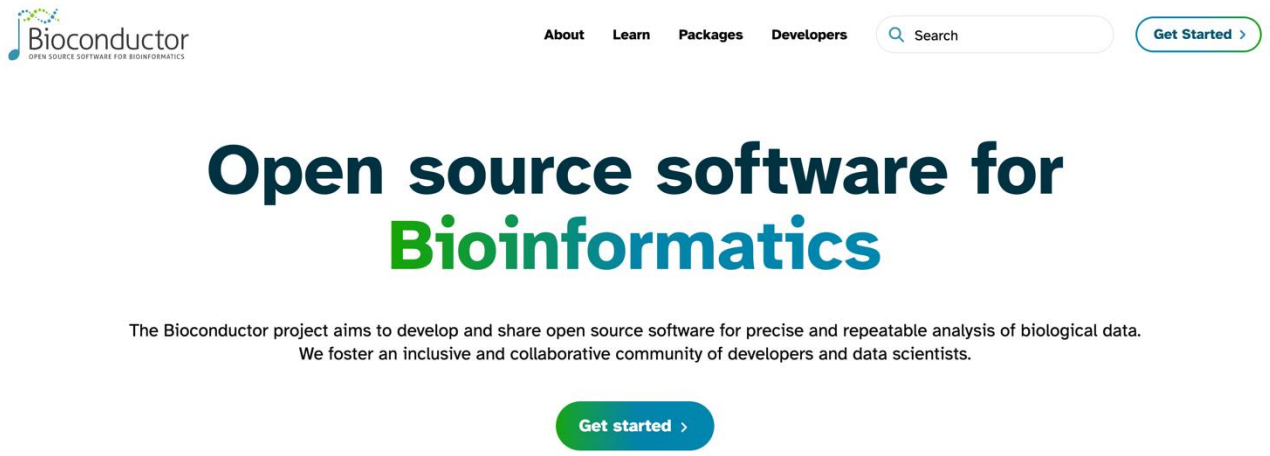
Note: If you click from the "Files" to the "Packages" tab in the lower right hand window you can see what's available change as you change .libPaths().

Install packages

```
install.packages("BiocManager")
# you can see this in your folder when it's done
BiocManager::install("SingleCellExperiment")
# this will take a while, so let it run. It also installs all dependent packages because we're starting from an almost empty folder.
BiocManager::install("scater")
BiocManager::install("OSCA.intro")
```

BiocManager installs from Bioconductor

- Bioconductor is a repository, like CRAN
- hosts R packages for bioinformatics
- has its own semi-solution for version control:
  - specific versions of BiocManager tied to specific version of R
  - specific versions of each package tied to each version of BiocManager
- https://www.bioconductor.org/



Install packages

```
install.packages("BiocManager”)
BiocManager::install("SingleCellExperiment")
# now should have many more R packages in install folder
# load the functions from SingleCellExperiment with
library(SingleCellExperiment)
```

Virtual environments for R and beyond

- mamba allows you to build software environments with more functionality than just a separate folder
- great for version control, and checking compatibility when you want to add something new
- exportable so others on the same system can copy your environment

In the terminal:

```
module load mamba/24.3.0
```

mamba create --name scexp -c conda-forge -c bioconda bioconductor-singlecellexperiment

Package	Version	Build	Channel
Install:			
+ _libgcc_mutex	0.1	conda_forge	conda-forge/linux-64
+ _openmp_mutex	4.5	2_gnu	conda-forge/linux-64
+ _r-mutex	1.0.1	anacondar_1	conda-forge/noarch
+ argcomplete	3.5.1	pyhd8ed1ab_0	conda-forge/noarch
+ binutils_impl_linux-64	2.40	hf600244_0	conda-forge/linux-64
+ bioconductor-biobase	2.62.0	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-biocgenerics	0.48.1	r43hdfd78af_2	bioconda/noarch
+ bioconductor-data-packages	20231203	hdfd78af_0	bioconda/noarch
+ bioconductor-delayedarray	0.28.0	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-genomeinfodb	1.38.1	r43hdfd78af_1	bioconda/noarch
+ bioconductor-genomeinfodbdata	1.2.11	r43hdfd78af_1	bioconda/noarch
+ bioconductor-genomicranges	1.54.1	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-iranges	2.36.0	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-matrixgenerics	1.14.0	r43hdfd78af_3	bioconda/noarch
+ bioconductor-s4arrays	1.2.0	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-s4vectors	0.40.2	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-singlecellexperiment	1.24.0	r43hdfd78af_0	bioconda/noarch
+ bioconductor-sparsearray	1.2.2	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-summarizedexperiment	1.32.0	r43hdfd78af_0	bioconda/noarch
+ bioconductor-xvector	0.42.0	r43ha9d7317_2	bioconda/linux-64
+ bioconductor-zlibbioc	1.48.0	r43ha9d7317_2	bioconda/linux-64

