

# Oscar

Oscar is Brown's high-performance computing cluster managed by the Brown Center for Computation and Visualization (CCV). Long-term visitors to ICERM with a Brown sponsored ID are provided with an exploratory account on Oscar upon arrival. Oscar access may also be requested by short-term visitors on a request-basis with advance notice.

- [Overview of Oscar](#)
- [Open OnDemand Access to Oscar](#)
- [SSH Access to Oscar](#)
- [Oscar Software Modules](#)
  - [Currently Available Modules](#)
  - [Oscar: Sage](#)
  - [Oscar: MATLAB](#)
  - [Oscar: Mathematica](#)

# Overview of Oscar

Long-term visitors to ICERM who receive a Brown ID are provided with an exploratory account on the Oscar high performance computing cluster maintained by Brown's [Center for Computation and Visualization](#). Oscar access may also be provided to short-term visitors on request-basis with *advance notice*.

*If you are coming for a short-term visit and would like to request Oscar access, please contact ICERM IT staff no later than three weeks before your planned visit to ensure enough time to process a sponsored ID and enable Oscar access.*

Oscar can be used via SSH terminal or via the Oscar VNC client. The Oscar VNC client is preinstalled and available on the thin clients in all ICERM 10th floor offices and the graduate student workspace in the 10th floor lobby.

## Using Oscar

Oscar provides both [command line \(SSH\)](#) access and [virtual desktop \(VNC\)](#) access to the computing cluster. All Oscar accounts are capable of using both interfaces, so you should choose the method that will work best for what you are trying to accomplish.

Oscar accounts are tied to Brown Shibboleth accounts, so you must have an ICERM Sponsored Brown ID to use Oscar. Once Oscar access is enabled for your Brown account, you can log in to SSH and VNC using your Brown Shibboleth username and password.

## SSH Usage

Oscar's SSH interface provides a standard command line shell for interacting with the cluster. This is the best option for using simple command line scripts and submitting scripts for batch job processing. See the [SSH Login Instructions](#) for information on how to connect and basic usage. It's important to note that you should never run complex scripts or computations within the SSH login nodes. Computations should either be submitted as batch jobs or run in an interactive compute session using the `interact` command.

## Open OnDemand

Oscar's OOD is a web portal to the Oscar computing cluster. This is the easiest way to access Oscar and navigate around. Once logged in you effortlessly be able to access active jobs, use MATLAB or Mathematica, and even use the Oscar Shell CLI. See the [OOD Login Instructions](#) for information on how to get started with the web portal.

## VNC Usage

Oscar's VNC client provides a standard Linux graphical desktop using virtual network computing. If you'd like to use applications with a GUI, such as MATLAB or Mathematica, this is the best way to do so. Oscar's VNC sessions run within the scheduling system directly on a compute node, similar to the `interact` command, so you can run computations directly in the VNC session. See the [VNC Login Instructions](#) for information on how to get started with the VNC client.

## Available Software

Oscar maintains a [large library of software packages](#) for use on the HPC cluster. Some of the most commonly used mathematical applications and languages available include:

- MATLAB
- Mathematica
- Sage
- Julia
- R
- Python
- Maple
- Magma

To see the most up to date list of available software, log in to your Oscar account and run the terminal command `module avail`.

## Loading Software Modules

Oscar has a large library of software available on the cluster, but only a few apps are pre-loaded in to your sessions. The commands below will allow you to list all available modules, search the list of modules, and load/unload software packages. When using Oscar over VNC, you must open the Terminal Emulator to run these commands.

- To view all available software packages, type `module avail`.

- To search the list of available packages, type `module avail <package>`.
- For example, to search for all available versions of mathematica: `module avail mathematica`. Many packages, like Mathematica, have multiple versions available. This command lets you see all available versions of the package you searched for.
- To load a package into your session, type `module load <package/version>`.
- For example, `module load mathematica/11.0`. This will load the Mathematica 11.0 into your session and make it available for use.
- To unload a package you are no longer using, type `module unload <packagename>`

## Module-specific Guides

- [Oscar: Mathematica](#)
- [Oscar: MATLAB](#)
- [Oscar: Sage](#)

## Running Jobs

Oscar supports two main methods of running jobs: **batch jobs** and **interactive sessions**.

- Batch jobs are pre-scripted and can be submitted to the cluster's scheduler via the `sbatch` command.
- Interactive jobs are command-line sessions run directly on a compute node via the `interact` command that can be used in real time.

The [Running Jobs](#) page of the [Oscar User Manual](#) provides the most detailed and up-to-date instructions on scripting and submitting jobs.

## Important Notes About the Oscar Cluster

- **Please do not run any computations or simulations on the login nodes, as they are shared with other users. Use SLURM to submit a batch job to the queue for computations or start an interactive session on one of the compute nodes with the command `interact`.**

- The full [Oscar User Manual](#) is available on the CCV website.
- Oscar uses SLURM for managing batch jobs and interactive sessions on the cluster. Detailed instructions on submitting jobs is available on the [Running Jobs](#) page of the [Oscar User Manual](#).
- CCV has a [large library of software](#) already installed on the cluster. For a full list of available software, run the command `module avail`. More information about Oscar's software packages are available in the [Oscar User Manual](#). If you require a software package that is not currently available on the Oscar cluster, please contact ICERM's IT staff and we will work with CCV to get the software installed.
- Users can install sub-packages for some modules (like Sage and Python) to their home folders on their own by using the `-user` as an option in the install command. For example, with Sage, the command would be something like `sage -i <package name> -user`.
- The [Oscar VNC Client](#) is a Java application that can be installed on most laptops with Java installed. If you wish to install the client on your own laptop, please make sure to download the latest version of CCV VNC Client 2.0+ to ensure proper support and compatibility.

*If you have questions about these instructions or require further assistance, please contact the ICERM IT staff by dropping by the administrative offices or emailing [support@icerm.brown.edu](mailto:support@icerm.brown.edu).*

# Open OnDemand Access to Oscar

Open OnDemand (OOD) is a web portal to the Oscar computing cluster. **An Oscar account is required to access Open OnDemand.** Visit this [link](#) in a web browser and sign in with your Brown username and password to access this portal.

Benefits of OOD:

- No installation needed. Just use your favorite browser!
- No need to enter your password again. [SSH into Oscar](#) in seconds!
- No need to use two-factor authentication multiple times. Just do it once, when you log into OOD.
- Use it with, or without, VPN. Your workflow remains the same.
- Just a good time over all ☺☺

Once logged in you'll see a landing page that looks similar to the photo below:

**OPEN**

# OnDemand

OnDemand provides an integrated, single access point for all of your HPC resources.

**Pinned Apps** A featured subset of [all available apps](#)



Active Jobs

System Installed App



Home Directory

System Installed App



Job Composer

System Installed App



OSCAR Shell Access

System Installed App

## Default GUIs



Basic Jupyter Notebook with  
Anaconda

System Installed App



Desktop

System Installed App



Jupyter Notebook for Python  
Environments

System Installed App



MATLAB on OSCAR

System Installed App



Guides to help you figure your way out once logged in:

[Using File Explorer on OOD](#)

[Web-based Terminal App](#)

[Interactive Apps on OOD](#)

[Using Python or Conda environments in the Jupyter App](#)

# SSH Access to Oscar

1. Open a terminal prompt. On the ICERM thin clients, enter the username `user` and the password `icerm` to log in to the terminal.
2. In the terminal, type `ssh <your ccv username>@ssh.ccv.brown.edu`. If you are asked to verify the authenticity of the host 'ssh.ccv.brown.edu', type `yes`.
3. You will now be prompted for your password. Enter your password (nothing will show up when you type in the terminal password prompt) and press enter.
4. Once logged in, you should see a “Welcome to Oscar!” message. This means you're now connected to one of the login nodes, which you can use to manage your files and submit batch jobs.

**Please DO NOT run any computations directly on the login nodes. Use the batch system to submit your computations to the queue to be processed on the computation nodes or start an interactive session on one of the compute nodes with the command `interact`.**

**See CCV's Oscar Documentation page on [Running Jobs](#) for detailed instructions on batch jobs and interactive sessions.**



# Oscar Software Modules

List and instructions for software modules available on the Oscar cluster.

# Currently Available Modules

If you require a software package that is not currently available on the Oscar cluster, please [contact ICERM's IT staff](#) and we will work with CCV to get the software installed.

