LARVA Update: Performance Improvements and P-value Calculations

Featuring the return of LL and his A.N.T.I.C.S.

(Amazingly Nifty and Timely Insectoid Computational System)

~~Annotation~~ Variation subgroup

2014-03-17

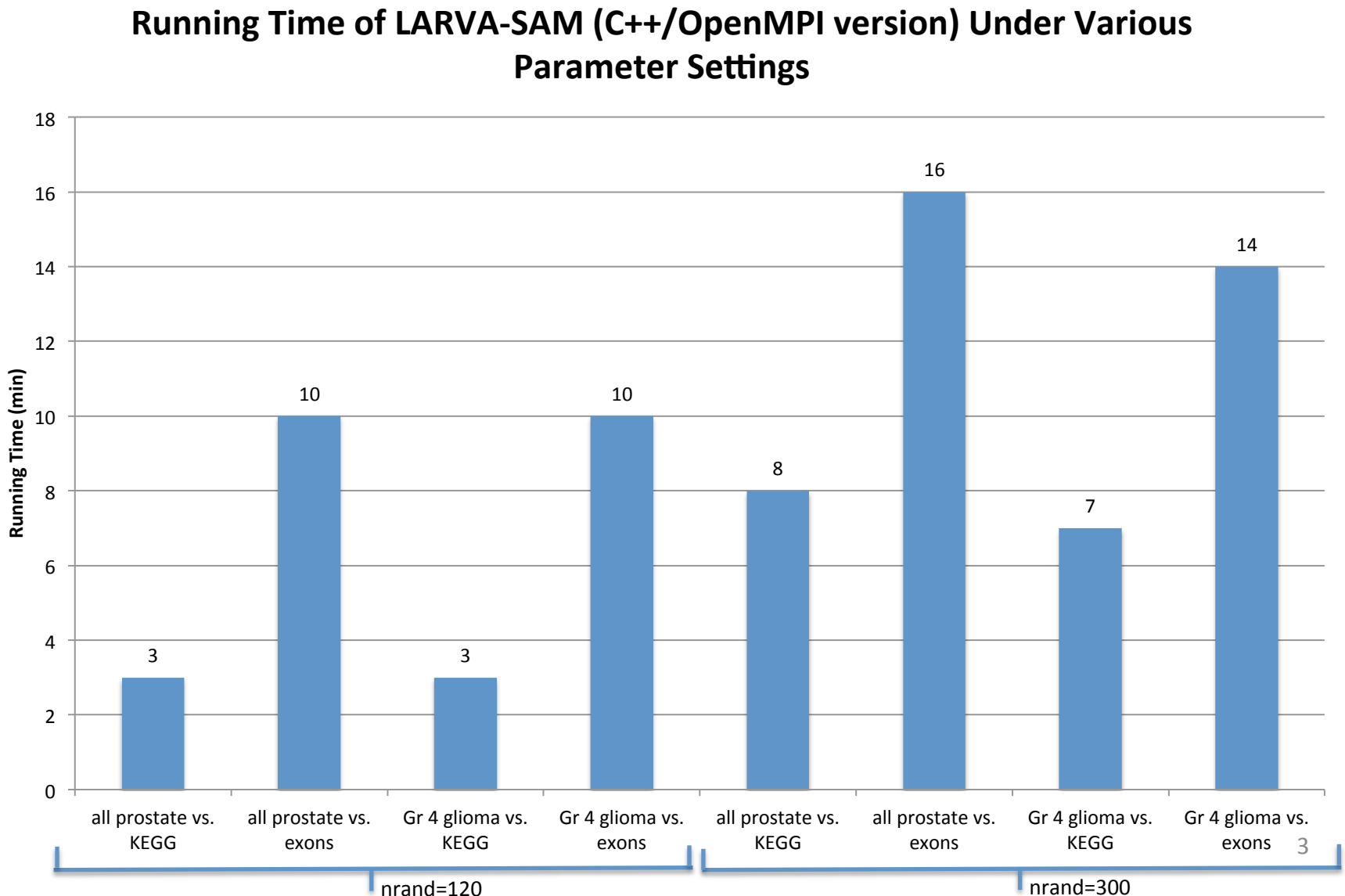
Previously…

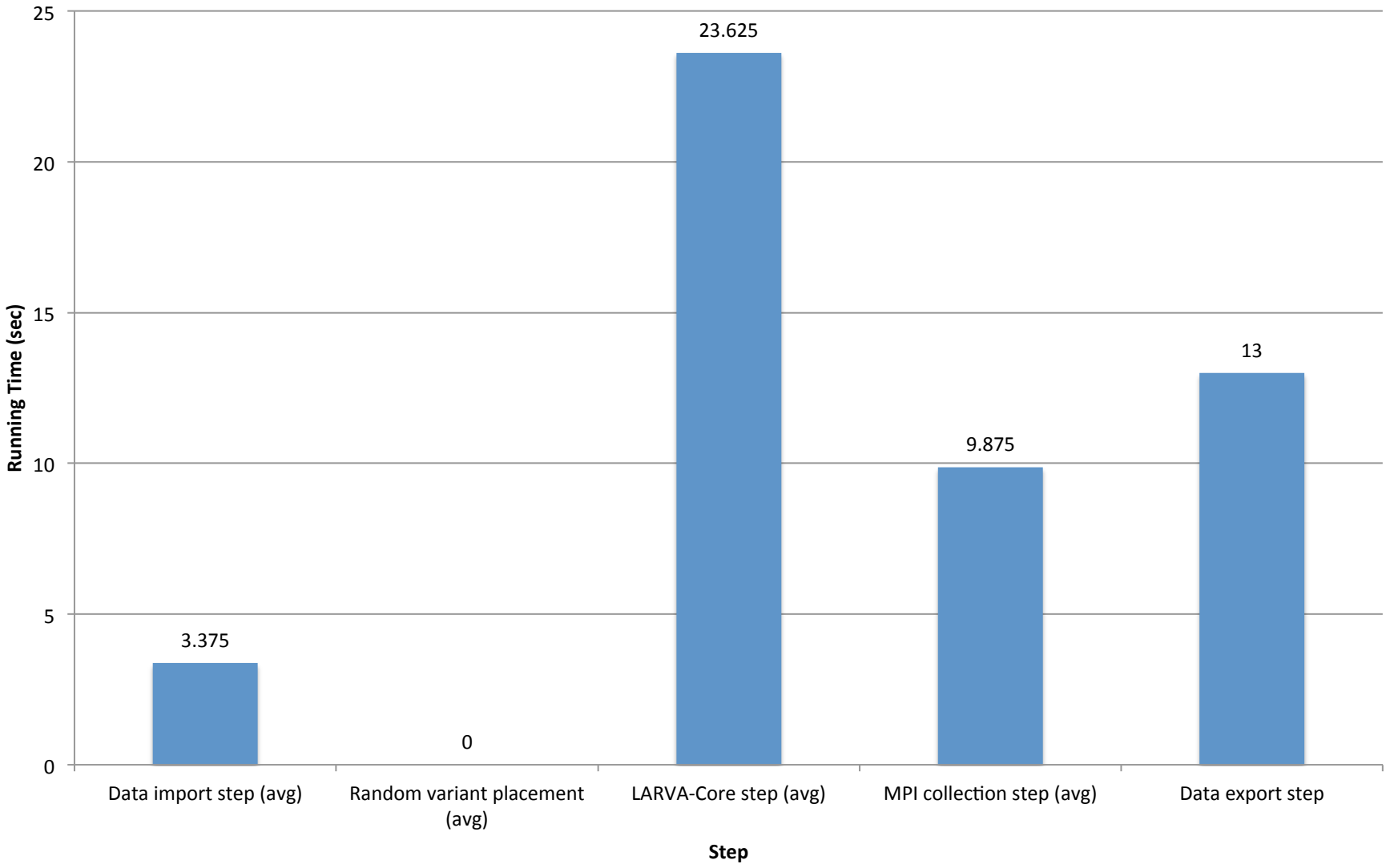
- Discovered that LARVA-SAM's efficiency was held back by the use of files to communicate results between processes
  - CPAN Parallel::ForkManager module has the same problem
- Decided to migrate to true interprocess communication with OpenMPI
- Interface between Perl and MPI was kludgy
- Decided to rewrite the entire software suite in C++, since the Perl MPI module was translating all the MPI commands into C anyway

