Introduction to Bioinformatics on Unity I

Cecile Cres & Anna Schrecengost
February 26 2024

UNITY

W: https://unity.rc.umass.edu
E: hpc@umass.edu
High-Performance Computing and Biology
Outline

- Accessing Unity
- Data storage
- Transferring data to and from Unity
- Bioinformatics tools available on Unity
- Installing softwares via conda, apptainer and R
- Biological datasets
- Submitting jobs on Unity
Different ways to access Unity
Signing up for an account

Full tutorial/demo available on the [Unity Onboarding video (Spring 2024)](#)

1. Go to [unity.rc.umass.edu](http://unity.rc.umass.edu)
2. Log in/request account → Log in with institution SSO → My PIs → input PI NetID
   - Need to be a PI or part of PI group
   - If PI: Account settings → request PI account
Connecting on CLI via SSH

- SSH = secure shell
- Key-based authentication
- Can generate keys on Unity portal or on command line
- Unity username is based on email:
  
  ```
  ssh <username>_<organization>_edu@unity.<organization>.edu
  ```
  
  ```
  ssh <username>_uri_edu@unity.uri.edu
  ```
Connecting via SSH

- Create config file on your computer which contains information for connecting to unity
- Full details can be found [here](#)
Last login: Sun Feb 18 11:28:16 on console
[(base) annaschrecengost@Annas-MacBook-Pro ~ % ssh aschrecengost_uri_edu@unity.uri.edu
Enter passphrase for key '/Users/annaschrecengost/.ssh/id_rsa':

[UNITY]

W: https://unity.rc.umass.edu

Support
The Unity Community Slack is a place to chat with staff & other users.
   * https://unity.rc.umass.edu/community-slack
Office hours are held from 2:30-4 PM EDT on Tuesdays via Zoom.
   * https://unity.rc.umass.edu/office-hours
To make a support ticket, send us an email at hpc@umass.edu

Login Node Usage
The login nodes have strict CPU and memory limits.
You can switch over to a compute node with the `unity-compute` command.
Learn more: https://docs.unity.rc.umass.edu/documentation/jobs/

Storage Redundancy
Your storage on Unity has temporary (3 days) snapshots but no backups.
It's recommended that you copy your results back to local storage.
Learn more: https://docs.unity.rc.umass.edu/documentation/managing-files/

/home/aschrecengost_uri_edu  13G (26%) used of 50G
/work/pi_rbeinart_uri_edu    61G (7%) used of 1000G
Connecting via OnDemand

- Simplest way to connect – no SSH keys
- Sign in on browser using organization SSO
- Can access the Unity login shell this way
- Can also access the filesystem and interactive Apps
  - RStudio, JupyterLab, MATLAB, Mathematica, Unity Desktop
OnDemand provides an integrated, single access point for all of your HPC resources.

This new interface is able to replace your terminal, your FTP client, and JupyterHub. You can learn how to use it on the Unity docs (Getting Started), but it is mostly straightforward. It has many features which have the potential to make your life easier:

**Active job viewer**
See information like job status, allocated resources, time remaining.

**Job builder/templatizer**
Build and submit your batch jobs with a GUI rather than the CLI.

**File explorer & Text Editor**
Manage your files with a GUI rather than the CLI.

**Desktop environment**
Graphical applications like Mathematica are now easily accessible. Spawn multiple windows, drag and drop files, use a taskbar and start menu, view images. Xfce is a fully fledged desktop environment like those of Windows and MacOS, though it lacks their visual appeal and can be confusing to learn.