*This list is current as of November 10, 2021. To see the most up to date list of software modules, log into your Oscar account and run the command `module avail`.*

```
~~~~~ category:
~~~~~
bxh_xcede_tools/1.11.14  confocal/1.0      potfit/20201014
cesm_post_process/Oct18  pigz/2.4
~~~~~ category: Microstructure modeling ~~~~~
oof2/2.1.17
~~~~~ category: access controlled ~~~~~
Molpro/2012.1.15
Molpro/2015_gcc
Molpro/2015_serial
Molpro/2018.2_ga
Molpro/2019.2
Molpro/2019.2_ga
Molpro/2020.1
Molpro/2020.1_ga
Molpro/2020.1_openmpi_4.0.5_gcc_10.2_slurm20
gaussian/g09
gaussian/g09-D01
gaussian/g09-D01-TEST
vasp/5.4.1
vasp/5.4.1_debug
vasp/5.4.1_mvapich2-2.3.5_intel_2020.2_slurm20
vasp/5.4.4
vasp/5.4.4_intel
vasp/5.4.4_mvapich2-2.3.5_intel_2020.2_slurm20
vasp/5.4.4_openmpi_4.0.5_gcc_10.2_slurm20
vasp/5.4.4a
vasp/6.1.1_ompi405_yqi27
```

```
vasp/6.1.1_openmpi_4.0.5_intel_2020.2_yqi27_slurm20
vasp/6.1.1_yqi27
~~~~~ category: graph partitioning ~~~~~
metis/5.1.0  parmetis/4.0.3  scotch/6.0.4
~~~~~ category: image processing
~~~~~
multinest/3.10  openslide/3.4.1  pstokes/1.0
~~~~~ category: libraries
~~~~~
astropy/3.2.1  cudnn/7.4  flann/1.8.4  lapack/3.7.0
astroquery/3.0.9  cudnn/7.6  hnn/1.0  mumps/5.0.2
blas/3.7.0  cudnn/7.6.5  horovod/0.16  mumps/5.0.2-seq
cudnn/5.1  cudnn/8.1.0  horovod/0.19.5  pcl/1.9.1
cudnn/6.0  cudnn/8.2.0  lapack/3.4.2  pcl/1.9.1_nurbs
cudnn/7.0  dlib/19.17  lapack/3.6.0  scalapack/2.0.2
~~~~~ category: library
~~~~~
openslide-python/1.1.1
~~~~~ category: machine learning
~~~~~
keras/2.0.9  keras/2.1.3_py3  nccl/2.8.4
keras/2.1.1  nccl/2.4.7
~~~~~ category: math
~~~~~
gsl/1.15  gsl/2.3  gsl/2.5
~~~~~ category: package
~~~~~
subread/1.6.2
~~~~~ category: software
~~~~~
mriconvert/2.1.0
~~~~~ category: utility
~~~~~
mriconvert/2.1.0
~~~~~ category: visualization
~~~~~
mayavi/4.6.0
~~~~~ category: (none)
~~~~~
HiC-Pro/2.7.8  jo/1.4  repeatmasker/4.1.0
```

HiC-Pro~/2.10.0      libsodium/1.0.17      repeatmodeler/1.0.11  
awscli/1.0      mumax/3.9.3      sdl2/2.0.12  
basilisk/1.0      ncl/6.4.0      siesta/3.2  
canu/1.7.1      neuron/7.5      siesta/4.1  
dropest/0.86      neuron/7.7      skewer/0.2.1  
fiji/2017-java6      neuron/7.7\_mpi      squashfs/4.3  
fiji/2020      nvidia-driver/440.33.01      tmux/2.8  
fmri/20.0.0      pdflib/7.0.5      topcat/4.7  
fv/5.5      pypy/6.0.0\_2.7      transrate/1.0.3  
gcm-core/2.0.498      pypy/6.0.0\_3.5      vscode/1.22  
grads/2.2.0      pypy/7.3.0\_3.6      zstd/1.5.0  
hotnet/1.0      repeatmasker/4.0.7

~~~~~ category: Applied Maths

~~~~~  
oof2/2.1.17

~~~~~ category: Astrophysics

~~~~~  
polychord/1.0 polychord/2.0

~~~~~ category: Bioinformatics

~~~~~  
MultiQC/1.0 muscle/3.8.31

~~~~~ category: CAD

~~~~~  
freecad/0.18 librecad/2.0

~~~~~ category: Chemistry

~~~~~  
dalton/2018.0

dalton/2018.0\_mvapich2-2.3.5\_intel\_2020.2\_slurm20

~~~~~ category: Computer Architecture ~~~~~

risc/1.0

~~~~~ category: EDA tools

~~~~~  
cadence/IC06.18.090 cadence/IC6.1.8.33 incisive/15.20.085

~~~~~ category: Executable for ceckpointings ~~~~~

dmtcp/2.6.0

~~~~~ category: Flag for CMAKE for fast build ~~~~~

ninja/1.9.0

~~~~~ category: GENETEICS

~~~~~  
eigensoft/6.0

```
~~~~~ category: Genomic
~~~~~
Genrich/0.5 mcscanx/1.0
~~~~~ category: Genomic Library
~~~~~
redundans/1.0
~~~~~ category: Genomics
~~~~~
baypass/2.2    beast/2.5.2    kissplice/2.5.1
beast/1.10.4    kissplice/2.5.0
~~~~~ category: Genomics Sequencing Alignment ~~~~~
qualimap/2.2.1
~~~~~ category: Graphics
~~~~~
beagle/1.0.0
~~~~~ category: Helper function type ~~~~~
datamash/1.3
~~~~~ category: I/O
~~~~~
cdo/1.9.8
cdo/1.9.9
hdf5/1.10.0
hdf5/1.10.1_mvapich2-2.3.5_gcc_10.2_slurm20
hdf5/1.10.1_parallel
hdf5/1.10.5
hdf5/1.10.5_fortran
hdf5/1.10.5_mvapich2-2.3.5_intel_2020.2_slurm20
hdf5/1.10.5_openmpi_3.1.3_gcc
hdf5/1.10.5_openmpi_3.1.6_gcc
hdf5/1.10.5_openmpi_4.0.0_gcc
hdf5/1.10.5_openmpi_4.0.5_gcc_10.2_slurm20
hdf5/1.10.5_parallel
hdf5/1.10.7_hpcx_2.7.0_intel_2020.2_slurm20
hdf5/1.10.7_openmpi_4.0.5_gcc_10.2_slurm20
hdf5/1.10.7_openmpi_4.0.5_intel_2020.2_slurm20
hdf5/1.12.0_hpcx_2.7.0_intel_2020.2
hdf5/1.12.0_hpcx_2.7.0_intel_2020.2_slurm20
hdf5/1.12.0_openmpi_4.0.5_intel_2020.2_slurm20
jasper/1.900.1-intel
nco/4.6.6
```

nco/4.8.2  
nco/4.9.3  
netcdf/3.6.3  
netcdf/4.4.1.1\_gcc  
netcdf/4.4.1.1\_intel  
netcdf/4.4.1.1\_pgi  
netcdf/4.7.0\_intel2019.3  
netcdf/4.7.4\_gcc8.3  
netcdf/4.7.4\_gcc\_10.2\_hdf5\_1.10.5  
netcdf/4.7.4\_intel\_2020.2\_hdf5\_1.12.0  
sqlite/3.25.2  
sqlite/3.31.1  
udunits/1.12.11  
udunits/2.2.24  
~~~~~ category: IDE  
~~~~~  
meshlab/20190129\_qt59 spyder/3.3.5  
~~~~~ category: Image processing and visualization ~~~~~  
opencv-python/4.1.0.25  
~~~~~ category: Library for optimizer ~~~~~  
numbbo/2.3\_python2 numbbo/2.3\_python3  
~ category: Material science: <https://icet.materialsmodeling.org/overview.html> ~  
icet/0.3  
~~~~~ category: Medical imaging  
~~~~~  
mricrogl/1.0 mricrogl/1.2.20210317  
~~~~~ category: Multifunctional Wavefunction Analyzer ~~~~~  
multiwfn/3.3.9  
~~~~~ category: Numerical Library  
~~~~~  
ngsolve/6.2.1901  
~~~~~ category: Numerical Solver  
~~~~~  
dedalus/2.1810  
dedalus/2.1905  
dedalus/2.1905\_openmpi\_4.05\_gcc\_10.2\_slurm20  
dedalus/Sep2019  
nektar++/4.5.0  
~~~~~ category: Quantum Monte Carlo  
~~~~~

