Colorado School of
PUBLIC HEALTH

University of Colorado
Anschutz Medical Campus

Statistical Methods Research: Running Simulations with R in a High Performance Computing Environment

Peter E. DeWitt, M.S., Ph.D. Candidate
peter.dewitt@ucdenver.edu

Colorado School of Public Health
University of Colorado Denver
Department of Biostatistics and Informatics

14 FEBURARY 2017

My Project

- Working towards a Ph.D. in Biostatistics
- Explain the interaction between day-of-cycle, age, and time-to-menopause observed in the changes to reproductive hormone profiles expressed during the menopausal transition.
- Aim 1: Parsimonious B-Spline Regression Model via Control Polygon Reduction (CPR)
- I've developed a novel method selection approach for enumeration and placement of knots for B-splines.

Why I Needed A HPC

- Simulations to show how CPR compares to
  - A standard likelihood-based forward-step model selection approach,
  - A standard likelihood-based backward-step model selection approach,
  - P-splines

- Start with L internal knots:
  - CPR requires $L + 1$ regression fits,
  - Forward-step requires $L + 1$ regression fits,
  - Backward-step requires $L(L + 1)/2 + 1$ regression fits,
  - P-splines requires $L + 1$ regression fits.

- For $L = 80$ a total of 3,484 regression models to fit.

Why I Needed A HPC

- Simulations:
  - Four functions.
  - Modeling under ordinary least squares and under mixed effect models.
  - Differing sample sizes, model, and subject specific errors.
- 144 sets of initial conditions.
- $501,696,000$ total regression models to fit.
- As a beta user I used:
  - 176,742 core hours
  - \$0.116 per core hour = \$20,502.07

My HPC Pipeline

Hardware, Software, RAM, and Disk Usage

- Hardware
  - A lot of computation cores
- Software
  - SLURM
  - R
  - tar
- RAM
  - Low enough that the 4GB per core was default sufficient.
- Disk Space
  - Less than 2 GB, tarballs total 633 MB.
  - A lot of small files where generated.
  - Originally only had 50,000 inode limit on `$HOME` and `$SCRATCH`, extended to 100,000 inode limit.

Difficulties

- Number of Files.
  - Solution: generate a controlled number of files and place into a tarball for each set of initial conditions.

- Getting a Job to run to completion.
  - Option 1: Request lots of nodes and cores.
    - Use GNU parallel to manage the run.
    - Did not work well for me.
    - Queue time for resources was high.
    - Getting a good estimate for the time required to run the sim was difficult. (More on this later.)
  - Option 2: Use SLURM Arrays to request a few resources many times.
    - This worked well for me.
    - Less queue time.
    - Leaves more resource open for others.

Estimating Time Need For An Iteration

- Differences between my desktop and Rosalind

	Rosalind	My Desktop
OS	Red Hat Enterprise 7.2	Debian 8.7 (jessie)
CPU	Intel(R) Xeon(R) E5-2680 v3	Intel(R) Core(TM) i7-3770
Speed	2.50GHz	3.40GHz

- I have configured my desktop to use OpenBLAS for linear algebra.
- QR Decompositions is the biggest time consuming part of my work.
- Benchmarking on desktop was not helpful.
- My one wish for Rosalind: A Testing QOS.

My Workflow

- Three types of files
  1. R scripts for generating one data set and running the sim.
  2. bash script for submitting a job
  3. bash script for submitting all jobs.

- Examples follow.

Example R Script

```
# file: doppler-lmer-sim.R
#
# Five input args
# s obs_per_subject sigma_alpha sigma_epsilon outdir

inparams <- commandArgs(trailingOnly = TRUE)

# Build a data set
# ....

# Run the sim
# ....

# Save results
# ....
```

Example Job Submission File

```
#!/bin/bash
# file: doppler-lmer-sruns.sh

#SBATCH --nodes 1
#SBATCH --ntasks-per-node=20
#SBATCH --cpus-per-task=1

if [ $# -ne 5 ]; then
  echo "expected five arguments"
  exit 1
fi

module load R

mkdir -p $5
mkdir -p $5/results
mkdir -p $5/timers
mkdir -p $5/logs

Rscript --vanilla doppler-lme4-sim.R $1 $2 $3 $4 $5 > \
  $5/logs/log.${SLURM_ARRAY_JOB_ID}_${SLURM_ARRAY_TASK_ID} &
# ... omitted lines ...
Rscript --vanilla doppler-lme4-sim.R $1 $2 $3 $4 $5 > \
  $5/logs/log.${SLURM_ARRAY_JOB_ID}_${SLURM_ARRAY_TASK_ID} &
wait
```

Example Submit all Jobs

```
#!/bin/bash

# submitjobs.sh

sbatch --array=1-50 --time=01:00:00 --job-name=d2a \
  doppler-lme4-sruns.sh 2   100 0.1 0.1 dop2-sa01-se01-n0100

sbatch --array=1-50 --time=02:00:00 --job-name=d2b \
  doppler-lme4-sruns.sh 2   500 0.1 0.1 dop2-sa01-se01-n0500

sbatch --array=1-50 --time=04:00:00 --job-name=d2c \
  doppler-lme4-sruns.sh 2  1000 0.1 0.1 dop2-sa01-se01-n1000

# ... omitted lots of rows ...
# ... omitted lots of rows ...
```

Interesting Results

- Control Polygon Reduction vs. other model selection tools:
  - Faster,
  - Regression models with as good, or better fits, on a degree-of-freedom for degree-of-freedom comparison.

- Want to know more:
  - Dissertation defense April 10.
  - Control Polygon Reduction,
  - Control Net Reduction,
  - Software, and
  - Clinical inference.

Wish List

- A Testing QOS
  - At least two Nodes.
  - Low wall time.
  - High priority use.
- Add to the "Getting Started" document
  - Limits on storage, size and quantity.
  - Details on queueing priority.
- Community site to hosting examples, Q&A.
- Configuration and Testing to minimize cost.
  - 20 parallel jobs, one core each, or
  - 10 parallel jobs, two cores each, or
  - 5 parallel jobs, four cores each?