

Notes for using OSCAR – 1

- <https://docs.ccv.brown.edu/oscar/getting-started>;
<https://docs.ccv.brown.edu/bootcamp/bootcamp-2025/schedule/day1>
- Checkout Open Ondemand, running slurm jobs
- Bash/Slurm: CLI, GUI, VSCode; LMOD, node/socket, gateway/compute node
 - checkquota, myaccount, myq (squeue -u xhuan171 = squeue --me; myjobinfo -j <jobID>), man <command> (<command> -h), groups, id, whoami, hostname; scontrol show node <node-name>, sacctmgr list assoc user=<username> -p, sacctmgr list qos -p, scontrol show partition; env
 - Node info: sinfo, sinfo -s, nodes, sfeature (OSCAR only)
 - find . -atime +25; stat <filename/dirname>
 - tar -czf X.tar.gz *, tar -xzf X.tar.gz; which <module_name>
 - interact -q batch -n 2 -m 16G -t 02:00:00; interact -h hostname; interact -q gpu -g 1 -t 1:00:00
 - ncdu; squeue -u <username> -t PENDING -start; sstat -j <jobid>, scontrol show job <jobid>, watch queue -u \$USER, sacct -S 2012-01-01 (-E XXX-XX-XX), sacct -lj <jobid>
 - To monitor: jobstats <jobid>, **seff** <jobid>; ssh <nodeid>, top
 - Chaining: ; && ||, if, loops, define a function
 - Other features: Job arrays(--array=i-j, SLURM_ARRAY_TASK_ID), dependent jobs (--dependency=afterany:\$jobID)

Notes for using OSCAR – 2

- File transfer: CIFS mount (GUI,samba), scp, rsync, sftp, GUI, OOD, Globus for large files
 - `rsync -prtv --progress --out-format="%t %f %b" /source/dir/ /destination/dir/` (maybe use `-a` archive and `-z` compress, too?)
 - (Windows) Robocopy `C:\some_directory D:\some_directory /DCOPY:T /COPYALL /E /R:0`
- Matlab, command line & OOD
 - `whos`; Best practice: pre-allocate memory, sequential access, vectorization,
 - `matlab -nojvm` (for parallel computing) `-nodisplay -nosplash, -r "run('test.m')"`
 - On OSCAR, matlab v.s. matlab-threaded
 - Parallelization: `parpool(2); spmd; COMMAND; end; delete(gcf('nocreate'))`
- To use GPU
 - To check: `sfeature, sinfo -s, myaccount`
 - To run: `interact -q gpu -g 1 -n 1 -m 32g -t 1:00:00; #SBATCH -p gpu -gres=gpu:1`
 - NGC container (apptainer) ??
 - Python venv, or Jupyter notebook
 - To monitor: `jobstats, (ssh XXX; top), watch -n 1 nvidia-smi; PyTorch/tensorflow profiler`
 - Estimate model size, `params < VRAM` (GPU memory)
- To preserve files in scratch space, touch access time, i.e.,
`'find . -exec touch -a {} +'`