

# DATASCI/STATS 531

## Parallel statistical computing in R on greatlakes

Edward L. Ionides

# Outline

- 1 Logging in to greatlakes
- 2 Moving files on and off greatlakes
- 3 Working with batch jobs
- 4 R modules on greatlakes
- 5 A test for foreach
- 6 Editing text files on greatlakes
- 7 pomp on greatlakes
- 8 Other ways to run R on greatlakes

# Requirements

We follow [Section 1.3](#) of the [greatlakes user guide](#). As preliminaries, you need:

- A Slurm account. Everybody in this class is a member of the account `stats531w24_class`. Graduate students in the Applied Statistics and Data Science masters programs, or Statistics PhD program, also have a primary departmental account, `stats_dept1`, and a smaller departmental backup account for if you exhaust your resources, `stats_dept2`.
- A greatlakes cluster login account. If you have not yet filled in the form at <https://arc-ts.umich.edu/greatlakes/user-guide/> then do so.
- A umich internet address. Use the umich VPN if you are not on campus.

# Connecting to greatlakes with macOS or Linux

- 1 Open a Terminal window (recall that, on a Mac, this can be done using Control-Spacebar and typing Terminal) and type

```
ssh username@greatlakes.arc-ts.umich.edu
```

where username is your username.

- 2 Login with your Kerberos level-1 password, and Duo two-factor authentication.

This creates a remote terminal shell on greatlakes.

## Connecting to greatlakes with Windows

This is essentially the same as for macOS, except that you may need to install a program that provides a terminal window

- 1 Follow instructions to install PuTTY at <https://documentation.its.umich.edu/node/350>
- 2 Launch PuTTY and enter `greatlakes.arc-ts.umich.edu` as the host name, then click open. If you receive a “PuTTY Security Alert” pop-up, this is completely normal, click the “Yes” option. This will tell PuTTY to trust the host the next time you want to connect to it. From there, a terminal window will open; you will be required to enter your UMich unixname and then your Kerberos level-1 password in order to log in. Please note that as you type your password, it may be that nothing you type appears on the screen; this is completely normal. Press “Enter/Return” key once you are done typing your password.
- 3 Complete the request for Duo two-factor authentication.

This creates a remote terminal shell on greatlakes.

## Moving files on and off greatlakes: scp

On Mac or Linux, you can use `scp` which has similar syntax to `cp`. To copy `myfile` on your laptop to a subdirectory `mydir` of your home directory on greatlakes:

```
scp myfile username@greatlakes-xfer.arc-ts.umich.edu:mydir
```

To copy an entire directory, use the `-r` flag for recursive copy:

```
scp -r mydir username@greatlakes-xfer.arc-ts.umich.edu:
```

These commands can also be reversed to copy files from greatlakes to your machine. The following copies `mydir` back to the current working directory:

```
scp -r username@greatlakes-xfer.arc-ts.umich.edu:mydir .
```

You will need to authenticate via Duo to complete the file transfer. On Windows, you can use WinSCP or FileZilla.

# Cluster batch workflow

- 1 You create a batch script and submit it as a job
- 2 Your job is scheduled, and it enters the queue
- 3 When its turn arrives, your job will execute the batch script
- 4 Your script has access to all applications and data
- 5 When your script completes, anything it sent to standard output and error are saved in files stored in your submission directory
- 6 You can ask that email be sent to you when your jobs starts, ends, or fails
- 7 You can check on the status of your job at any time, or delete it if it's not doing what you want
- 8 A short time after your job completes, it disappears

# Useful batch commands

## Submit a job

```
sbatch sample.sbat
```

## Query job status

```
squeue -j jobid  
squeue -u username
```

## Delete a job

```
scancel jobid
```

## Check a job script and estimate its start time

```
sbatch --test-only sample.sbat
```



## More Slurm commands to try

<code>sacct -u user</code>	show recent job history
<code>seff jobid</code>	show cpu utilization for jobid
<code>my_accounts</code>	list accounts you have permission to use

# R modules on greatlakes

Software on greatlakes is packaged in modules which must be loaded

```
module load R
```

Other versions of R are available:

```
module avail R
```

- We see that R4.3.1 is currently the default. For simple multicore computing, sending jobs to multiple cores on a single node, the default R module is appropriate.

# Set up test for foreach

- The greatlakes subdirectory of the 531w24 git repository has a file `test.sbat` which submits a batch job running the parallel foreach test in `test.R`.
- A basic Linux exercise is to set up a directory on greatlakes with these files, at which point you can run

```
sbatch test.sbat
```

to submit the job. If you have little or no experience with Linux or Unix, this is a nontrivial task. You could ask for help or read <https://ubuntu.com/tutorials/command-line-for-beginners>.

- You can transfer the files from your laptop via `scp`, or by copy-paste, but it may be simplest to clone the class git repository into your greatlakes account,

```
git clone https://github.com/ionides/531w24.git
```

# Editing text files on greatlakes

- Inspect the text file `test.sbat`, for example by

```
more test.sbat
```

Is it fairly self-explanatory?

- One thing that needs changing is to set your email address for alerts about jobs beginning and ending.
- To make these edits on greatlakes, you need a text editor.
- It is convenient to use a text editor that runs in a terminal. Options include

```
vi test.sbat  
emacs -nw test.sbat  
nano test.sbat
```

- It is useful to acquire some familiarity with each of these editors.

# Comparing results

- You are now ready to run a batch job

```
sbatch test.sbat
```

- From inspecting the code in `test.R`, we see that the results are saved in `test.csv`
- You can assess what you learn from comparing run times on greatlakes and your laptop, though the main goal here is just to practice running the code.

# Installing packages, including pomp

- After making R available by running

```
module load R
```

you can start an R session in a terminal on the login node just by typing R.

- This is useful for setting up all the R libraries you may need.
- You are not supposed to do heavy multi-core computing on the login node, but installing libraries and small tests is okay.
- For example, if I run the following:

```
[ionides@gl-login1 ~]$ module load R
[ionides@gl-login1 ~]$ R
> install.packages("tidyverse")
> install.packages("pomp")
```

then my subsequent R jobs run via sbatch are able to use `library(tidyverse)` or `library(pomp)`.

# Other ways to run R on greatlakes

- It is sometimes useful to start an interactive session on greatlakes, particularly for debugging. This is done from the terminal as follows:

```
module load R
srun --nodes=1 --account=stats531w24_class --ntasks-per-node=8 \
    --pty /bin/bash
```

- You can then run R in the terminal as usual, just by typing

```
R
```

- This R session will have access to the cores you have requested.
- Here, we require `nodes=1` since `library(doParallel)` alone cannot work with cores spread across different machines.
- You can also run [web-based Rstudio](#) However, batch jobs remain the basic tool for intensive statistical computing.

# Acknowledgments

- This lesson builds on the [Greal Lakes User Guide](#), an introduction by [Charles Antonelli and John Thiels](#), and notes from [STATS 810](#).
- Compiled on March 12, 2024.