Using and Installing Custom Software on HARDAC
Part 1 - Environment Modules
Goals of Class

Be able to

• Use cluster-provided software
• Install and use R packages
Class Requirements

Access to the HARDAC cluster

• Email gcb-help@duke.edu to request access if necessary

Complete previous gcb-academy course

• High Performance Computing (HPC) / SLURM Best Practices for HARDAC
Cluster Software Challenges

• Conflicting version and dependency requirements
• Ability to retain all versions of software used in analyses

❌ Global software installation
✓ Environment Modules
Environment Modules

- Software packages already on the HARDAC cluster
- Must be activated to use
- Commonly used on computing clusters
- Only higher profile software packages are provided
Class Exercises Directory

Create a directory to hold files for this class

```
$ mkdir software_class
```

Copy example fasta data files into your new directory

```
$ cp /data/common/software_class/*.fasta software_class
```

Change to the new directory

```
$ cd software_class
```
Environment Module commands

The **module** command updates your shell environment adding and removing directories containing software packages

- List available modules - `module avail`
- Search modules - `module avail <searchTerm>`
- Activate a module - `module load <moduleName>`
- Deactivate a module - `module unload <moduleName>`
- Deactivate all loaded modules - `module purge`
## List Environment Modules

```bash
$ module avail
```

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Version</th>
<th>ainer Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmpliconNoise</td>
<td>1.27-fasrc01</td>
<td></td>
</tr>
<tr>
<td>Anaconda/1.9.2-fasrc01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anaconda/2.1.0-fasrc01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anaconda/2.5.0-fasrc01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anaconda/4.3.0-fasrc01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anaconda3/2.1.0-fasrc01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anaconda3/4.3.0-gcb01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anaconda3/5.0.1-gcb01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...  

**NOTE:** The module command does not work on the login node.
$ module avail blast

------------------------------------- /data/itlab/helmod/modulefiles/Core -------------------------------------
  ncbi-blast/2.2.22-fasrc01          samblaster/0.1.24-gcb01
  ncbi-blast/2.3.0+-gcb01           (D)

Where:
  (D): Default Module
  ...

Pause the video and try it yourself
Use an Environment Module

load environment module

```
$ module load ncbi-blast/2.3.0+-gcb01
```

verify command is working as expected

```
$ blastp -h
USAGE
    blastp [-h] [-help] [-import_search_strategy filename]
          [-export_search_strategy filename] [-task task_name] [-db database_name]
...
```

cleanup - purge environment modules

```
$ module purge
$ blastp -h
bash: blastp: command not found
```
Environment Module sbatch script

Make a file called blastp.sh

```bash
#!/bin/bash
module load ncbi-blast/2.3.0+-gcb01
blastp -query query.fasta -subject subject.fasta -out result.txt
```

Run blastp.sh using the **sbatch** command

```bash
$ sbatch blastp.sh
Submitted batch job 23231383
```

Look at results (once job completes)

```bash
$ cat result.txt
```
R Environment Module

Comes with many R packages already installed

Allows installing additional R packages

• Requires a directory to hold the packages

• Must update the `R_LIBS_USER` environment variable
Create a directory named "envs" in your lab's data directory:

```
$ mkdir /data/<LabName>/envs
```

We will store custom environments such as a directory of R packages within this directory.

NOTE: In this course when you see `<LabName>` replace it with your lab's name.
Install an R package

Create a directory to hold R packages

```bash
$ mkdir /data/<LabName>/envs/r_libs
```

Load R module

```bash
$ module load R/3.6.1-gcb03
```

Add your directory path to the beginning of R_LIBS_USER

```bash
$ export R_LIBS_USER=/data/<LabName>/envs/r_libs:$R_LIBS_USER
```

Install the "here" R package

```bash
$ R
install.packages('here')
q()
```
Verify the R Package

Run R interpreter and test the package

```r
$ R
library(here)
here()
q()
```
Use the R Package - part 1

Create a file named here.R with contents:

```r
library(here)
here()
sessionInfo()
```

Create a file named here.sh with contents:

```bash
#!/bin/bash
module load R/3.6.1-gcb03
export R_LIBS_USER=/data/<LabName>/envs/r_libs:$R_LIBS_USER
Rscript here.R
```
Use the R Package - part 2

Run here.sh using the `sbatch` command

```bash
$ sbatch here.sh
Submitted batch job 23231383
```

Verify expected output

```bash
$ cat slurm-...
R version 3.6.1...
...
[1] "/gpfs/fs1/home/jpb67"
...
Real World Example - Seurat

• Review the software installation instructions
• Look for environment modules that contain the software or its requirements
• Install Software
• Verify Software
Real World Example - Seurat Documentation

Installation Instructions for Seurat

1. Install R, version 3.6 or greater
2. Recommended: Install R Studio

v4.0 Beta  CRAN (Version 3.2.1)  Previous Versions  Development Version

Install from CRAN

Seurat is available on CRAN for all platforms. To install, run:

```r
# Enter commands in R (or R studio, if installed)
install.packages('Seurat')
library(Seurat)
```
Real World Example - Requirements

Is there a seurat module?

```
$ module avail seurat
```

If no seurat module .. is there a R 3.6+ module?

```
$ module avail R
```
Real World Example - Install Seurat

Install using the same method as before

```
$ module load R/3.6.1-gcb03
$ export R_LIBS_USER=/data/<LabName>/envs/r_libs:$R_LIBS_USER
$ R
install.packages('Seurat')
```

NOTE: Installing/Compiling software can take a while
Real World Example - Verify Seurat

Try first few lines from a Seurat tutorial

```r
library(dplyr)
library(Seurat)
library(patchwork)

# Load the PBMC dataset
pbmc.data <- Read10X(data.dir = "../data/pbmc3k/filtered_gene_bc_matrix")
# Initialize the Seurat object with the raw (non-normalized data)
```
When Software Installations Fail

Conda Environments
- presented in HARDAC Software Part 2

Singularity Images
- presented in HARDAC Software Part 3
Helpful Resources

HARDAC WIKI

https://wiki.duke.edu/display/HAR/

Requesting Help or Video Feedback

gcb-help@duke.edu