About R and R Studio Server

R version updates August 2023

After the August 15, 2023 maintenance the default R version is **R 4.3.1** both for RStudio Server on all compute servers, and on the command line (typing **R** or **Rscript** on the command line will use **R 4.3.1**). The **RStudio Server** application version itself was also upgraded.

To make sure your RStudio Server session picks up the version change, do both of the following:

- Start a new session (red/orange "power on" button  at the top right
- Logout then log back in to the RStudio web application (arrow on paper icon  at the top right)

> The R version upgrade may cause issues if you have installed many R packages locally. This can happen because your local package version is no longer compatible with the latest version of some (e.g. Bioconductor) program we have installed, but since your local installs take precedence they shadow the more up to date globally installed version. See Local/Global package installation conflicts below for more information.

R and RStudio Server R versions

The issue of R versions is a difficult one, especially now that many important single-cell packages are only available in newer R versions, but not all older (but still popular) R packages are. This section describes the versioning issues in both the system R and in the **RStudio Server** web application.

After the August 2023 maintenance the default R version is **R 4.3.1** both for RStudio Server on all compute servers, and on the command line (typing **R** or **Rscript** on the command line will use **R 4.3.1**).

We also have other versions of R installed "side by side" on the command line – **R-4.0.3** and **R-3.6.1** – which can be accessed by typing the specific version from the command line (e.g. **R-4.0.3**, **Rscript-3.6.1**) However multiple R versions are not available in **RStudio Server** because its R version setting can only be set to ***one*** value system-wide and cannot be specified per-user.

All POD compute servers now use **R 4.3.1** as the R version in their **RStudio Server** web application. If your POD has multiple compute servers and you would like one to run a different R version, please contact us at rctf-support@utexas.edu.

We have also installed many popular add-on packages in all the R versions (e.g. **tidyverse**, **ggplot2**, **DESeq2**); however be aware that not all packages are available in all R versions.

Other R versions

If you need a GUI environment to access versions of R other than 4.3.1 an option that provides maximum per-user flexibility is as follows. Use the **RStudio Server** web application for R 4.3.1-compatible workflows. For workflows requiring other R versions, users can install a different version of **R** on their own desktop/laptop computers along with the **RStudio** desktop application. Then users can access files on shared storage by mounting their Work area file system via Samba (see Samba remote file system access for more information). The main drawback to this workflow is that typical personal computers do not have as much RAM as POD compute servers, and some R tasks can be memory intensive. What users can do in such cases is test the code in **RStudio** on their desktop computer, using smaller data sets if necessary. Then run the "full" workflow from the POD compute server command line using the appropriate R version (you can locally install newer R versions yourself).

Understanding R add-on packages

The libraries/add-on packages available in any given R version depend on the configured package installation directories, which can be listed in the R environment via the **.libPaths()** function. Typically, each user has a local package installation directory with packages they have installed. This local directory is searched first, followed by one or more system directories where we have installed add-on packages system-wide.

User local package installations directories are typically under the user's **~/R** directory (e.g. **/stor/home/<user_name>/R**). If a user has installed packages under multiple versions of R, there will be sub-directories for the different versions (e.g. **~/R/x86_64-pc-linux-gnu-library/3.6**, **~/R/x86_64-pc-linux-gnu-library/4.0**). Users can list the contents of these directories to see what packages they have installed locally.

To see what packages are installed system-wide for a given R version, users can look at the version's package installation directories:

- **3.6.3**
  - **/usr/lib/R/library**
  - **/usr/lib/R/site-library**
- **R 4.0.3** – **/stor/system/opt/R/R-4.0.3/lib/R/library**
- **R 4.3.1** – **/stor/system/opt/R/R-4.3.1/lib/R/library**

Local/Global package installation conflicts

Globally-installed R add-on packages may be updated during system maintenance. This can sometimes cause problems when users invoke R tools with many dependencies (e.g. **DESeq2**), some of which have been updated system-wide, but others of which have been locally installed and are not at a compatible level. The resulting error messages can be rather obscure, but typically show up after system maintenance has been performed. For example:

```
Error in checkSlotAssignment(object, name, value) :
  assignment of an object of class “NULL” is not valid for slot ‘NAMES’ in an object of class “DESeqDataSet”; is
(value, "characterORNULL") is not TRUE
```

To determine if this is due to a Local/Global package conflict, users can make their local installation directory invisible to R and see if the error goes away like this:

```
mv ~/R ~/R.bak
```

If this produces a different error indicating that one or more locally installed packages are missing, the user can re-install them then see if the problem is resolved. Check the now-named **~/R.bak/x86_64-pc-linux-gnu-library/x** directory, where **x** is the R version being used to see the packages that were locally installed previously. Even if this resolves the immediate issue, the user may later find that they need to re-install other packages that were previously installed locally.

Finally, if renaming the local R installation directory does not resolve the issue, it may be an issue with the globally installed packages, so Contact Us.

Troubleshooting other R/RStudio server issues

In addition to the Local/Global package conflict issue described above, other issues can arise involving **RStudio Server** (or less commonly, command-line R). If all else fails, submit a help request to our **rctf-support@utexas.edu** support email.

RStudio Server becomes unresponsive

One common problem is that **RStudio Server** may become unresponsive, even with repeated attempts to establish a new session. To troubleshoot this sort of issue, close the **RStudio Server** application and make some R-associated files and directories invisible to R like this:

```
mv ~/.rstudio ~/.rstudio.bak
mv ~/.RData ~/.RData.bak
mv ~/.local/share/rstudio ~/.local.rstudio.bak
```

Note that **.RData** files may be in different directories. For example, if you a working in an R Project you have set up, there may be an **.RData** file in the project directory.