If you scroll down, you'll see this list also includes r-base (which is R) as well as other R packages.

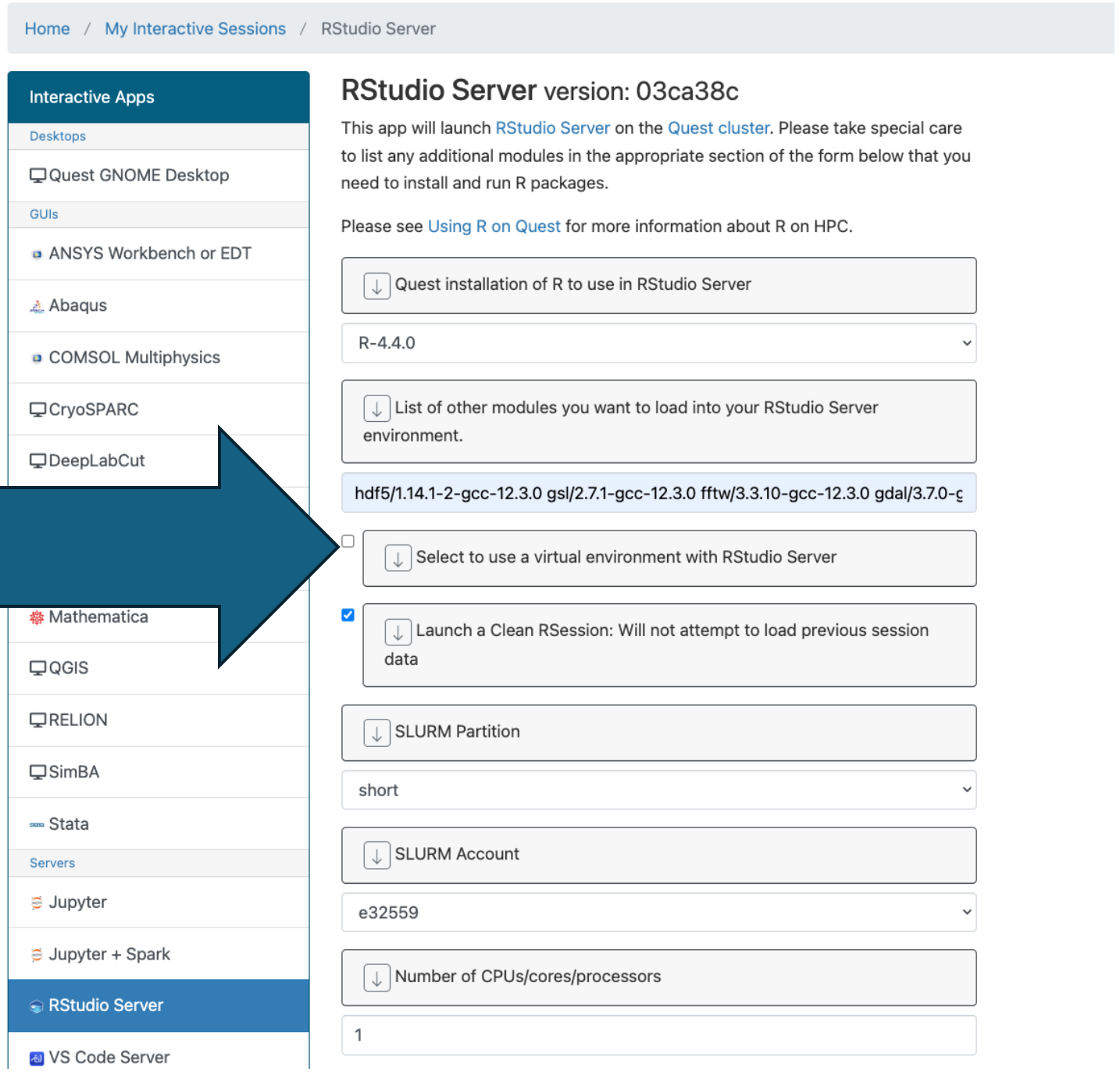
Hit Y and then enter.

