

# Using R at CHPC

Brett Milash and Anita Orendt

Center for High Performance Computing

03/28/2023

# Methods of using R at CHPC

Command	Attributes
R	Interactive, command line style
RStudio	Interactive, either web browser or X-Windows GUI
Jupyter Lab or Notebook	Interactive, document based, web browser GUI
Rscript	Non-interactive, batch script oriented
R Markdown	Non-interactive, document based

# Methods to access resources at CHPC

Method	Attributes	Resource
ssh to interactive node	command line or GUI	interactive node
Fastx to interactive node	command line or GUI, persistence	interactive node
OnDemand cluster shell access	convenient, like ssh, command line only	interactive node
SLURM sbatch command	non-interactive (batch mode)	compute node(s)
SLURM salloc command	interactive command-line or GUI	compute node(s)
OnDemand system installed applications	web-based access	compute node(s)

Remember the appropriate uses for interactive and compute nodes:

- Interactive nodes: writing code, installing code, small-scale testing, debugging, managing SLURM jobs
- Compute nodes: heavy-duty computing (simulations, stats, data visualization) whether interactive or not

# R use methods vs. CHPC access methods

	R	RStudio	Jupyter	RScript	R Markdown
ssh (to interactive node)					
Fastx (to interactive node)					
OnDemand cluster shell access (runs on interactive node)					
SLURM sbatch (compute node)					
SLURM salloc (compute node)					
OnDemand system installed applications (runs on compute nodes)					

# ssh, Fastx, and OnDemand shell access demo

- These methods provide access to interactive nodes
- Graphics (whether GUI or graphical output) requires X-forwarding
  - On Mac use “ssh -Y *username@hostname*”
  - On Windows use Xming (<https://xming.en.softonic.com/> )
- X-forwarding can be slow without some help

# R use methods vs. CHPC access methods

	R	RStudio	Jupyter	RScript	R Markdown
ssh (to interactive node)	✓ good for testing	✓ but slow	Inefficient - not recommended	✓ good for testing	✓ good for testing
Fastx (to interactive node)	✓ good for testing	✓	Inefficient - not recommended	✓ good for testing	✓ good for testing
OnDemand cluster shell access (on interactive node)	✓ but no graphics	✗ - requires X windows	✗ - requires X windows	✓ good for testing	✓ good for testing
SLURM sbatch (compute node)					
SLURM salloc (compute node)					
OnDemand system installed applications (on compute nodes)					

“✓ good for testing” means software works well within computing limits of interactive node

# SLURM sbatch and salloc demo

- Both methods provide access to compute nodes
- sbatch is batch oriented - therefore non-interactive
- salloc starts an interactive shell session on a compute node

# R use methods vs. CHPC access methods

	R	RStudio	Jupyter	RScript	R Markdown
ssh (to interactive node)	✓ good for testing	✓ but slow	Inefficient - not recommended	✓ good for testing	✓ good for testing
Fastx (to interactive node)	✓ good for testing	✓	Inefficient - not recommended	✓ good for testing	✓ good for testing
OnDemand cluster shell access (on interactive node)	✓ but no graphics	✗ - requires X windows	✗ - requires X windows	✓ good for testing	✓ good for testing
SLURM sbatch (compute node)	✗	✗	✗	★★★★★	★★★
SLURM salloc (compute node)	✓	✓	Inefficient - not recommended	✓	✓
OnDemand system installed applications (on compute nodes)					

“✓ good for testing” means software works well within computing limits of interactive node



# OnDemand demo

- Web portal
- Access to compute nodes
- Very good for web and GUI applications

# R use methods vs. CHPC access methods

	R	RStudio	Jupyter	RScript	R Markdown
ssh (to interactive node)	✓ good for testing	✓ but slow	Inefficient - not recommended	✓ good for testing	✓ good for testing
Fastx (to interactive node)	✓ good for testing	✓	Inefficient - not recommended	✓ good for testing	✓ good for testing
OnDemand cluster shell access (on interactive node)	✓ but no graphics	✗ - requires X windows	✗ - requires X windows	✓ good for testing	✓ good for testing
SLURM sbatch (compute node)	✗	✗	✗	★★★★★	★★★
SLURM salloc (compute node)	✓	✓	Inefficient - not recommended	✓	✓
OnDemand system installed applications (on compute nodes)	✗	★★★★★	★★★★★	✗	✗

“✓ good for testing” means software works well within computing limits of interactive node

# R code for interactive vs batch jobs

- Some tasks are inherently interactive
  - Coding
  - Debugging
  - Data visualization
- Some tasks are inherently batch oriented
  - Large or long-running simulations
  - Processing lots of data files
- To write R code that adapts to both use cases:
  - Write lots of functions
  - Use `interactive()` to test whether job is interactive or batch

# Running parallel R code in a SLURM job

- A SLURM job may not have access to all the cores on a node
- To count all the cores on the machine (which is not what we want):

```
# How many cores are on this machine?  
> library(parallel)  
> detectCores()  
[1] 64
```

- To count the cores available to your job:

```
# How many cores are available to me on this node?  
> strtoi(Sys.getenv("SLURM_TASKS_PER_NODE"))  
[1] 10  
# How many cores are available to my potentially multi-node job:  
> strtoi(Sys.getenv("SLURM_NTASKS"))  
[1] 10
```