```

casino/2.13
~~~~~ category: Tool for Image Manipulation ~~~~~

netpbm/10.47.71
~~~~~ category: Utility
~~~~~

globus/1.11
~~~~~ category: Visualization Package Qt's Python API ~~~~~

pyqt/4.12.1
~~~~~ category: X
~~~~~

turbovnc/2.1.1 turbovnc/2.2.1 turbovnc/2.2.6 turbovnc/2.2.b1
~~~~~ category: astronomy
~~~~~

ares/0.1          iraf/2.16          sextractor/2.8.6
ares/0.5          iraf/2.16.1+2018.11.01 supermongo/2.4.34
ccfits/2.5        lsst/14.0          swarp/2.38
cfitsio/3.450     planck_likelihood/3.0 theli/1.9.5
cfitsio/3.48      planck_likelihood/3.01 wcstools/3.9.5
ds9/7.6          scamp/2.0.4
galfit/3.0.5      sextractor/2.19.5
~~~~~ category: astrophysics
~~~~~

fiat/Oct2019
~~~~~ category: astropy computing
~~~~~

astropy/3.2.1
~~~~~ category: astroquery computing
~~~~~

astroquery/3.0.9
~~~~~ category: benchmark
~~~~~

elbencho/2.0-1
~~~~~ category: binary for genome studies ~~~~~

bedGraphToBigWig/1.04
~~~~~ category: bio
~~~~~

SNAP/2013-11-29          mira/4.0.2
abyss/2.1.1             mne/0.17.0
ale/20140120            moseq2/0.1.2
aliview/1.25            moseq2/0.2.0

```

amplisat/20170208	mothur/1.39.5
angsd/0.920	mrBayes/3.2.6
anvio/5.2	mriqc/0.15.2
anvio/5.5	msmc/1.0.0
anvio/6.1	msmc/1.1.0
anvio/7	msmc/2.1.2
arb/6.0.6	msprime/0.7.0
augustus/3.3	msprime/0.7.2
bamcleave/Oct2018	msprime/0.7.4
bamtools/2.3.0	mummer/4.0.0.beta2
bamtools/2.4.1	namd/2.11-multicore
bbmap/38.23	namd/2.13b1-multicore
bbtools/38.12	nda-tools/1.0
bcftools/1.10.2	nextclip/1.3.1
bcftools/1.13	nextgenmap/0.5.2
bcftools/1.9	nipype/1.5.0
bcl2fastq/2.20.0	nseg/20181012
bedops/2.4.35	nvtop/1.1.0
bedtools/2.25.0	obitools/1.2.12
bedtools/2.26.0	octopus/0.7.4
beeline/1.0	oligotyping/2.0
biolite/1.0.0	omero/5.6.2
biomed/1.0	opera/2.0.6
biopython/1.66	orthomcl/2.0.9
biopython/1.73	paml/4.8
bismark/0.20.0	pandaseq/2.11
blast/2.2.30+	paris/1.1.3
blast/2.6.0+	paup/4.0a157
blast/2.7.1+	paup/4.0a166
blast/2.8.1+	paup/4.0a168
blast/2.9.0+	pbsuite/15.8.24
blat/36x2	pcangsd/0.98
bowtie/1.2.0	picard-tools/2.17.11
bowtie/2.4.1	picard-tools/2.9.2
bowtie2/2.3.0	pilon/1.22
bowtie2/2.3.5.1	pilon/1.24
bowtie2/2.4.2	plink/1.07
braker/2.1.0	plink/1.90
busco/3.0.2	plink/2.00
bwa/0.7.15	polyphen/2.2.2



cabana/1.0	poolparty/0.8
cabana/1.1_hpcx_2.7.0_gcc_10.2_slurm20	popoolation2/1.201
cantera/5.9.7	prinseq/0.20.4
ccp4/7.0	prodigal/2.6.3
circlator/1.5.5	psmc/0.6.5
conn/16b	psmc/0.6.5a
conn/18a	pymol/2.2b
conn/18b	qctool/2.0
conn/18b_runtime	qiime/1.9.1
conn/18b_standalone	quast/5.0.0
conn/19c	raremetal/4.14.1
conn/20b	raxml/8.2.10
conn/20b_standalone	raxml-ng/0.9.0
connectome_workbench/1.3.1	recon/1.08
connectome_workbench/1.3.2	repeatscout/1.0.5
connectome_workbench/1.4.2	revbayes/1.0.5
cpac/1.0.1	revbayes/1.0.6
cpac/1.5.0	rmblastn/2.10.0
cufflinks/2.2.1	rmblastn/2.6.0
cutadapt/1.14	rsem/1.3.1
dada2/1.4.1	salmon/0.11.3
dada2/1.8	salmon/0.8.2
dbg2olc/Sep2018	salmon/1.0.0
dendropy/4.2.0	salmon/1.2.0
diamond/0.9.12	salmon/1.3.0
diamond/0.9.24	samblaster/0.1.24
diamond/2.0.8	samtools/0.1.18
dida/1.0.1	samtools/1.10
diffreps/1.55.6	samtools/1.12
dlcpar/1.1	samtools/1.13
edge-pro/1.3.1	samtools/1.2
eems/1.0	samtools/1.3.1
emboss/6.6.0	samtools/1.4.1
entap/0.8.1-beta	samtools/1.9
entap/0.8.4-beta	schrodinger/2017-3
exonerate/2.2.0	schrodinger/2020-1
fastme/2.1.5	schrodinger/2020-3
fastqc/0.11.5	schrodinger/2021-1
fastsimcoal/2	schrodinger/2021-3
fasttree/2.1.10	selscan/1.2.0a

fasttree/2.1.11	seqkit/0.10.1
fastuniq/1.1	seqtk/1.3
fastx-toolkit/2.6	sga/0.10.15
fgap/1.8.1	sift/6.2.1
fgbio/0.6.1	sift4g/2.0.0
figtree/1.4.4	sjaracne/0.2.0
flashpca/2.0	skesa/1.68
freebayes/1.1.0	slicer/4.10.0
freebayes/1.2.0	slim/3.1
garlic/1.1.6	slim/3.2.1
gatk/4.0.9.0	slim/3.3
gatk/4.1.6.0	snippy/Oct2017
gatk/4.2.2.0	spades/3.10.1
gdcclient/1.4.0	spades/3.13.0
genemark/4.33	span/0.10.0.4787
guppy/4.0.11	spm/spm12
guppy/4.2.2	spm/spm12_7771
guppy/5.0.7	sprint/0.1.8
hapbin/1.3.0	sratoolkit/2.10.5
hisat2/2.1.0	sratoolkit/2.11.0
hmmer/3.1b2	sratoolkit/2.8.2-1
homer/4.10	stacks/2.2
hotnet2/1.2.1	stampy/1.0.32
htseq/0.11.1	star/2.6.1b
htseq/0.9.1	star/2.7.2a
idba/1.1.3	star/2.7.3a
idemp/20180928	starfusion/1.7.0
ilash/1.0.0	starfusion/1.8.1
ilash/1.0.2	stringtie/1.3.3b
interproscan/5.26-65.0	stringtie/1.3.4d
interproscan/5.27-66.0	tabix/0.2.6
interproscan/5.28-67.0	tablemaker/2.1.1
interproscan/5.32-71.0	taxonkit/0.3.0
interproscan/5.33-72.0	tophat/2.1.1
iq-tree/1.6.7	transabyss/2.0.1
iq-tree/2.0.6	transindel/20181017
jellyfish/1.1.11	trf/4.09
jellyfish/2.2.10	trimal/1.4
kneaddata/0.6.1	trimgalore/0.5.0
kraken/2.0.7-beta	trimmomatic/0.36

