SPRINT: a Simple Parallel INTerface to High Performance Computing and a Parallel R Function Library.

Muriel Mewissen[1], Thorsten Forster[1], Terry Sloan[2], Savvas Petrou[2], Michal Piotrowski[2], Bartek Dobrelewski[2], Peter Ghazal[1], Arthur Trew[2], Jon Hill[3]

1. Division of Pathway Medicine, The University of Edinburgh, Chancellor's Building, 49 Little France Crescent, Edinburgh, EH16 4SB, UK.
2. Edinburgh Parallel Computing Centre, James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ, Uk.
3. AMCG, Earth Science and Engineering, Imperial College, London, SW7 2AZ, UK.

Keywords: High Performance Computing, Bioconductor, Correlation, Permutation, Microarray

The analysis of post genomic data is increasingly becoming harder to perform on standard computing infrastructures due to the sheer amount of data involved requiring more disk space and longer processing times. High Performance Computing (HPC) is an obvious answer to the need for more computing power. Access to computer clusters is common now with HPC resources becoming available to all through local or national initiatives such as the UK supercomputing service HECToR. However, the transition from general computing, such as *R* Language and Environment for Statistical Computing, to parallel computing is not straight forward. Software application and tools have to be adapted to take advantage of the extra computing power. SPRINT aims to provide bioinformaticians using *R/Bioconductor* to analyse microarray data with easy access to HPC providing maximum performance but requiring minimal expert knowledge and minimal changes to existing *R* scripts. The SPRINT framework consists of an HPC harness and a library of parallelized R functions. SPRINT is very flexible; it runs on a range of HPC systems and allows the addition of user contributed functions. It handles functions that are trivial to parallelize, functions that are non trivial to parallelize and functions generating very large output.

The SPRINT parallel harness is written in *C* and uses the Message Passing Interface (**MPI**) library. It takes as input the *R* script and data to be analyzed. The use of the parallel IO support of the **MPI** library helps ensure that the results can be output in parallel providing great scalability. The use of **ff** objects from the **ff** package which allows the manipulation of large objects on file almost as if they were in memory, also removes the limitation on the size of the data that can be successfully analyzed. SPRINT can therefore handle very large amount of data increasing further its scalability.

The SPRINT library currently includes a parallel implementation of functions that have been highlighted as bottlenecks in the analysis of post genomic data by a user requirement survey. These include a Person pair-wise correlation (`cor` from **stats**) and a permutation test function (`mt.maxT` from **multtest**). Benchmarking runs on the HECToR Cray XT system have shown an almost perfect scaling to 512 processors.

References

J. Hill (2008). SPRINT: A new parallel framework for R. *BMC Bioinformatics*, 9, 558.

(2010). *SPRINT: A new parallel framework for R*, http://www.r-sprint.org/.

(2010). *ECDF Edinburgh Compute and Data Facilities*, http://www.ecdf.ed.ac.uk/.

(2010). *HECToR: UK National Supercomputing Service*, http://www.hector.ac.uk/.