

HPC Software Environment

Consult [Desktop Software](#) for information on connecting to HPRC resources.

Click on the following two links to provide expanded information related to the title.

You will need to submit a service-now *request* to potentially get software installed onto the HPC cluster. There are some caveats to be aware of:

- You may be responsible for covering initial and ongoing costs of commercially available software.
- You are expected to comply with any terms and/or agreements associated with software you use.
- Windows emulation environments are not installed on the HPC cluster. This decision will be reversed if/when suitable security controls are in place. Access to Windows session hosts will be configured for researchers wishing to use software written for Microsoft Windows operating system.
- JCU doesn't have server-grade infrastructure to support running Apple/MacOS software.

As at 01-Mar-2021, users can choose to install software with one of their directories. You should only be doing this for software you have developed. Researchers choosing to do this, do so at their own risk and on their own time - you are responsible for any issues associated with software you develop/install. Viewed across JCU, individual software installs are not a demonstration of JCU's sustainability core value.

Technology Solutions security staff have/are pushing for a "no execute" setting be placed on all user filesystems. **IF** HPC staff are directed to make this change, you will not be able to run software you have installed into one of the directories you own - exceptions might be granted following a threat analysis. I believe this is largely due to the threat and impacts associated with ransomware.

Environment Modules

JCU HPC uses [environment modules](#) to deliver multiple versions of software in a multi-user HPC cluster.

As a general rule, you can view a list of higher-level software installed on the HPC cluster using

```
module avail
```

For more information on environment modules commands, see [Environment modules cheat sheet](#)

Extra Packages/Add-ons Information

Some of the software environments installed on the HPC cluster have significant packages added to the base. The information below is provided for the more commonly/widely used software environments.

The version of python3 provided by RedHat Enterprise Linux 7 (`/usr/bin/python3`) does not include any packages not released by RedHat - to maintain operating system integrity. Research should use versions of python available through environment modules and/or anaconda. To view an up-to-date list of addons available for python 3.6.5 (using environment modules), use the following command:

```
module load python3/3.6.5
pip3 list --format=columns
```

IMPORTANT: Python 2.7 has hit end of life and should *only* be used if functionality isn't available in Python 3. HPC staff may be directed to remove all software based on python2, for security/risk reasons.

To view an up-to-date list of addons available for python 2.7.14, use the following command:

```
module load python/2.7.14
pip list --format=columns
```

There are three R installs (Dec-2020) available for use. Each one will have a different set of packages installed. To list the available packages, run the relevant command(s) below.

R 4.0.3:

```
module load anaconda3
ls $CONDA_HOME/envs/R-403/lib/R/library
```

R 3.6.x (conda):

```
module load anaconda3
ls $CONDA_HOME/envs/R-36/lib/R/library
```

R 3.6.1:

```
module load R/3.6.1
ls $R_HOME/lib64/R/library
```

Note: The non-conda version of R is likely to be removed at some point in the future.

JCU has recently purchased *campus wide* licensing for MATLAB. This license gives all current JCU staff/students rights to use MATLAB and **all** its toolboxes on any platform, including personal computing devices. Details on toolboxes etc. available with this license can be found at <https://au.mathworks.com/products.html>.

MATLAB 2020a and earlier versions remain tied to perpetual licenses purchased by HPC. The JCU HPC perpetual license has the following licensing limits:

- **50 User License (core):** MATLAB, MATLAB Coder, Simulink, Simulink Coder, Simulink Control Design
- **50 User License (toolboxes):** Control System, Optimization, Signal Processing, Symbolic Math, System Identification
- **11 User License (toolboxes):** Statistics & Machine Learning
- **10 User License (toolboxes):** Global Optimization, Parallel Computing
- **5 User License (toolboxes):** Mapping, Neural Network
- **4 User License (core):** MATLAB Compiler
- **4 User License (toolboxes):** Fuzzy Logic, Image Processing, Wavelet
- **2 User License (toolboxes):** Audio System, DSP System, Curve Fitting, Instrument Control, Partial Differential Equation, Simscape, Simscape Power Systems
- **1 User License (toolboxes):** Aerospace

It is my understanding that one (1) **User License** equates to "one user, one computer, using no more than 2 CPU cores".

[HPC conda3 software environments](#)

[HPC conda2 software environments \(LEGACY\)](#)

Software supplied by RedHat

Using software supplied by RedHat is generally safest, however it comes with some gotchas:

- The version is often well behind latest.
- Extra packages/addons are not installed, due to potential to manage core operating system functions.
- HPC staff have created environment modules for RedHat supplied software, but they don't make any environment changes to your account.

In order to minimize costs associated with porting Bioinformatics & related software to RedHat Enterprise Linux, a lot of software will only be available as a **conda** environment.

Future Possibilities

JCU HPC currently doesn't provide access to containers (docker, podman, or singularity) centrally. One consideration is the added complexity to commands required to get work done. For example, a traditional run of an analysis with `salmon` is

```
salmon index -t athal.fa.gz -i mus.rna.index
```

To get the same analysis done inside a docker container, the line might become (in future)

```
sudo docker run -v `pwd`: /scr -w /scr -u $(id -u):$(id -g) combinelab/salmon salmon index -t athal.fa.gz -i mus.rna.index
```

Provision of containerised software environments in future requires security controls to be approved and in place. The potential security controls include; limiting command usage, firewall controls that only allow access to "trusted sources", resources to assess threats/impacts of requirements that come from "untrusted sources".