kraken/2.0.9-beta	trinity/2.11.0
kraken/2.1.2	trinity/2.4.0
last/956	trinity/2.8.4
macs/2.1.1	usearch/5.2.32
macs/2.2.6	usearch/8.1.1861
mafft/7.310-with-extensions	varscan/2.3.9
maker/2.31.10	vcftools/0.1.16
marsbar/0.42	velvet/1.2.10
maxbin2/2.2.5	velvet/1.2.10a
megahit/1.1.3	vep/101
meraculous/2.2.6	vep/104
metabat/2.12.1	viennarna/2.4.9
metaphlan/2.6.0	vmd/1.9.3
metavelvet/1.2.01	vsearch/2.11.1
migrate/3.6.11	weblogo/2.8.2
~~~~~ category: bio packages	
~~~~~	
bwameth/0.10	
~~~~~ category: bioinformatics	
~~~~~	
umitools/1.0.0	
~~~~~ category: biology	
~~~~~	
haplostrips/1.2.1	htslib/1.9
htslib/1.13	libBigWig/0.4.4
meme/5.0.5	mmsplice/0.2.7
meme/5.3.0	wiggletools/1.2.3
~~~~~ category: build tool	
~~~~~	
scons/3.0.1	
~~~~~ category: cave	
~~~~~	
gigapixelviewer/alpha	minvr/dev_ben
minvr/0.3master	planetaryviewer/alpha
minvr/0.4	scalable/20150828
minvr/beta	scalable/opengl32+
minvr/beta-bleeding	vrg3d/265
~~~~~ category: chemistry	
~~~~~	
Molpro/2012.1.15	
Molpro/2015_gcc	
Molpro/2015_serial	

Molpro/2018.2\_ga  
Molpro/2019.2  
Molpro/2019.2\_ga  
Molpro/2020.1  
Molpro/2020.1\_ga  
Molpro/2020.1\_openmpi\_4.0.5\_gcc\_10.2\_slurm20  
ambertools/amber16  
ambertools/amber16-gpu  
ambertools/amber17  
ambertools/amber17\_lic  
ambertools/amber21  
ase/3.13.0  
ase/3.19.1  
ase/3.8.1  
bagel/1.2.2  
cp2k/7.1  
cp2k/8.1.0  
crossrate/2016.3.23  
dacapo/2.7.16\_mvapich2\_intel  
dice/1.0  
elk/5.2.14  
gaussian/g09  
gaussian/g09-D01  
gaussian/g09-D01-TEST  
gaussview/14Aug20  
gromacs/2016.6  
gromacs/2018.2  
gromacs/2018.2\_gpu  
gromacs/2018.2\_hpcx\_2.7.0\_gcc\_10.2\_slurm20  
gromacs/2020.1  
gromacs/2020.1\_hpcx\_2.7.0\_gcc\_10.2\_slurm20  
gromacs/2020.4\_gpu  
gromacs/2020.4\_gpu\_hpcx\_2.7.0\_gcc\_10.2\_slurm20  
gromacs/2020.4\_hpcx\_2.7.0\_gcc\_10.2\_slurm20  
hande/1.1.1  
hande/1.1.1\_64  
hande/1.1.1\_debug  
hotbit/10mar17  
lammps/11Aug17\_serial  
lammps/29Oct20\_openmpi\_4.0.5\_gcc\_10.2\_slurm20

medea/3.2.3.0  
molden/5.7  
mrcc/2014-intel  
mrcc/2014-intel-threaded  
mrcc/2018-intel  
mrcc/2018-intel-threaded  
mrcc/2020  
n2p2/1.0.0  
n2p2/2.0.0  
n2p2/2.0.0\_hpcx  
nbo/7.0  
nwchem/6.8-openmpi  
nwchem/7.0  
nwchem/7.0.2\_mvapich2-2.3.5\_intel\_2020.2\_slurm20  
openbabel/2.4.1  
openbabel/3.0.0  
openmolcas/18.09.617  
orca/4.0.1.2  
orca/4.1.1  
orca/4.2.1  
orca/5.0.0  
orca/5.0.1  
prophet/augustegm\_1.2  
pyscf/1.4.7  
pyscf/1.6.3  
qchem/5.0.2  
qchem/5.0.2-openmpi  
qmcpack/3.10.0\_openmpi\_4.0.5\_intel\_2020.2\_slurm20  
qmcpack/3.7.0  
qmcpack/3.9.1  
qmcpack/3.9.1\_openmpi\_3.1.6  
qmcpack/3.9.2\_hpcx\_2.7.0\_gcc\_10.2\_slurm20  
qmcpack/3.9.2\_intel\_2020  
qmcpack/3.9.2\_openmpi\_4.0.0\_gcc\_8.3\_slurm20  
qmcpack/3.9.2\_openmpi\_4.0.0\_gcc\_8.3\_slurm20\_complex  
qmcpack/3.9.2\_openmpi\_4.0.1\_gcc  
qmcpack/3.9.2\_openmpi\_4.0.4\_gcc  
qmcpack/3.9.2\_openmpi\_4.0.5\_intel\_2020.2\_slurm20  
quantumespresso/6.1  
quantumespresso/6.3

quantumpresso/6.4  
quantumpresso/6.4.1  
quantumpresso/6.4\_openmpi\_4.0.0\_gcc\_8.3\_slurm20  
quantumpresso/6.4\_openmpi\_4.0.5\_intel\_2020.2\_slurm20  
quantumpresso/6.4\_openmpi\_4.0.5\_intel\_slurm20  
quantumpresso/6.5  
quantumpresso/6.5\_openmpi\_4.0.5\_intel\_slurm20  
quantumpresso/6.6  
quantumpresso/6.6\_openmpi\_4.0.5\_intel\_2020.2\_slurm20  
rotd/2014-11-15\_mvapich2  
schrodinger/2017-3  
schrodinger/2020-1  
schrodinger/2020-3  
schrodinger/2021-1  
schrodinger/2021-3  
sharc/2.0  
sharc/2.1.1  
sharc/2.1.1\_intel  
vasp/5.4.1  
vasp/5.4.1\_debug  
vasp/5.4.1\_mvapich2-2.3.5\_intel\_2020.2\_slurm20  
vasp/5.4.4  
vasp/5.4.4\_intel  
vasp/5.4.4\_mvapich2-2.3.5\_intel\_2020.2\_slurm20  
vasp/5.4.4\_openmpi\_4.0.5\_gcc\_10.2\_slurm20  
vasp/5.4.4a  
vasp/6.1.1\_ompi405\_yqi27  
vasp/6.1.1\_openmpi\_4.0.5\_intel\_2020.2\_yqi27\_slurm20  
vasp/6.1.1\_yqi27  
xcrysden/1.5.60  
~~~~~ category: class  
~~~~~  
class/1435  
~~~~~ category: clients  
~~~~~  
aspera/3.8.1  
~~~~~ category: climate  
~~~~~  
cesm/1.2.1 cesm/2.1.1 esmf/8.0.0 esmf/8.1.0b11  
cesm/1.2.2 esmf/7.1.0r esmf/8.0.0b esmf/8.1.9b17

~~~~~ category: compilers

~~~~~

Xvfb/1.20.12	gcc/10.2	java/8u111
clang/3.9.1	gcc/4.5.4	java/jdk-11.0.11
clang/7.0.0	gcc/4.7.2	java/jdk-12.0.2
clang/7.1.0	gcc/4.9.4	kokkos/3.3.1
cuda/10.0.130	gcc/5.4	kokkos/3.4.1
cuda/10.1.105	gcc/6.2	llvm/11.0.1
cuda/10.2	gcc/6.3	llvm/3.8.1
cuda/11.1.1	gcc/7.2	llvm/4.0.0
cuda/11.1.1_intel_2020	gcc/8.3	llvm/6.0.1
cuda/11.3.1	intel/2011.11.339	llvm/7.1.0
cuda/7.5.18	intel/2013.1.106	nag/6.2
cuda/8.0.61	intel/2017.0	nag/mb16i25dnl
cuda/9.0.176	intel/2018.1	pgi/16.7
cuda/9.1.85.1	intel/2019.3	pgi/2019
cuda/9.2.148	intel/2020.2	swig/3.0.10

~~~~~ category: computer architecture ~~~~~

llvmOpenmp/1.0 llvmOpenmp/2.0

~~~~~ category: containers

~~~~~

singularity2/2.5.2

~~~~~ category: cross webbrowser web application development environment ~~~~~

chromedriver/2.46

~~~~~ category: data analysis(<http://www.mega-nerd.com/>) ~~~~~

libsnd/1.0.28

