## Using SPRINT and parallelised functions for analysis of large data on multi-core Mac and HPC platforms

Eilidh Troup<sup>1\*</sup>, Thorsten Forster<sup>2</sup>, Luis Cebamanos<sup>1</sup>, Terence Sloan<sup>1</sup>, Peter Ghazal<sup>2</sup>

- 1. Edinburgh Parallel Computing Centre, University of Edinburgh, Edinburgh, UK
- 2. Division of Pathway Medicine, University of Edinburgh Medical School, Edinburgh, UK \*Contact author: e.troup@epcc.ed.ac.uk

Keywords: HPC, Big Data, Genomics, SPRINT, Parallelisation

We here present computation performance (CPU time, memory requirements) increases we can obtain in the analysis of large biological (or other) data sets through use of the **SPRINT** package (www.r-sprint.org).

With the arrival of "big data" (microarrays, screens, next-generation sequencing) in the life sciences, standard analyses of these data for regular users of *R* now run into severe issues of computation time or computer memory. Many projects (including parallelisation efforts of the *R* core) offer *R* packages and functions that allow programming of solutions for large-scale analysis problems. However, these usually require familiarity with HPC programming as well as sufficient and funded time to employ, which is feasible for one-off analysis problems but impractical for common analysis methods.

To make High Performance Computing (HPC) solutions available to *R* users without HPC experience, we started development on the **SPRINT** package in 2008. It allows these users straightforward use of already implemented parallelised versions of many relevant *R* functions on multi-core Macs as well as large-scale clusters/HPC platforms like the UK's HECToR or ARCHER (we have also tested on Amazon Elastic Compute Cloud). In addition to addressing speed-critical problems, we also address memory-critical problems.

We will here introduce recent upgrades to **SPRINT**, discuss for regular *R* users how to use **SPRINT** and for users with HPC background how our parallelisation strategies are particularly aimed at problems that go beyond 'simple' task farming. We outline case examples for use of **SPRINT** as well as performance and limitations of our approach in context of biological high-throughput data (although most individual functions are generically usable for other larger data sets).

Based on our needs and those we established in R user surveys, we currently support parallelised versions [1] of original [2] functions (our function names add prefix 'p', apart from pmaxt, which is based on mt.maxT) that are essential in clustering, classification and non-parametric statistics when applied to very large data sets: pstringdistmatrix, pboot, papply, pcor, ppam, prandomForest, pmaxt, pRP, psvm.

## References

- [1] Publications of our function implementations can be found on www.r-sprint.org -> Publications
- [2] Source citations for these packages can be found on www.r-sprint.org -> Overview and R functions