And now…

Same scalability as before

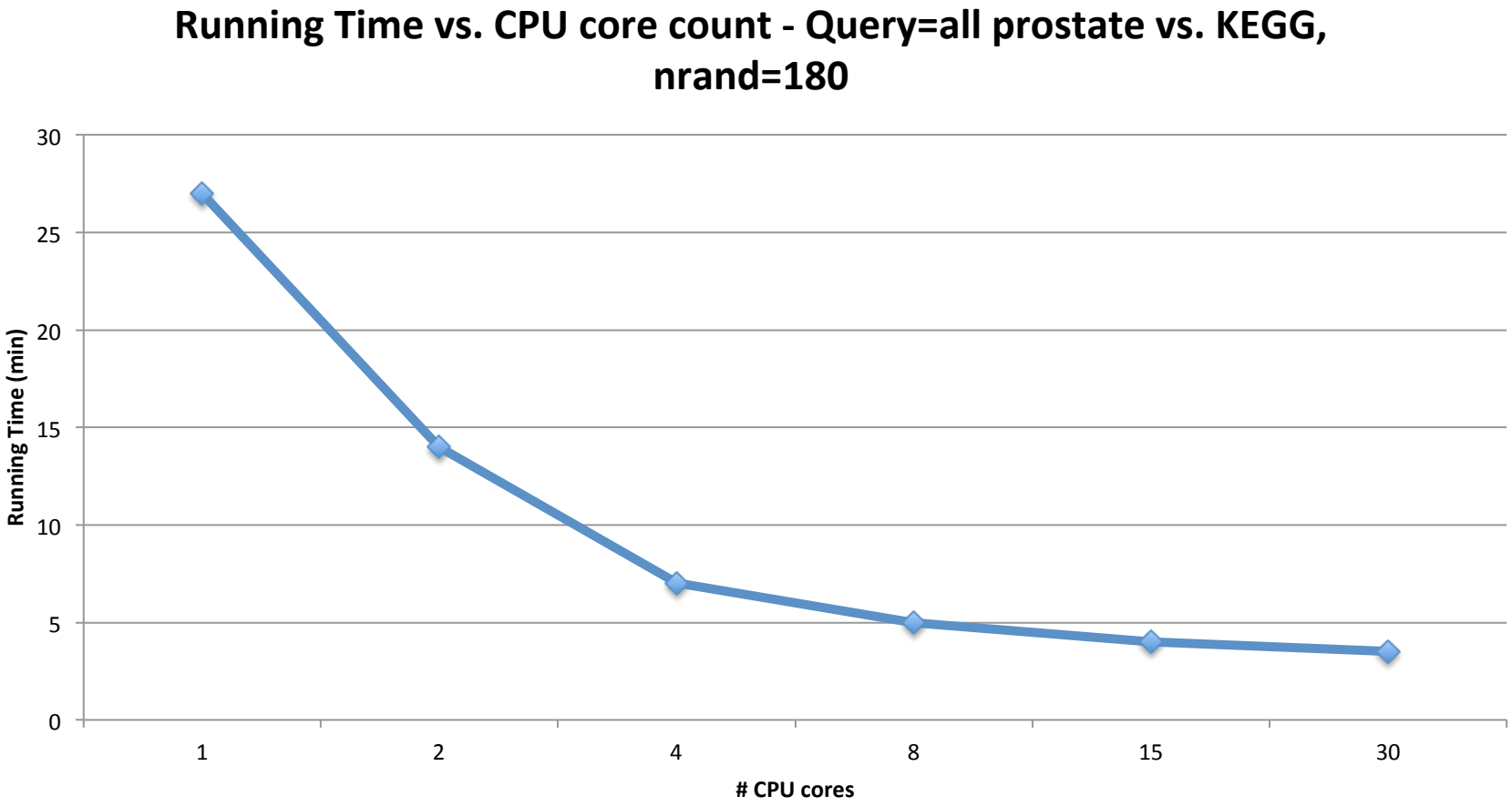
Up to 40x speedup over Perl version!