**JupyterLab**
This should be similar to JupyterHub (without Rstudio/Matlab), but with a few improvements:
- Environment module support
- Access to Anaconda packages
- The file browser now starts at your home directory, which means you won't get permission denied when you try to create a new file
- When your job is in the queue, you aren't left staring at a blank white screen
- More robust job builder with support for custom Slurm arguments
- Debug mode

**Rstudio**
This should be similar to JupyterHub Rstudio, but with a few improvements:
- Rstudio is containerized, and the new image include tidyverse, devtools, tex, geospatial, and cuda
- Previous R modules (r-st, r-terra, etc.) are now obsolete, they are pre-installed
- More robust job builder with support for custom Slurm arguments
- Debug mode
Data storage

- /home directory → 50 GB quota
- /work/pi_ → PI's work directory → 1TB quota
- /project → available to PIs upon request → quota depends on storage needed
- /scratch → scratch space → no quota

More info on Unity docs
Transferring data to & from Unity
Transferring data using CLI - scp

- These CLI tools are best for small/quick data transfers
- Do not need to install scp
- Copy files or a whole directory to/from Unity
- Add "-r" flag after scp to copy contents of entire directory

```
scp path/to/source username_uri_edu@unity.uri.edu:/path/to/destination
```

```
scp username_uri_edu@unity.uri.edu:/path/to/source path/to/destination
```
Transferring data using CLI - scp

- If have created config file can simplify:

```
scp path/to/source unity:/path/to/destination
```

Local computer  Remote computer (Unity)

```
scp unity:/path/to/source path/to/destination
```
Transferring data using CLI - rsync

- Mac and Linux users can install
- Add "-r" or "-a" flag after rsync to copy contents of entire directory
- Resume interrupted downloads using flag "--partial"
- Compress files during transfer with "-z"

```
rsync path/to/source unity:/path/to/destination
```

```
rsync unity:/path/to/source path/to/destination
```
Transferring data using CLI - rclone

- Command-line utility for managing files in cloud storage
- Can use to transfer large files from Unity to a sequencing database such as ncbi SRA, for example
- Unity provides rclone - can use on CLI or also as an interactive app on Unity OnDemand
- Rclone documentation is [here](#)
Transferring data using GUI - Cyberduck

- Transfer small amounts of data to/from cluster via user-friendly interface
- Can also be used for most types of file transfer (e.g. local storage, cloud storage, etc.)
- Set-up tutorial is available [here](#)
- FileZilla is a similar client - tutorial is available [here](#)
Transferring data using Globus

- Best option for transferring large amounts of data between clusters
- Can also transfer between other Globus endpoints, or Unity and your local machine (e.g. backup to external hard drive)
  - To transfer to/from local machine, must install Globus Connect Personal
- To get started, go to app.globus.org
- Tutorial available here
Transferring data using Unity onDemand

- Simplest way to transfer files - does not require use of CLI or ssh configuration
- Cannot transfer large files - use for files less than 5GB
OnDemand provides an integrated, single access point for all of your HPC resources.

This new interface is able to replace your terminal, your FTP client, and JupyterHub. You can learn how to use it on the Unity docs (Getting Started), but it is mostly straightforward. It has many features which have the potential to make your life easier:

**Active job viewer**
See information like job status, allocated resources, time remaining.

**Job builder/templatener**
Build and submit your batch jobs with a GUI rather than the CLI.

**File explorer & Text Editor**
Manage your files with a GUI rather than the CLI.

**Desktop environment**
Graphical applications like Mathematica are now easily accessible. Spawn multiple windows, drag and drop files, use a taskbar and start menu, view images. Xfce is a fully fledged desktop environment like those of Windows and MacOS, though it lacks their visual appeal and can be confusing to learn.

**JupyterLab**
This should be similar to JupyterHub (without Rstudio/Matlab), but with a few improvements:
- Environment module support
- Access to Anaconda packages
- The file browser now starts at your home directory, which means you won’t get permission denied when you try to create a new file
- When your job is in the queue, you aren’t left staring at a blank white screen
- More robust job builder with support for custom Slurm arguments
- Debug mode