This will also take a while, be patient.

Using a virtual environment with QOD

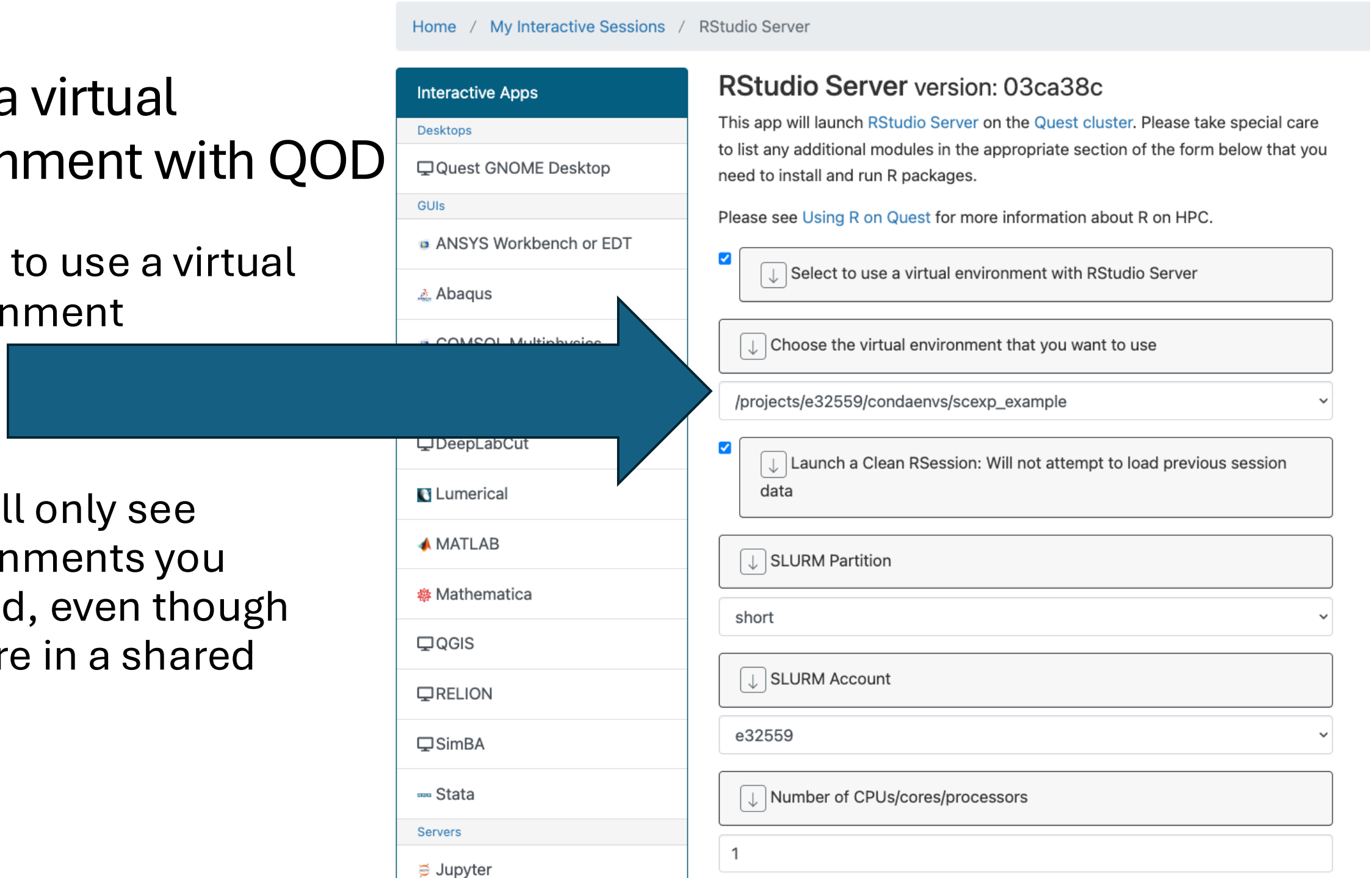
- Select to use a virtual environment with RStudio Server

