**Midway cheat sheet**

# Connecting

On a Mac or Linux computer open a terminal and run the following command:

ssh [cnet\_ID@midway.rcc.uchicago.edu](mailto:cnet_ID@midway.rcc.uchicago.edu)

On a Windows machine, please download an SSH client and set it up as:

Hostname: midway.rcc.uchicago.edu

Port: 22

Username: CNet\_ID

Password: CNet password

# Storage

There are three types of storage available for you:

* Home directory: To store important files such as source codes, scripts, etc. It is small in size (25GB) and gets backed up regularly.
* Project directory: A persistent storage located at /project/CNetID or /project/group\_name to store files that are shared among members of a group. For example, /project/rossby has data files, etc. related to the workshop.
* Scratch space: High performance scratch space to be used for reading or writing data by your jobs. It is accessible via $HOME/midway-scratch symlink or /scratch/midway/CNetID. The size of scratch space is 100GB and data may be removed after some time. Once your job is finished, please make sure you move your data to either project or home directory.

# Transferring files

On Mac or Linux computers using the terminal run the following commands:

To copy a file from your computer to Midway:

* scp local\_file [cnet\_ID@midway.rcc.uchicago.edu](mailto:cnet_ID@midway.rcc.uchicago.edu):/home/CNetID

To copy a directory from your computer to Midway

* scp –r local\_folder [cnet\_ID@midway.rcc.uchicago.edu](mailto:cnet_ID@midway.rcc.uchicago.edu):/home/CNetID

Using a GUI-based tool such as WinSCP or FileZilla on Windows, Mac, or Linux computers setup a connection with the following information:

Hostname: midway.rcc.uchicago.edu

Port: 22

Username: CNet\_ID

Password: CNet password

# Finding software

Software is available by loading the proper module.

Show the list of available modules

* module avail

Show the list of available R modules

* module avail R

Show the list of currently loaded modules

* module list

Load the R module

* module load R (loads the default module)
* module load R/3.1 (loads version 3.1)

Unload the R module

* module unload R

Unload all the currently loaded modules

* module purge

# Installing a Python package

To install a system-wide Python package, please send an email to [help@rcc.uchicago.edu](mailto:help@rcc.uchicago.edu) and ask for the required package to be installed.

To install a Python package locally in your home directory, please run the following commands:

* module load python/desired\_version
* pip install --user missing-python-package
* export $PATH=$HOME/.local/bin:$PATH

The missing-python-package will be installed under your home directory (located at the /home/CNetID/.local folder). To use the installed package, you always need to load the python/desired-version module first. You could put the export command in the $HOME/.bashrc file to avoid running it every time after logging in.

# Installing an R package

To install a system-wide R package, please send an email to [help@rcc.uchicago.edu](mailto:help@rcc.uchicago.edu) and ask for the required package to be installed. If you are using a specific version of R on Midway, you also need to indicate the version of R that you want this package to be installed for.

To install an R package locally in your home directory, please run the following commands:

* module load R/desired-version
* export R\_LIBS=”$HOME/Your\_R\_libs"
* mkdir -p $HOME/Your\_R\_libs
* Within R run the install.packages(‘pkg\_name’) command

To avoid typing the export command every time you want to run R, put the export R\_LIBS=”$HOME/Your\_R\_libs” command in the $HOME/.bashrc file

The .libPaths() command within R shows where R searches for libraries

# Interactive session

To get a core on one node with for 3 hours and 2GB of memory on the amd partition:

* sinteractive --reservation=rossby --partition=amd   
  --account=rossby --time=03:00:00 --nodes=1 --ntasks-per-node=1 --mem-per-cpu=2000M

Running the following command will give you one core and 2GB of memory for 2 hours

* sinteractive --reservation=rossby --partition=amd   
  --account=rossby

Running the sample script /project/rossby/scripts/get\_interactive.sh will give you an interactive session with one core and 2GB of memory for 12 hours.

# Running IPython notebook

* Please make a copy of /project/rossby/Stephan-IPython folder to your home directory
* Get an interactive session
* Once you get the interactive session, run the /project/rossby/scripts/run\_ipython.sh script and open the displayed address in your browser (on your laptop)

# Submitting and monitoring jobs

Note: Jobs should run from the /scratch/midway/$USER

A simple sbatch script (saved as example.sbatch) to run a Matlab program on one core using 2GB of memory per core for 1 hour:

#!/bin/sh

#SBATCH --job-name=rossby\_test

#SBATCH --output=%j\_rossby\_test.out

#SBATCH --error=%j\_rossby\_test.err

#SBATCH --nodes=1

#SBATCH --ntasks-per-node=1

#SBATCH --mem-per-cpu=2000M

#SBATCH --reservation=rossby

#SBATCH --account=rossby

#SBATCH --partition=amd

#SBATCH --time=1:00:00

#make sure newFitModel.m code is in the following location

cd /scratch/midway/$USER/rossby\_project

module load matlab

matlab -nodisplay < newFitModel.m

To submit the above batch file:

* sbatch example.sbatch

To list a user’s submitted jobs:

* squeue –u username

To cancel a submitted job:

* scancel jobid

To see details of a running job:

* scontrol show job job\_id

# Running IPython notebook in batch

* Run the sbatch /project/rossby/scripts/ipython\_batch.sbatch command
* This command will submit a job and create a file called ipy\_address.txt
* Open the address written in the ipy\_address.txt file in your browser