**Rstudio**
This should be similar to JupyterHub Rstudio, but with a few improvements:
- Rstudio is containerized, and the new image include tidyverse, devtools, tex, geospatial, and cuda
- Previous R modules (r-st, r-terra, etc.) are now obsolete, they are pre-installed
- More robust job builder with support for custom Slurm arguments
- Debug mode
Bioinformatics tools available on Unity
List of bioinformatics tools coming soon on the Unity docs!
Search and load modules available on Unity

Using the command line interface:

- See list of modules available on Unity: `module avail`
- Searching for modules on Unity: `module spider <package name>`
- Load module: `module load <package name>`
- See list of modules loaded: `module list`

Search for modules using Unity OnDemand: [Unity module explorer](https://example.com/unity-module-explorer)
Unity Module Explorer

last updated Mon Feb 26 2024 15:01:13 GMT+0000 (Coordinated Universal Time)

For information on using modules, read the documentation on Environment Modules.
Click the elements to expand/collapse each category. All nodes in the general partitions are
x86_64 architecture and use that category of module. If you're having trouble navigating the
module hierarchy, ask the Slack helpdesk.

☐ expand/collapse all
☐ show hidden modules

Search for a module... [Go]

Modules

• x86_64 : ...
• aarch64 : ...
• ppc64le : ...
Workflow management systems

- Snakemake
  
  module load snakemake/7.22.0+py3.11.0

- Nextflow
  
  module load nextflow/23.04.1
  
  - nf-core provides bioinformatics pipelines to use with Nextflow
Installing softwares on Unity
Installing softwares via Conda

- Package manager - allows you to install softwares without admin privileges
- Create conda environments and install all of the software that you need for a particular project/analysis
- Conda manages packages via conda channels - bioconda is a channel which contains 1000s of bioinformatics applications (see packages in bioconda)
Installing softwares via

Anaconda and miniconda are available as software modules on Unity

- module load anaconda/2022.10
- module load miniconda/22.11.1-1

- Can install python into a conda environment when you create it or after
- Helpful cheat sheet for conda commands
- Check your software’s documentation for how to install via conda, or Google to find its page on anaconda.org
Installing softwares via CONDA

1. Start an interactive session: `salloc -p cpu -c 1 --mem=5G --time=01:00:00`

2. Load miniconda module: `module load miniconda/22.11.1-1`

3. Create conda environment in your PI’s work directory:
   - `mkdir -p /work/pi_name/$USER-conda/envs`
   - `conda create --prefix=/work/pi_name/$USER-conda/envs/chopper-env python=3.8`

$USER → environment variable specifying your username
Installing softwares via Conda

Why install conda environments in your PI’s work directory?

● Your home directory has a storage capacity of 50GB
● The conda environments are available to every member of the lab
4. Activate conda environment: `conda activate /work/pi_name/$USER-conda/envs/chopper-env`

5. Install software: `conda install bioconda::chopper`

or create conda environment and install packages via a yaml file:

```
conda env create --file meta.yaml --prefix=/work/pi_name/$USER-conda/envs/chopper-env
```
Installing softwares via

CONDA

6. Get usage message

chspper -h
Filtering and trimming of fastq files. Reads on stdin and writes to stdout.

Usage: chopper [OPTIONS]

Options:
  -q, --quality <MINQUAL>    Sets a minimum Phred average quality score [default: 0]
  --maxqual <MAXQUAL>       Sets a maximum Phred average quality score [default: 1000]
  -l, --minlength <MINLENGTH> Sets a minimum read length [default: 1]
  --maxlength <MAXLENGTH>   Sets a maximum read length [default: 2147483647]
  --headcrop <HEADCROP>     Trim N nucleotides from the start of a read [default: 0]
  --tailcrop <TAILCROP>     Trim N nucleotides from the end of a read [default: 0]
  -t, --threads <THREADS>   Use N parallel threads [default: 4]
  -c, --contam <CONTAM>     Filter contaminants against a fasta
  --inverse                 Output the opposite of the normal results
  -h, --help                Print help
  -V, --version             Print version
Installing softwares via