~~~~~ category: data science

~~~~~

catboost/0.8.1 manureadr/1.0 pandas/py\_3.6.6

catboost/0.8.1\_py3 pandas/py\_3.5.2

~~~~~ category: data structure library ~~~~~

dcm2bids/2.1.4

~~~~~ category: debugging and profiling ~~~~~

forge/18.2.3 forge/19.1.2 forge/21.0.2

~~~~~ category: dev

~~~~~

googletest/1.8.0 qemu/4.1.0

~~~~~ category: devel

~~~~~

bazel/0.23.0 glew/1.13.0 opengl/nvidia-375.66

bazel/0.25.2      glew/2.1.0      opengl/nvidia-390.30  
bazel/0.4.4      opengl/mesa-12.0.6      opengl/nvidia-410.72  
bazel/0.5.4      opengl/mesa-18.3.3      protobuf/3.4.1  
bazel/1.2.1      opengl/nvidia      protobuf/3.6.0\_gcc5.4

~~~~~ category: editor

emacs/26.3    neovim/0.4.4

~~~~~ category: engineering

~~~~~  
abaqus/2017      comsol/5.6  
abaqus/2021      mentor-calibre/2020.1\_36.18  
abaqus/2021.1      spectre/191  
abaqus/2021.1\_intel17      su2/7.0.2  
abaqus/6.12sp2      synopsys/2018.06  
ansys/18.1      synopsys/2020.06  
assura/04.16.107      synopsys/L\_2016.03-SP2

comsol/5.2

~~~~~ category: fMRI

~~~~~  
brainiak/Feb2018

~~~~~ category: fonts

~~~~~  
freetype/2.7.1

~~~~~ category: gene and species tree ~~~~~

phyldog/1.0

~~~~~ category: genetics

~~~~~  
megacc/10.1.8

~~~~~ category: genomic

~~~~~  
admixture/1.3.0      glactools/1.0.7      pybigwig/0.3.15  
deeptools/3.2.1      partitionfinder/2.1.1      pysam/0.15.2  
deeptoolsintervals/0.1.7      py2bit/0.3.0      pyslim/1.0

~~~~~ category: genomics

~~~~~  
cellranger/3.1.0      cellranger/arc-1.0.1      faststructure/1.0  
cellranger/5.0.1      chromopainter/0.0.4      relernn/6Dec2019  
cellranger/6.0.0      clustal\_omega/1.2.4

~~~~~ category: graphics



|                                  |                     |                                |
|----------------------------------|---------------------|--------------------------------|
| campari/3.0                      | gnuplot/5.2.3       | inkscape/Sep-2019              |
| ~~~~~ category: image            |                     |                                |
| ~~~~~                            |                     |                                |
| c3d/1.0.0                        | imagej/1.52a        | leptonica/1.79.0 openexr/2.2.1 |
| ffmpeg/3.2.4                     | imagemagick/7.0.7   | libgd/2.2.5                    |
| ffmpeg/4.0.1                     | isis/3.5.1          | libgit/1.1.0                   |
| ~~~~~ category: image converter  |                     |                                |
| ~~~~~                            |                     |                                |
| mriconvert/2.1.0                 |                     |                                |
| ~~~~~ category: image processing |                     |                                |
| ~~~~~                            |                     |                                |
| libgif/5.1.9                     | libjpeg-turbo/2.0.2 | openjpeg/2.3.1                 |
| libjpeg/9.0                      | mayavi/4.6.0        | openslide-python/1.1.1         |
| ~~~~~ category: languages        |                     |                                |
| ~~~~~                            |                     |                                |
| R/3.3.2                          | julia/1.0           | miniconda/4.10                 |
| R/3.4.0                          | julia/1.0.2         | perl/5.16.0                    |
| R/3.4.3                          | julia/1.1.0         | perl/5.18.2                    |
| R/3.4.3_mkl                      | julia/1.2.0         | perl/5.24.1                    |
| R/3.4.4                          | julia/1.4.1         | perl/5.30.0                    |
| R/3.5.2                          | julia/1.4.2         | perl/5.8.9                     |
| R/3.6.0                          | julia/1.5.0         | python/2.7.12                  |
| R/3.6.3                          | julia/1.5.1         | python/2.7.12_clean            |
| R/4.0.0                          | julia/1.5.2         | python/2.7.16                  |
| R/4.0.3                          | julia/1.5.3         | python/3.5.2                   |
| R/4.0.5                          | julia/1.5.4         | python/3.6.6                   |
| R/4.1.0                          | julia/1.6.0         | python/3.6.6_test              |
| anaconda/2-4.3.0                 | julia/1.6.1         | python/3.6.8_gcc8.3            |
| anaconda/2-5.3.0                 | julia/1.6.2         | python/3.7.4                   |
| anaconda/2020.02                 | julia/1.6.3         | python/3.8.12_gcc8.3           |
| anaconda/3-4.3.0                 | lua/5.3.4           | python/3.9.0                   |
| anaconda/3-5.2.0                 | maple/16            | qt/3.3.8b                      |
| golang/1.15.6                    | maple/20            | qt/3.3.8b-51                   |
| golang/1.16.6                    | materialstudio/2020 | qt/5.10.1                      |
| golang/1.17.1                    | mathematica/10.3.1  | qt/5.12                        |
| idl/8.5.1                        | mathematica/11.0    | qt/5.12.0                      |
| idl_DEEPS/8.7.2                  | mathematica/12.0    | qt/5.13.1                      |
| java/8u111                       | matlab/R2016a       | qt/5.14.2                      |
| java/jdk-11.0.11                 | matlab/R2017a       | qt/5.7.0                       |
| java/jdk-12.0.2                  | matlab/R2017b       | qt/5.9.0                       |

|             |               |              |
|-------------|---------------|--------------|
| julia/0.5.1 | matlab/R2018a | ruby/2.4.0   |
| julia/0.6.1 | matlab/R2018b | rust/1.45.1  |
| julia/0.6.4 | matlab/R2019a | rust/1.50.1  |
| julia/0.7.0 | matlab/R2021a | scala/2.12.2 |

~~~~~ category: lib

~~~~~

lftp/4.8.4

~~~~~ category: libraries

~~~~~

Xvfb/1.20.12

acml/5.2.0-gfortran

agalma/1.0.0

armadillo/9.200.4

assimp/4.1.0

assimp/5.0.0

blast-legacy/2.2.26

boost/1.44.0

boost/1.49.0

boost/1.55

boost/1.57

boost/1.62.0-intel

boost/1.63.0

boost/1.68

boost/1.69

boost/1.75.0\_openmpi\_4.0.5\_intel\_2020.2\_slurm20

boost/1.76.0\_hpcx\_2.7.0\_gcc\_10.2\_slurm20

boost/1.76.0\_hpcx\_2.7.0\_intel\_2020.2\_slurm20

bzip2/1.0.2

c-blosc/1.16.3

cdhit/4.6.8

cuda/10.0.130

cuda/10.1.105

cuda/10.2

cuda/11.1.1

cuda/11.1.1\_intel\_2020

cuda/11.3.1

cuda/7.5.18

cuda/8.0.61

cuda/9.0.176

cuda/9.1.85.1

cuda/9.2.148  
ea-utils/1.04.807  
fabm/1.0.2  
fastq-tools/0.8  
fftw/2.1.5  
fftw/2.1.5-double  
fftw/3.3.6  
fftw/3.3.8  
fftw/3.3.8a  
freeglut/3.0.0  
geos/3.4.2  
geos/3.7.1  
geos/3.8.1  
gerris/1.0  
global\_arrays/5.6.1  
global\_arrays/5.6.1\_i8  
global\_arrays/5.6.1\_openmpi\_2.0.3  
global\_arrays/5.8\_openmpi\_4.0.5\_gcc\_10.2\_slurm20  
gmp/6.1.2  
gsl/1.15  
gsl/2.3  
gsl/2.5  
igraph/0.7.1  
keras/2.0.9  
keras/2.1.1  
keras/2.1.3\_py3  
lemon/1.3.1  
leveldb/1.20  
libcutensor/10.2  
libflint/2.7.1  
libint/2.5.0  
libpng12/1.2.57  
libspatialindex/1.9.3  
libtiff/4.0.10  
libxc/4.3.4  
libzip/0.9-3.1.el6  
lp\_solve/5.5.2.5  
metis/5.1.0  
mpfr/3.1.5  
mpfr/4.0.2