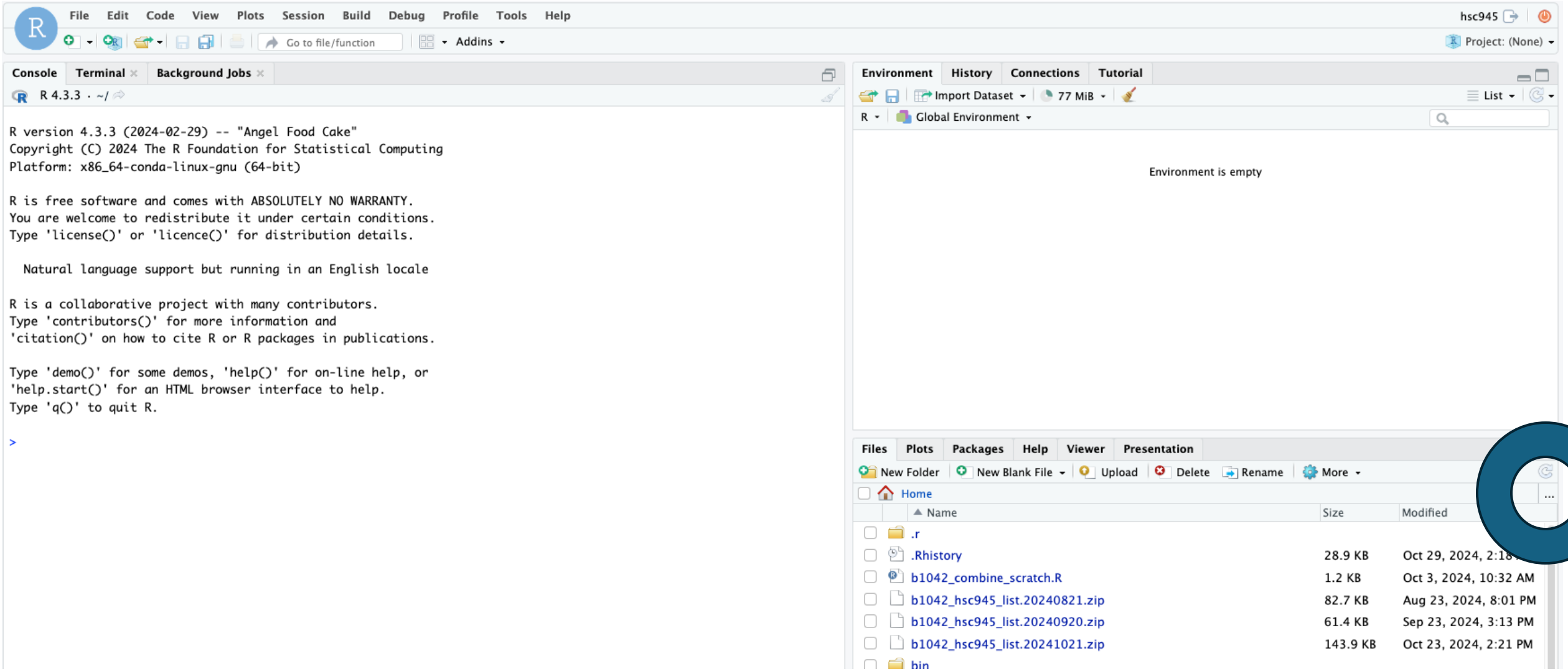
- This will remove the modules, as everything should be in the environment.
- Change the job name so you can keep track!