- Apptainer (formerly Singularity) is an open source container platform created to run applications on HPC systems.
- Container: lightweight executable built by packaging the software code together with the related configuration files, libraries, and dependencies.
- Check your software’s documentation if it’s possible to install it via a container.
Installing softwares via

- Do a search in the following container image registries:
  - Docker Hub
  - Quay Container Registry
Installing softwares via Apptainer

Build a container from a given Uniform Resource Identifier (URI):

1. Start an interactive session: `salloc -p cpu -c 1 --mem=5G --time=01:00:00`
2. Load Apptainer module: `module load apptainer/latest`
3. Define location of /.apptainer directory used to store cache files:

   `export APPTAINER_CACHEDIR=<your PI's work directory>/apptainer`
4. Pull the Docker image:

```
apptainer pull [pull options...] [output file] docker://<URI>:tag
```

Example: `apptainer pull chopper.sif docker://quay.io/biocontainers/chopper:0.7.0--hdcf5f25_0`
Installing softwares via

5. Run a command within a container:

```
apptainer exec [exec options...] <container> <command>
```

Example:
```
apptainer exec chopper.sif chopper -h
```
Installing softwares via

Start an interactive shell within a container:

```
apptainer shell [shell options...] <container>
```

Example: `apptainer shell chopper.sif`
Installing softwares via Apptainer

Build a container from a definition file (= recipe to build a container image)

Initialize image with a base image from Docker Hub and specify which image in the next line

Define environment variables that will be set at runtime

Install necessary libraries and Centrifuge

Bootstrap: docker
From: ubuntu:xenial
%environment
export PATH=/opt/bin:$PATH
%post
# base os
apt-get -y update
apt-get -y install build-essential git

git clone https://github.com/DaehwanKimLab/centrifuge
cd centrifuge
make
make install prefix=/opt/

More info on Apptainer documentation
Installing softwares via Apptainer

Build a container from a definition file:

1. In an interactive session and after loading the Apptainer module, run:
   
   ```bash
   unset APPTAINER_BINDPATH
   ```

2. Build container: 
   ```bash
   apptainer build centrifuge.sif centrifuge.def
   ```

3. Run command within container: 
   ```bash
   apptainer exec centrifuge.sif centrifuge -h
   ```
Installing softwares via Apptainer

Sharing files between the host (Unity) and the container:

- By default, Apptainer binds your home directory ($HOME) and the current directory ($PWD) into the container.
- Use --bind (-B) to specify additional bind mounts/paths. Syntax for using the bind option is “source:destination” (source = destination if not specified). Source has to be an absolute path.

Example: bind your PI’s work directory from the host as /data inside the container:

```
apptainer exec --bind <your PI's work directory>:/data centrifuge.sif centrifuge -h
```
Installing softwares via

1. Start an interactive session: 
   \texttt{salloc -p cpu -c 1 --mem=5G --time=01:00:00}

2. Load R container module:
   \texttt{module load r-rocker-ml-verse/4.2.3+apptainer}

3. Start an R interactive session

4. Get list of R packages already installed:
   \texttt{installed.packages()}

5. Check if a package is already installed:
   \texttt{installed.packages("<package name>")}

6. Install a package:
   \texttt{install.packages("<package name>")}

Packages will automatically be installed in $HOME/R/x86_64-pc-linux-gnu-library/4.2

Package name has to be surrounded by quotes!
Installing softwares via

Share R packages amongst members in the same lab

- Create folder to store packages in PI’s work directory:
  
  ```bash
  mkdir -p <path to PI's work directory>/R/4.2
  ```

- Install packages using:

  ```r
  install.packages("<package name>", lib='<path to PI’s work directory>/R/4.2')
  ```

- Add .libPaths("<path to PI’s work directory>/R/4.2") to ~/.Rprofile file
Installing softwares via

- **Bioconductor** is an open software development for computational biology and bioinformatics
- It provides statistical and graphical methods for the analysis of genomic data
- Over 2000 packages