Large **.RData** files can be extremely slow to load from both **R** and **RStudio Server**. If you must save R data this way, consider renaming the **.RData** file to a different name so that it can be loaded explicitly only when needed, instead of always when R is invoked.

Browser issues

A number of other issues can occur when attempting to use RStudio server in a browser. Most commonly these occur when first connecting to the server, but can also occur at later stages of use.

One common error is the "**This site can't be reached**" browser error (Windows), which may indicate a browser-related security issue.

For this and other browser related problems:

- Close all browser windows and restart your computer.
  - Fragmented RAM (computer memory) can sometimes cause problems.
- Make sure the browser software being used is the latest version.
  - Version checks are usually found in the browser's "Help" "About" dialog, accessible from the options menu in the upper right of a browser's taskbar.
- Try different browsers (e.g. Firefox instead of Chrome, Microsoft Edge, or Safari).

    - If it is a security issue, a different browser may indicate that there are security risks accessing the website, but give the option to accept the risk and go ahead.
- Try accessing the site from an "Incognito" or "Private" browser window (usually found in an options menu in the upper right of a browser's taskbar.
    - This can bypass browser cache or cookie issues that can interfere with connections.
- Clear the browser's Cookie cache

If problems persist, please email our **rctf-support@utexas.edu** support email, including a description of what error or issue you are experiencing.

EDU pod connection issue

An issue on the EDU pod may arise when students use the **edupod.cns.utexas.edu** virtual host to access RStudio server. The **edupod.cns.utexas.edu** virtual host acts as a front end load balancer for the request, and forwards it to one of the back-end compute servers. If this occurs, try accessing individual servers specifically:

- https://educcomp01.ccbb.utexas.edu
- https://educcomp02.ccbb.utexas.edu
- https://educcomp04.ccbb.utexas.edu

If accessing a specific server works when using the virtual host does not, please let us know by emailing our **rctf-support@utexas.edu** support email.

Zombie processes

A "zombie" RStudio process may be preventing you from successfully logging in to an RStudio session. If your POD has more than one compute server, try logging in to one of the others. If that works, the issue is related to something on the compute server you can't login to.

You can get rid of any zombie RStudio processes by using **ps -ef | grep <user_name>** to see processes associated with your account, then kill any RStudio processes using **kill -9 <pid>**, where <pid> is the 1st number after your account name in the **ps** list. E.g.:

```
ps -ef | grep abattenh
```

Can report information like this:

```
# output from ps -ef | grep abattenh:
abattenh   65327   5968  0 Aug01 ? 00:00:49 /usr/lib/rstudio-server/bin/rsession -u abattenh --session-use-
secure-cookies 0 --session-root-path / --session-same-site 0 --session-use-file-storage 1 --launcher-token
F62AEC5D --r-restore-workspace 2 --r-run-rprofile 2
```

Here the pid is 65327, and the process can be killed like this

```
kill -9 65327
```

File upload issues

Sometimes users will get an "Unexpected response from server" error when attempting to upload a file. Here are some troubleshooting tips:

- Try the Browser issues troubleshooting tips, especially:
    - Make sure your browser is up to date with the most current version
    - Try the operation in at least 2 different up-to-date browsers (e.g. Chrome and Firefox)
    - Try the operation in a private/incognito/safe browser session
- Try the upload operation from the JupyterHub web application instead of from RStudio
- Try connecting to a specific back-end compute server as described at EDU pod connection issue

Disk quota exceeded

Another type of problem can arise when a user's 100 GB Home directory quota has been exceeded (not applicable on the EDU pod, which does not have Home directory quotas). This can produce errors when trying to start **RStudio Server** or R, perform work in R, or even install additional packages. For example, you may see a "**Cannot connect to service**" message after logging in to **RStudio Server**. Or, if an R session has been established and saving a new file would exceed the Home directory quota, users will often (but not always) see an error like the following:

```
cannot create file'/stor/home/abattenh/output.tsv', reason 'Disk quota exceeded
```

To determine the status of your Home directory quota, just use SSH to login to one of your POD's compute servers. A message such as the one below will be displayed:

```
Quota Report for abattenh
Mount Point          Used           Total      Last Checked
stor/home/abattenh   52G (51%)      100G       Mon 21 Sep 2020 11:32:02 AM CDT
```

If this issue arises, you should Contact Us to help relocate some of your Home directory contents to your Work or Scratch area. Just moving them yourself does not resolve the problem because Home directories have frequent snapshots taken that preserve copies of deleted files, and it requires a systems administrator to remove these snapshots (see Home directories for more information).

This issue can arise because R's default input/output directory is the user's home directory – but large files should not be stored or created there due to the 100 GB quota. Instead, R processing of large files should take place in the user's Work or Scratch area (e.g. **/stor/work/<user's group name>** or **/stor/scratch/<user's group name>**; users can find out which group(s) they belong to by typing the **groups** command on the command line). Users can navigate to Work or Scratch area directories using R's **setwd** function or using **RStudio Server**'s file browser (e.g. via "Session" menu "Set Working Directory" "Choose Directory", or when a new R Project is created). Note that **RStudio Server**'s file browser dialog will default to the user's Home directory, and the full path of the desired Work or Scratch area must be typed in.