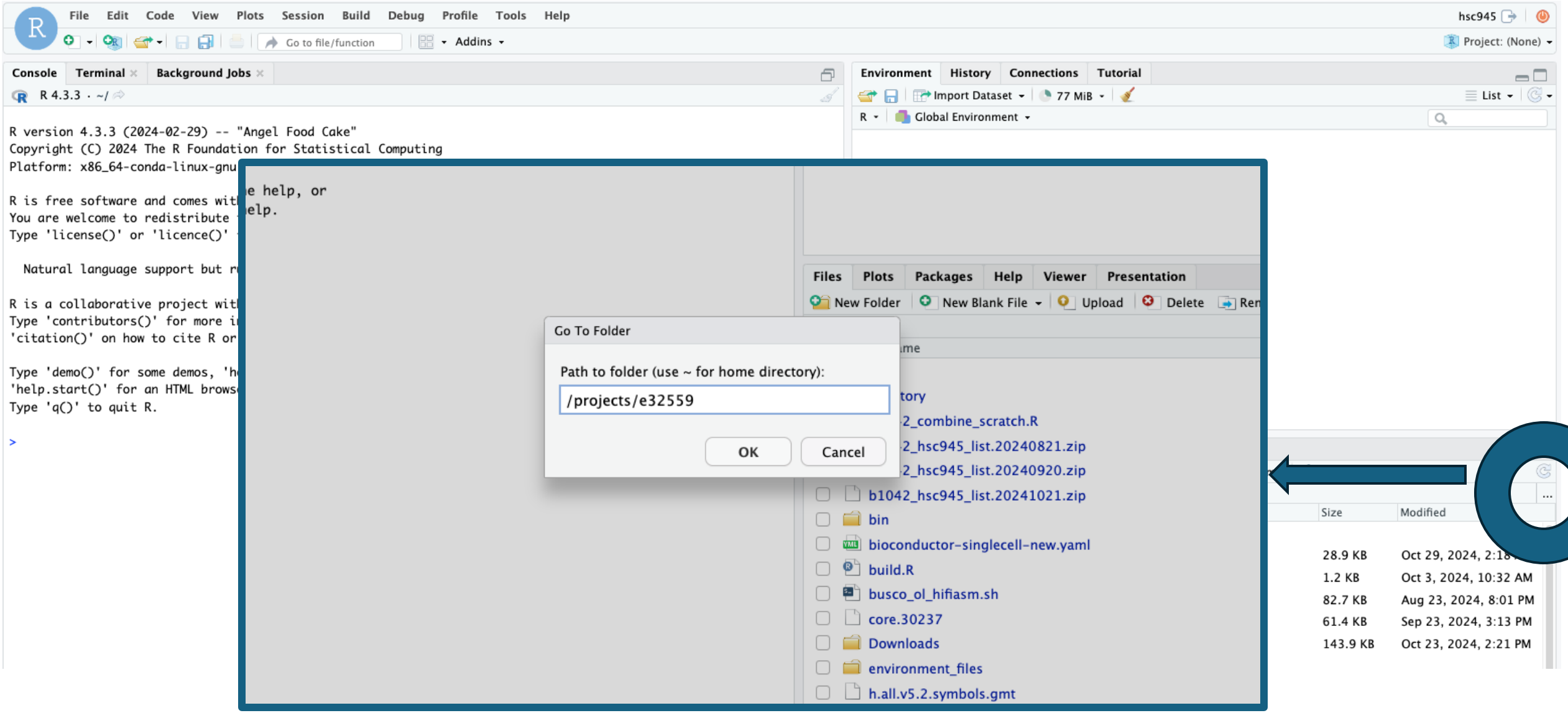


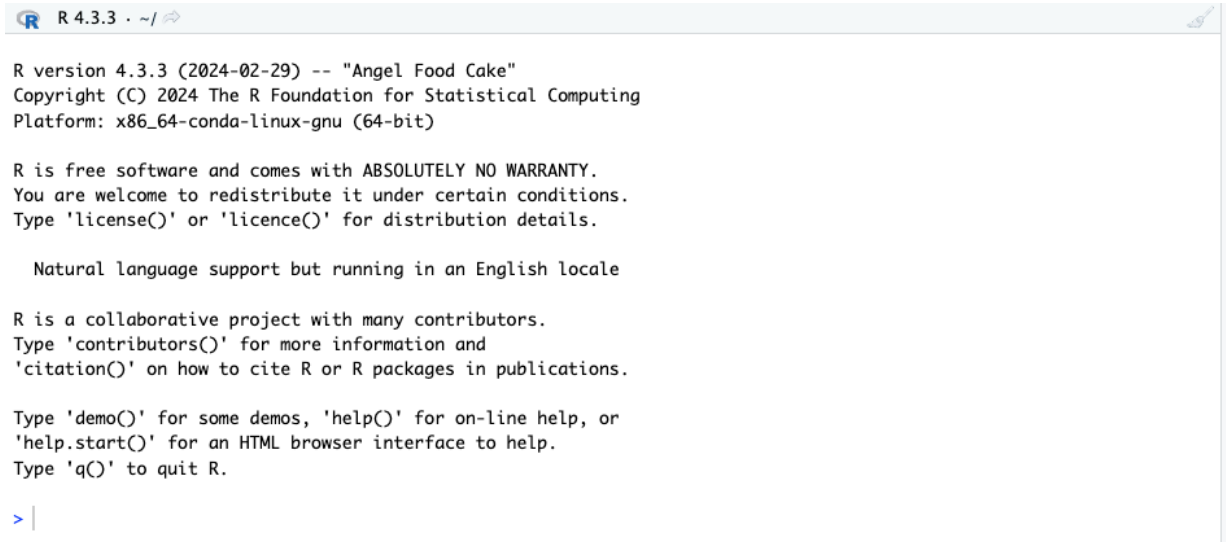
Using a virtual environment with QOD

- Select to use a virtual environment
- You will only see environments you created, even though they are in a shared folder.