Also available as a container, see on [Docker Hub](https://hub.docker.com)
Biological datasets available on Unity
- Biological databases are located in /datasets/bio
- List of databases:
  - NCBI databases (ncbi nr, ncbi nt, RefSeq Prokaryotic and Eukaryotic db)
  - Dfam → repetitive DNA families, sequence models, and genome annotations
  - EggNOG → biological information hosted by the EMBL
  - Protein databases: BFD, M Gnify, pdb70 pdb_mmcif, pdb_seqres, uniclust30, UniProt, uniref90
  - TARA Oceans

List of biological datasets coming soon on the Unity docs!
Submitting jobs on Unity
Submitting a batch script

● Submit a batch job to the job scheduler (slurm)
● The scheduler manages resources on Unity and executes the job when the resources become available
● The batch job is a task which can be executed without user intervention
Components of a batch script

**Create file**

```
#!/bin/bash
#SBATCH -p cpu-preempt
#SBATCH --mem=30G
#SBATCH --time=03:00:00
module load miniconda/22.11.1-1
conda activate /work/pi_name/$USER-conda/envs/chopper-env
gunzip -c reads.fastq.gz | chopper -q 10 -l 500 | gzip > filtered_reads.fastq.gz
```
How to check on job status

- `squeue --me`: Check on the jobs that you are currently running
- `scancel <job_id>`: Cancel a specific job
- `sinfo`: Check on the status of the cluster
- `sacct`: Look at your entire slurm history
- `scontrol show job <job_id>`: Show information for queued or running job
- `scontrol update <job_id> partition=cpu,cpu-long`: If job is in queue for a long time, can update partition
Running an interactive job

- Allows you to access CLI on a compute node
- 8 hour time limit

```bash
salloc -p cpu-preempt --mem=30G --time=03:00:00
```

- Load the necessary software
```bash
module load anaconda/2022.10
conda activate /work/pi_name/$USER-conda/envs/chopper-env
```

- Run your commands
```bash
gunzip -c reads.fastq.gz | chopper -q 10 -l 500 | gzip > filtered_reads.fastq.gz
```
Running interactive applications on Unity Ondemand

- This is a great option if you like to run your code in e.g. RStudio or in a Jupyter notebook and don’t need more than 8 hours
- Available interactive apps: Jupyter, MATLAB, Mathematica, RStudio, rclone, VScode, and Unity Desktop
  - Requests for new graphical apps are accepted, or we will tell you if it’s available via Unity Desktop
  - [Tutorial for Jupyter OnDemand](#)
## RStudio

All fields can be left blank to use their default settings.

**The resources you allocate must fit inside one compute node. See the hardware specifications of our nodes on the node list.**

**R Version**

4.2.3

**Partition**

cpu,cpu-preempt

**Maximum job duration**

1:00:00

**CPU core count**

2

**Memory (in GB)**

8

**GPU count**

0

**Host**

cpu033.unity.rc.umass.edu

**Created at**

2024-02-25 20:46:27 UTC

**Time Remaining**

58 minutes

**Session ID**

f8ed7026-9cb7-4d67-9e7c-d8a61eec209a

**Launch**

*The RStudio session data for this session can be accessed under the data root directory.*
if (requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install("dada2", version = "3.16")
library(dada2)
setwd("/work/pi_rbeinart_url_edu/Anna")

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.

[Workspace loaded from ~/.RData]

Connected to your session in progress, last started 2024-Feb-25 20:45:50 UTC (3 minutes ago)
If there is a GUI that you would like to run on Unity but which isn’t available as an interactive app, you may be able to run it via Unity Desktop.
Additional Resources

- Unity Onboarding video (Spring 2024)
- Snakemake workshop
- Conda docs on Unity
- AI lab workshops available this semester
- Unity community Slack
- More contact information

Next workshop

- More details about job submission, resource allocation, and how to ensure your code is running efficiently on Unity
- Monday, March 25 @ 11 am - Zoom link here