mpi4py/3.0.1\_py3.6.8  
mxnet/1.3.0  
node.js/14.16.0  
node.js/6.10.3  
occa/1.2  
openblas/0.2.19  
openblas/0.2.8  
openblas/0.3.7  
opencv/3.2.0  
opencv/3.4.1  
parmetis/4.0.3  
pcre2/10.35  
phylobales/1.8b  
proj/4.9.3  
proj/5.2.0  
proj/7.0.0  
pytorch/1.3.1  
sparsehash/2.0.2  
suiteparse/4.5.4  
sysstat/12.5.4  
tensorflow/1.1.0\_cpu  
tensorflow/1.1.0\_gpu  
tensorflow/1.13.1\_cpu\_py3  
tensorflow/1.13.1\_gpu  
tensorflow/1.13.1\_gpu\_keras  
tensorflow/1.13.1\_gpu\_py3  
tensorflow/1.14.0\_gpu\_py36  
tensorflow/1.4.1\_cpu  
tensorflow/1.4.1\_cpu\_py3  
tensorflow/1.4.1\_gpu  
tensorflow/1.4.1\_gpu\_py3  
tensorflow/1.4.1\_gpu\_py3\_cuda9.1  
tensorflow/1.5.0\_cpu\_py3  
tensorflow/1.5.0\_gpu  
tensorflow/1.5.0\_gpu\_py3  
tensorflow/2.0.0\_cpu\_py37  
tensorflow/2.0.0\_gpu\_py37  
theano/1.0.1\_py3  
transdecoder/5.4.0  
v8/3.14.5

voro++/0.4.6

vrpn/7.33

whatshap/Sep2018

xeyes/1.0

xz/5.2.4

yaml-cpp/0.6.2

yaml-cpp/0.6.2\_intel2019.3

yaml-cpp/0.6.3\_intel\_2020.2

zlib/1.2.11

~~~~~ category: library

~~~~~

catch2/2.3          matplotlib/2.2.4      pmclib/1.1

dcmstk/3.6.6          multineat/3.10      pstokes/1.0

dotnet/5.0.202      nccl/2.4.7          sprng/5.0

easydict/1.7          nccl/2.8.4          statsmodels/0.9.0

fastq\_screen/0.13.0      openslide/3.4.1

gsutil/350          p7zip/16.02

~~~~~ category: machine learning

~~~~~

1/1.01                  py-faster-rcnn/July2018

1/2.03                  rapidjson/1.1.0

attend2u/20180216          scikit-learn/0.19.1

cuda/5.1                  scikit-learn/0.21.2

cuda/6.0                  sciml\_class/pytorch-21.06

cuda/7.0                  tensorflow/1.1.0\_cpu

cuda/7.4                  tensorflow/1.1.0\_gpu

cuda/7.6                  tensorflow/1.13.1\_cpu\_py3

cuda/7.6.5                  tensorflow/1.13.1\_gpu

cuda/8.1.0                  tensorflow/1.13.1\_gpu\_keras

cuda/8.2.0                  tensorflow/1.13.1\_gpu\_py3

deeparg/Jan2019                  tensorflow/1.14.0\_gpu\_py36

deeparg/Oct2018                  tensorflow/1.4.1\_cpu

deeplabcut/1.01                  tensorflow/1.4.1\_cpu\_py3

deeplabcut/2.03                  tensorflow/1.4.1\_gpu

deeplabcut/2.1.4                  tensorflow/1.4.1\_gpu\_py3

deeplabcut/2.1.9                  tensorflow/1.4.1\_gpu\_py3\_cuda9.1

deeplabcut/2.2                  tensorflow/1.5.0\_cpu\_py3

dlib/19.17                  tensorflow/1.5.0\_gpu

flann/1.8.4                  tensorflow/1.5.0\_gpu\_py3

hnn/1.0                  tensorflow/2.0.0\_cpu\_py37

horovod/0.16                tensorflow/2.0.0\_gpu\_py37  
horovod/0.19.5            tf-horovod/1.0  
pcl/1.9.1                xgboost/1.3.3  
pcl/1.9.1\_nurbs

~~~~~ category: math  
~~~~~

atlas/3.10.3            lapack/3.7.0            ripser/0.5.3  
blas/3.7.0            macaulay2/1.12-1        rss/1.0  
cgal/3.14.1            magma/V2.23-10        sage/8.7  
fenics/2017.1           magma/V2.25-5        sage/9.0  
fenics/2018.1.0        magma/V2.25-5-gpu    scalapack/2.0.2  
gap/4.9.1            magma-gpu/2.4.0        sympy/1.4  
gmsh/3.0.1            magma-gpu/2.5.4\_volta trlan/2010.09  
lapack/3.4.2           nlopt/2.5.0  
lapack/3.6.0           numpy/intel\_1.15.1

~~~~~ category: misc  
~~~~~

mark/Dec18            mysql/8.0.13  
mysql/5.7.28           netcdf4-python/4.1.4.2

~~~~~ category: ml  
~~~~~

bonito/0.3.1  
~~~~~ category: model  
~~~~~

gotm/5.0\_qingli gotm/5.3  
~~~~~ category: molecular dynamics  
~~~~~

mdanalysis/0.19.2  
~~~~~ category: mpi  
~~~~~

mpi/cave\_mvapich2\_2.3b\_gcc  
mpi/cave\_mvapich2\_2.3b\_intel  
mpi/cave\_mvapich2\_2.3rc2\_gcc  
mpi/hpcx\_2.7.0\_gcc\_10.2\_slurm20  
mpi/hpcx\_2.7.0\_intel\_2020.2\_slurm20  
mpi/mpich3.3a3\_intel\_2020.2  
mpi/mvapich2-2.3.5\_gcc\_10.2\_slurm20  
mpi/mvapich2-2.3.5\_intel\_2017.0\_slurm20  
mpi/mvapich2-2.3.5\_intel\_2020.2\_slurm20  
mpi/openmpi\_2.0.3\_intel\_2020.2\_slurm20

```
mpi/openmpi_3.1.6_gcc
mpi/openmpi_3.1.6_gcc_10.2_slurm20
mpi/openmpi_4.0.0_gcc
mpi/openmpi_4.0.1_gcc
mpi/openmpi_4.0.4_gcc
mpi/openmpi_4.0.4_gcc_8.3_slurm20
mpi/openmpi_4.0.5_gcc_10.2_slurm20
mpi/openmpi_4.0.5_intel_2020.2_cuda_9.1.85_slurm20
mpi/openmpi_4.0.5_intel_2020.2_slurm20
mpi/openmpi_4.1.1_gcc_10.2_slurm20
osu-mpi/5.3.2
osu-mpi/5.6.2_mvapich2-2.3a_gcc
~~~~~ category: mri
~~~~~

afni/17.1.00      freesurfer/6.0.0
afni/18.2.06      freesurfer/6.0.0_rh7
afni/19.0.17      freesurfer/6.0.0_rh7-30-Nov-2018
afni/19.3.08      freesurfer/7.1.1
afni/19.3.10      fsl/5.0.10
afni/20.0.03      fsl/6.0.0
afni/20.1.06      fsl/6.0.3
afni/21.2.04      mmvt/2020-06
ants/2.1.0        mricron/05-2016
ants/2.3.1        mricron/12-2012
ants/2.3.4        mricron/2019-09
dsi/april2019     tortoise/3.1.0
~~~~~ category: neuro
~~~~~

dtitk/2.3.1 qit/1.0  qit/2.0  qit/Jun21
~~~~~ category: neuroscience
~~~~~

sct/4.2.2
~~~~~ category: numerical optimization ~~~~~
bayop/1.0
~~~~~ category: other
~~~~~

sbt/1.2.3
~~~~~ category: package
~~~~~

ffte/6.0  fftf/6.0/mpi
```

~~~~~ category: performance

~~~~~

ior/2.10.3	ipm/2.0.6_r
ior/3.0.1	osu-mpi/5.3.2
ior/3.2.1	osu-mpi/5.6.2_mvapich2-2.3a_gcc
ipm/2.0.6	papi/5.4.3