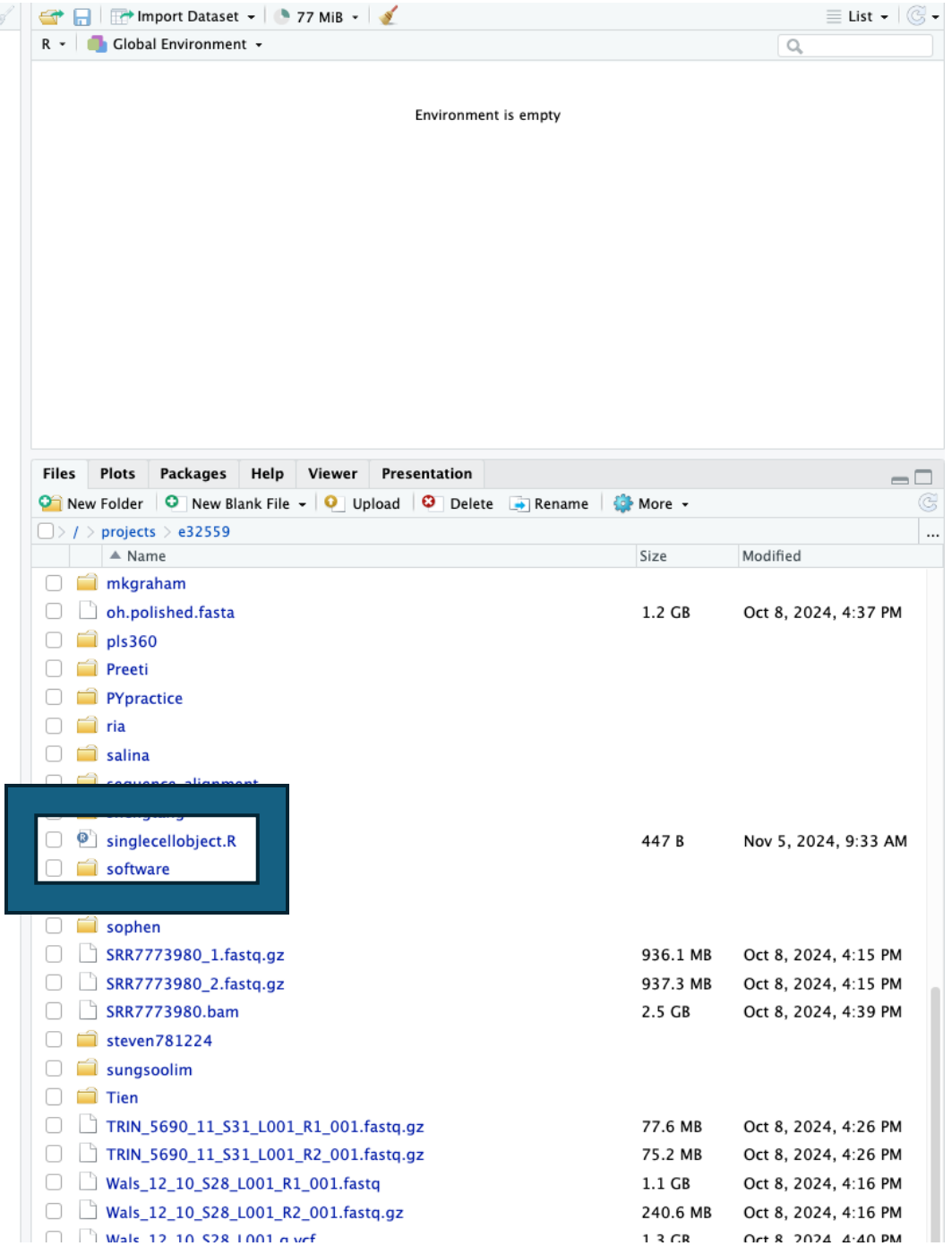



File Edit Code View Plots Session Build Debug Profile Tools Help
hsc945
Go to file/function
Addins
Project: (None)
Console
Terminal
Background Jobs
R 4.3.3 · ~/
R version 4.3.3 (2024-02-29) -- "Angel Food Cake"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-conda-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment
History
Connections
Tutorial
Import Dataset
77 MiB
List
R
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Presentation
New Folder
New Blank File
Upload
Delete
Rename
More
Home
Name
Size
Modified
.r
.Rhistory
28.9 KB
Oct 29, 2024, 2:18
b1042_combine_scratch.R
1.2 KB
Oct 3, 2024, 10:32 AM
b1042_hsc945_list.20240821.zip
82.7 KB
Aug 23, 2024, 8:01 PM
b1042_hsc945_list.20240920.zip
61.4 KB
Sep 23, 2024, 3:13 PM
b1042_hsc945_list.20241021.zip
143.9 KB
Oct 23, 2024, 2:21 PM
bin


File Edit Code View Plots Session Build Debug Profile Tools Help
hsc945
Go to file/function
Addins
Project: (None)
Console
Terminal
Background Jobs
R 4.3.3 · ~/
R version 4.3.3 (2024-02-29) -- "Angel Food Cake"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-conda-linux-gnu
R is free software and comes wit
You are welcome to redistribute
Type 'license()' or 'licence()'
Natural language support but r
R is a collaborative project wit
Type 'contributors()' for more i
'citation()' on how to cite R or
Type 'demo()' for some demos, 'h
'help.start()' for an HTML brows
Type 'q()' to quit R.
>
Environment
History
Connections
Tutorial
Import Dataset
77 MiB
List
R
Global Environment
Files
Plots
Packages
Help
Viewer