New code profile indicates that none of the support steps dominates over the LARVA-Core step. And apparently, C++ has a much more efficient random number generator.

CPU core count influence on performance



Diminishing returns as # CPU cores increase, but what is the optimal number of cores?

Determining the Optimal Number of CPU Cores

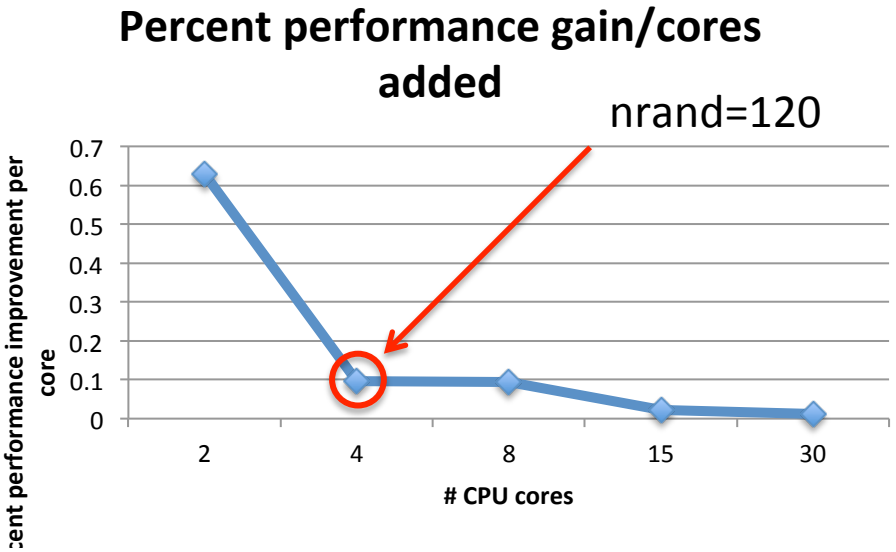
- Decided to determine this based on the performance gain relative to the number of CPU cores added
- For two timing tests t1 and t2, where t1.ncpu < t2.ncpu, calculate performance gain as follows:

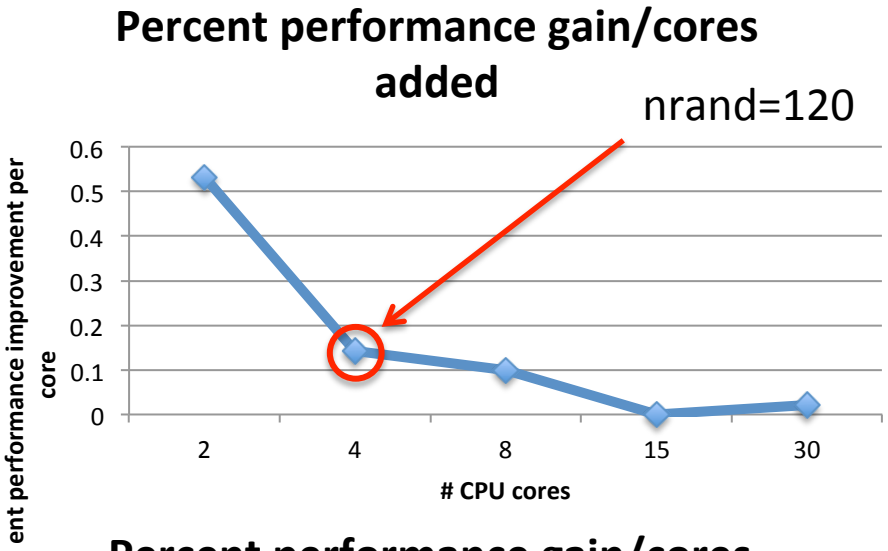
$$\frac{\left|t2.running\_time - t1.running\_time\right| / \left(t1.running\_time\right)}{t2.ncpu - t1.ncpu}$$