~~~~~ category: physics

~~~~~

casa/5.1.2  
geant/4.10.04.p01  
geant/4.10.3  
geant/4.10.5  
geant/4.9.4.p04  
gpaw/0.10.0  
gpaw/1.2.0  
gpaw/1.2.0\_hpcx\_2.7.0\_gcc  
gpaw/1.2.0\_mvapich2-2.3a\_gcc  
gpaw/20.10.0\_hpcx\_2.7.0\_intel\_2020.2\_slurm20  
gpaw/20.10\_hpcx\_2.7.0\_intel\_2020.2\_slurm20  
gpaw/21.1.0\_hpcx\_2.7.0\_gcc\_10.2\_slurm20  
gpaw/21.1.0\_openmpi\_4.0.5\_gcc\_10.2\_slurm20  
gpaw/21.1.0a\_openmpi\_4.0.5\_gcc\_10.2\_slurm20  
hoomd/2.9.0  
jdftx/1.4.2  
lumericalfdtd/8.16.982  
mcx/20180525  
mcxlab/2017.7  
mujoco-py/1.50.1.23  
pymultinest/2.9  
root/6.10

~~~~~ category: profiling and debugging ~~~~~

|                |                           |
|----------------|---------------------------|
| cube/4.3.4     | ncdu/1.14                 |
| ddd/3.3.12     | scalasca/2.3.1_intel      |
| gdb/7.12.1     | scorep/3.0_intel_mvapich2 |
| gperftools/2.5 | valgrind/3.12.0           |

~~~~~ category: python

~~~~~

numpydoc/0.9.1	pytables/3.5.2	six/1.12.0
plotly/3.9.0	python_igraph/0.7.1	

~~~~~ category: python library



~~~~~

wx/1.0

~~~~~ category: software

~~~~~

annovar/2018Apr16 orthofinder/2.2.7 orthofinder/2.3.3

~~~~~ category: software management

~~~~~

maven/3.2.2 maven/3.8.1

~~~~~ category: solvers

~~~~~

eigen/3.2.2

eigen/3.3.2

eigen/3.4.0

mcl/12.135

mineos/1.0

mumps/5.0.2

mumps/5.0.2-seq

openfoam/4.1

openfoam/4.1a

openfoam/7.0\_hpcx\_2.7.0\_gcc\_10.2\_slurm20

pari/2.11.2

pastix/5.2.3

petsc/3.14.2\_hpcx\_2.7.0\_gcc\_10.2\_slurm20

petsc/3.14.2\_hpcx\_2.7.0\_intel\_2020.2\_slurm20

petsc/3.14.2\_mpich3.3a3\_intel\_2020.2

petsc/3.7.5

petsc/3.7.7

petsc/3.8.3

polyrate/17C

scotch/6.0.4

suiteparse/4.5.4

trilinos/12.12.1

~~~~~ category: stat

~~~~~

seaborn/0.10.0

~~~~~ category: statistical analysis package ~~~~~

glmnet/1.0

~~~~~ category: statistics

~~~~~

JAGS/4.2.0          R/3.6.3          caffe/1.0\_with\_cudnn

JAGS/4.3.0	R/4.0.0	mallet/2.0.8rc3
R/3.3.2	R/4.0.3	rstudio/1.0.44
R/3.4.0	R/4.0.5	rstudio/1.1.463
R/3.4.3	R/4.1.0	rstudio/1.4.1103
R/3.4.3_mkl	SAS/9.4M6	stata/14
R/3.4.4	SAS/9.4b	stata/15
R/3.5.2	caffe/1.0	wfu_pickatlas/3.0.5b
R/3.6.0	caffe/1.0_CPU_ONLY	

~~~~~ category: sys

~~~~~

dos2unix/7.4.0

~~~~~ category: system

~~~~~

wayland/1.18.0	xfce/4.10	xfce/4.16
wayland-protocols/1.20	xfce/4.12	

~~~~~ category: tex

~~~~~

pandoc/2.9.2.1

~~~~~ category: text editing

~~~~~

texstudio/2.12.16

~~~~~ category: tool

~~~~~

dcm2niix/25.0

~~~~~ category: tool kit

~~~~~

subread/1.6.2

~~~~~ category: tools

~~~~~

idr/2.0.2 omero/5.6.2

~~~~~ category: utilities

~~~~~

atom/1.19.3	git/2.10.2
bamaddrg/20180928	git/2.20.2
binutils/2.29.1	git/2.29.2
binutils/2.31	h4cf/1.2
chrome/55.0	h5py/2.9.0
chrome/73.0	intltool/0.51.0
cmake/3.10.1	itstool/2.0.4
cmake/3.15.4	json_cpp/1.9.4

cmake/3.20.0                json\_fortran/8.1.0  
cmake/3.6.3                kallisto/0.46.1  
cmake/3.8.0                libevent/2.1.8  
colordiff/1.0.18           libwnck/3.24.1  
comsol/5.2                mercurial/5.1  
comsol/5.6                ncurses/6.2  
cppunit/1.14.0            pdftk/2.02  
curl/7.61.1               perf-tools/2.7  
cvs/1.11.23               rclone/1.51.0  
depot\_tools/Jan2019       rsync/3.1.3  
engineering/19.10.237      spfft/0.9.12  
engineering/calibre2020.1\_36   sublime/2.0.2  
express/1.5.1            svn/1.8.17  
fileZilla/3.10.0           svn/1.9.5  
firefox/56.0.2            texinfo/4.13a  
firefox/59.0.2            texlive/2018  
firefox/66.0.3            vim/8.1  
firefox/68.0              vim/8.1\_py3  
firefox/87.0              vnc-apps/7.2  
ghostscript/9.21          xxdiff/4.0.1

~~~~~ category: utility

redis/6.2.4

~~~~~ category: version record for large files ~~~~~

gitlfs/2.7.1

~~~~~ category: vision

opencv/3.2.0            scikit-image/0.13.1    tesseract/3.05.00

opencv/3.4.1            scikit-image/0.15.0    tesseract/4.00.00

~~~~~ category: visualization

atomeye/3.0

basemap/1.2.0

blender/2.78

blender/2.79

blender/2.90.1

cartopy/0.16.0

cartopy/0.17.0

cave-demo/yurt

cave-utils/yurt

caviar/1.0  
ferret/7.1  
gdal/2.1.3  
gdal/2.4.0  
gdal/3.0.4  
gephi/0.9.2  
gimp/2.8.22  
gimp/2.9.6  
glm/0.9.9  
glm/0.9.9.5  
grace/5.1.25  
graphviz/2.40.1  
minvr/0.3master  
minvr/0.4  
minvr/beta  
minvr/beta-bleeding  
minvr/dev\_ben  
ncview/2.1.7  
openscad/2015.03-3  
openscenegraph/3.4.0  
osgearth/2.7  
ovito/2.8.2  
ovito/2.9.0  
paraview/5.1.0  
paraview/5.1.0\_yurt  
paraview/5.4.1  
paraview/5.6.0\_no\_scalable  
paraview/5.6.0\_yurt  
paraview/5.8.0  
paraview/5.8.0\_mesa  
paraview/5.8.0\_release  
paraview/5.8.1\_openmpi\_4.0.5\_intel\_2020.2\_slurm20  
paraview/5.9.0  
photoscan/1.3.0  
pyshp/2.0.1  
shapely/1.6.4  
tecplot/2017  
tecplot/2017R3  
tecplot/2018R2  
tecplot/2019R1

unity/Unity-2017.3.0b1

unity/unity-editor-2017.2.0f3

visit/2.13.2

visit/2.7.2

vmd/1.9.3

vrg3d/265

vrg3d/265\_compat

vrg3d/265\_demos

vtk/5.6.1

vtk/7.1.1

vtk/7.1.1a

vtk/8.1.0

xcrysden/1.5.60

~~~~~ category: vizualization

~~~~~

metashape/1.5.4

~~~~~ category: weather

~~~~~

wrf/3.6.1

wrf/4.2.1\_hpcx\_2.7.0\_intel\_2020.2\_slurm20

~~~~~ category: workshop

~~~~~

workshop/1.0 workshop/3.0

# Oscar: Sage

## Loading and Launching Sage

1. Once authenticated to Oscar, use the following commands at the command line.
2. Start an interactive job by using the `interact` command. This command can take additional parameters to extend the resources and time allotted to the node as well as the partition that the node operates on.
3. The Sage module provides containers. To load them use `module load sage-container/10.3`.
4. To start the container use `apptainer shell /oscar/rt/9.2/software/0.20-generic/0.20.1/opt/spack/linux-rhel9-x86_64_v3/gcc-11.3.1/sage-container-10.3-avpqi pfsnbneig726l72jrgdmlrivg4m/sage.sif`
5. Once inside the container's shell use `sage` to launch the Sage console.

## Sage on Oscar OnDemand

The easiest way to run Sage on Oscar OnDemand is to run sage in an interactive job via the terminal in your OnDemand session.

Use the `interact` command with parameters for your specific job to start the interactive session, then load your modules and run the sage binary (steps 2-4 above).

