

Homework 12. Due by 5pm on Thursday 11/19.

Statistical computing on greatlakes

From homework 11, we understand the relevance of being able to implement embarrassingly parallel calculations. In the context of statistical computing in R, we are motivated to learn to use `foreach` on the greatlakes cluster. In this homework we carry out a simple example, since many of us are new to Linux clusters in general and greatlakes in particular. However, once you can run a multicore `foreach` on greatlakes you are prepared for a large fraction of all statistical parallel computing tasks. Work through the questions below and write brief answers to the following questions, by editing the tex file available at <https://github.com/ionides/810f20>, and submit the resulting pdf file via Canvas.

1. R has a reputation for having appealing flexibility for data analysis but the potential to be slow. Some for loops can be slow, though this may be avoidable by careful coding (<https://adv-r.hadley.nz/control-flow.html#common-pitfalls>). If R run time starts limiting your research—i.e., whenever there is a numerical result you would like to see but you don't have the patience to wait for it to be computed—you have various options, including:
 - (a) Use smaller examples.
 - (b) Write the critical part of your code in C called from R. This can make use of the C functions underpinning the R language (<https://cran.r-project.org/doc/manuals/R-exts.html#The-R-API>).
 - (c) Work on rewriting your R code to make it more efficient.
 - (d) Run your code on a more powerful machine, maybe moving from your laptop to greatlakes.

Comment on situations when each approach might be suitable. Do you have advice, or relevant experiences?

YOUR ANSWER HERE.

2. Follow the instructions at <https://ionides.github.io/810f20/gl/slides.pdf> to run a job testing `foreach` on greatlakes. Report back on your results, and any problems that arose or advice to share.

YOUR ANSWER HERE.