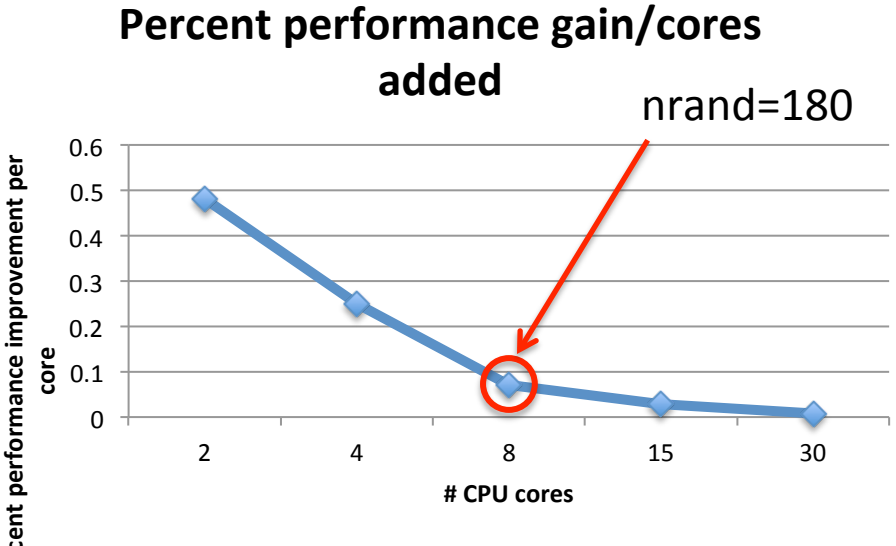
- **Example:**
  - t1: 2 cores, 23 min
  - t2: 4 cores, 13 min

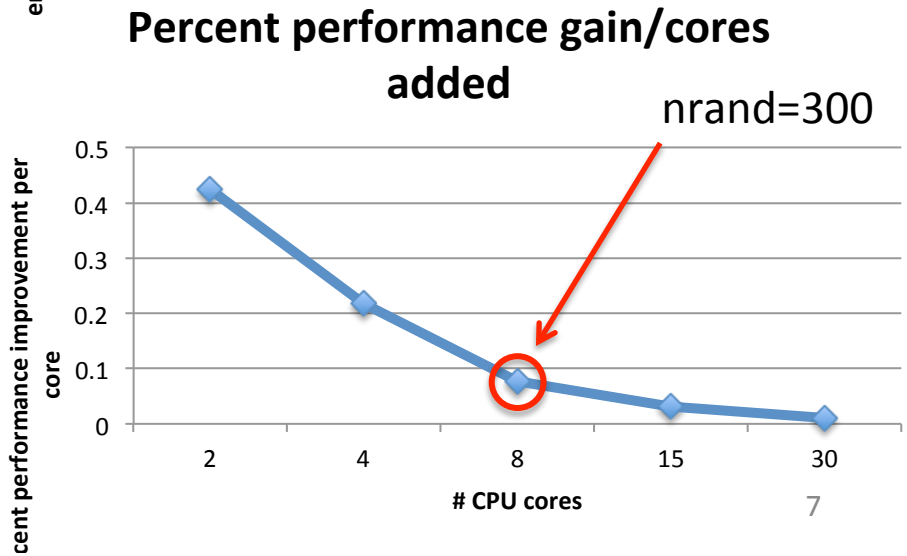
$$Perf\_gain = \frac{\left|13 - 23\right| / \left(23\right)}{4 - 2} \approx 0.217$$

Determining the Optimal Number of CPU Cores









Other Planned Features of this Presentation

- Perl version of LARVA could only do *nrand* of a few hundred feasibly
  - Found that significance test p-values varied by orders of magnitude at that level
  - Running on much higher *nrand* important
- Evaluation of C++ LARVA at *nrand* approaching 10,000
  - And determine variance of p-values
  - Find threshold *nrand* at which p-value does not change appreciably at higher *nrand* values
- Goblin01 server crashed over the weekend
  - Results not available