```
interact -n 2 -m 32g -t 04:00:00 -f 'haswell|broadwell|skylake'
```

## Using Sage with Batch Scripts

*Thanks to Trevor Hyde from Summer@ICERM 2019 for these instructions.*

One method for running computations with Sage on Oscar is to write a script and use the slurm batch scheduler to have Oscar run your script. This requires two pieces:

1. A shell script to configure and submit your batch job to the cluster.
2. Your Sage code/program you'd like to run.

# Example Batch Script

## sage-batch.sh

```
#!/bin/bash

#SBATCH -J test_program
#SBATCH --array=0-9
#SBATCH -t 1:00:00
#SBATCH --mem=8G

#SBATCH -e data/<oscar-username>/test_output/test%a.err
#SBATCH -o data/<oscar-username>/test_output/test%a.out

module load sage-container/10.3

apptainer shell /oscar/rt/9.2/software/0.20-generic/0.20.1/opt/spack/linux-rhel9-x86_64_v3/gcc-11.3.1/sage-
container-10.3-avpqipfsnbneig726l72jrgdmlrivg4m/sage.sif

sage test_program.sage $SLURM_ARRAY_TASK_ID
```

- `#!/bin/bash` tells the system this is a bash (shell) script.
- `#SBATCH -J test_program` sets the name of the job which appears when you check the status of your jobs.
- `#SBATCH --array=0-9` is an easy way of doing parallel computations. In this case it says our job will run on 10 different nodes, each node will be passed a parameter and we have specified that the parameters will take the values 0 through 9. You can specify several ranges or even list individual parameters if you prefer.
- `#SBATCH -t 1:00:00` specifies a time limit in `HH:MM:SS` for each node. Once this time runs out your program will stop running on that node. Be careful setting the time limit too high as doing so may make it take a long time for your job to get scheduled to run. Before starting a big computation try to do some smaller tests to see how long you expect to need.
- `#SBATCH --mem=8G` specifies how much memory each node gets. Standard exploratory accounts get 123GB total to use at any one time. So if you allocate too much per job, fewer jobs will run at once. On the other hand, if you allocate too little and a computation needs more than it has, then it will terminate. If this happens an “out of memory” error will show up in the `.err` file for that node.

- `#SBATCH -e data/<ccv-username>/test_output/test%a.err` and `#SBATCH -o data/<ccv-username>/test_output/test%a.out` specify where the error messages and output for each computation should be sent. You should store these files in your user folder, not on the submit node. We each have a folder inside the `data` directory which you can see from the submit node. In this example I have created a folder titled `test_output` where I'm putting both of these files. **You need to make these folders before you run the computation otherwise the output will be dumped into the void!** The `%a` will get replaced with the array parameter. So for example, since we set our array parameters to be `0-9` there will be 10 nodes running and each of them gets a number between 0 and 9; this node corresponding to the parameter 7 will create two files `test7.err` and `test7.out`.
- `module load sage-container/10.3` loads the sage container into the node.
- `apptainer shell /oscar/rt/9.2/software/0.20-generic/0.20.1/opt/spack/linux-rhel9-x86_64_v3/gcc-11.3.1/sage-container-10.3-avpqi pfsnbneig726l72jrgdmlrivg4m/sage.sif` initiates the container's Sage console shell.

Everything after this in the script happens as if you typed it yourself onto the command line.

- In our example, we want to run sage code, so the line `sage test_program.sage` `$SLURM_ARRAY_TASK_ID` runs our example sage program `test_program.sage`.
- The file needs to have the `.sage` extension.
- You should write this file in a text editor, not in a Jupyter notebook (although you can first write and test your program in a Jupyter notebook and then copy and paste it into a new file when it's ready).
- This program is written to accept one input and I have passed it `$SLURM_ARRAY_TASK_ID` which is the array parameter passed to each node. You can use this parameter to select which input parameters to run your program on.

## Example Sage Program

### `test_program.sage`

```
import sys

def fun_math(message):
    print message
    sys.stdout.flush()

job_id = int(sys.argv[1])
fun_math('hi this is a test')
fun_math('my job id is' + str(job_id))
```



- In the Sage program, you first define all of your functions and then you include the code you want to run.
- Import `sys` so you can access the array parameter passed to your function from the node. This is accessed in this case by `sys.argv[1]`. Make sure you explicitly coerce to be an integer if you want to use it as an integer; it's a string by default.
- The output of the `print` command is appended to the `.out` file for this node as a new line.
- Notice the line `sys.stdout.flush()` included in the function. This makes the program immediately send whatever output it has to the output file when called. Otherwise the program won't output **anything** until it has completely finished running. If each node is running 100 potentially long computations and it finishes the first 99 but then times out on the 100th computation, and you don't include any `sys.stdout.flush()` commands, everything will be lost when time runs out.

## Submitting the Batch Job

- To run this batch program go back to the submit node and type `sbatch` `<NAME_OF_BATCH_FILE>`. In our example here, our batch file is called `sage-batch.sh`, so we simply type `sbatch sage-batch.sh`. Slurm will return a line that tells you your job has been submitted together with a job id number.
- To check the progress of your jobs type `myq` from anywhere on Oscar. This will show you what jobs you have running, how much time they have left, and which jobs are still waiting to run. Be patient, sometimes it takes a minute for things to get started.
- If you realize your code is never going to finish or that you've made some terrible mistake, you can cancel a batch job by typing `scancel <JOB_ID>`. You can specify a single node or just put the general job id for the whole run and cancel everything.

# Oscar: MATLAB

## Loading and Launching MATLAB

1. Open the Terminal and use the following commands at the command line.
2. `module avail matlab` to list all the available matlab versions.
3. `module load matlab/R2018a` to load matlab. This command should return the confirmation “*module loaded*”.
4. `matlab` to launch the MATLAB app.

## Installing MATLAB Packages such as YALMIP

MATLAB script packages, such as YALMIP, can be installed directly by the user on their Oscar account. These instructions assume you are using Oscar via the VNC client.

1. Open the Terminal on your VNC session.
2. Navigate to your home folder by typing `cd ~`
3. `mkdir -p MATLAB`
4. `wget -O yalmip.zip https://github.com/yalmip/yalmip/archive/master.zip`
5. `unzip yalmip.zip`
6. In MATLAB, add the YALMIP-master directory to your path.
  1. In the MATLAB file browser, navigate to the MATLAB folder you created in your home folder. `cd ~/MATLAB`
  2. Right click on the YALMIP-master folder.

3. Select Add to Path > Selected Folders and Subfolders. This adds the YALMIP folders to your path.

7. To save your MATLAB path, use the savepath command in the MATLAB command prompt.

```
savepath ~/MATLAB/pathdef.m
```

YALMIP also requires a solver like SDPT3. The steps below add SDPT3 to MATLAB.

1. Open the Terminal.

2. `cd ~/MATLAB`

3. `wget -O sdpt3.zip https://github.com/sqlp/sdpt3/archive/master.zip`

4. `unzip sdpt3.zip`

5. In MATLAB, add the sdpt3 directory to your path.

1. In the MATLAB file browser, navigate to the MATLAB folder you created in your home folder. `cd ~/MATLAB`

2. Right click on the sdpt3-master folder.

3. Select Add to Path > Selected Folders and Subfolders. This adds the SDPT3 folders to your path.

6. To update/save your MATLAB path, use the savepath command in the MATLAB command prompt. `savepath ~/MATLAB/pathdef.m`

# Oscar: Mathematica

## Loading and Launching Mathematica

1. Open the Terminal and use the following commands at the command line.
2. `module avail mathematica` to list all the available mathematica versions.
3. `module load mathematica/11.0` to load mathematica. This command should return the confirmation *"module loaded"*.
4. `mathematica` to launch the Mathematica app.