Presentation
New Folder
New Blank File
Upload
Delete
Go To Folder
Path to folder (use ~ for home directory):
/projects/e32559
OK
Cancel
2_combine_scratch.R
2_hsc945_list.20240821.zip
2_hsc945_list.20240920.zip
b1042_hsc945_list.20241021.zip
bin
bioconductor-singlecell-new.yaml
build.R
busco_ol_hifiasm.sh
core.30237
Downloads
environment_files
h.all.v5.2.symbols.gmt
Size
Modified
28.9 KB
1.2 KB
82.7 KB
61.4 KB
143.9 KB
Oct 3, 2024, 10:32 AM
Aug 23, 2024, 8:01 PM
Sep 23, 2024, 3:13 PM
Oct 23, 2024, 2:21 PM



You should be able to open and execute the example code in this file now with your virtual environment. If you'd like to learn more about this example, it was taken from a SingleCellExperiment vignette: https://www.bioconductor.org/packages/release/bioc/vignettes/SingleCellExperiment/inst/doc/intro.html



To use this environment with Jupyter we would need to add one more thing…

source activate /home/<netid>/.conda/envs/scexp

mamba install -c conda-forge jupyter r-irkernel

R -e “IRkernel::installspec(name=‘scexp’, displayname=‘Rscexp’)”

More resources

- Youtube video on installing R packages generally: https://www.youtube.com/watch?v=GM6MCBkVNtQ
- Bioconductor website: https://www.bioconductor.org/
- Quest Knowledge Base article on troubleshooting R installs: https://services.northwestern.edu/TDClient/30/Portal/KB/ArticleDet?ID=1834
- Quest Knowledge Base article on using R on Quest: https://services.northwestern.edu/TDClient/30/Portal/KB/ArticleDet?ID=1556#:~:text=R%20packages%20intended%20for%20use,the%20details%20for%20further%20assistance.