Use of molecular markers to estimate genomic relationships and marker effects: computation strategies in R

Filippo Biscarini1, Andrea Pedretti1, Ulrike Ober2, Malena Erbe2, Hossein Jorjani3, Ezequiel Nicolazzi1 and Matteo Picciolini1

1. PTP (Parco Tecnologico Padano), Via Einstein - Loc. Cascina Codazza, 26900 Lodi (Italy)
2. Department of Animal Sciences, Georg-August-Universität, Albrecht-Thaer Weg 3, 37075 Göttingen (Germany)
3. Interbull Centre, Box 7023, S-75007 Uppsala (Sweden)

*Contact author: filippo.biscarini@gmail.com

Keywords: SNPs, matrix of genomic relationships, matrix inversion, parallel R, GPU

The R programming environment for data analysis has acquired popularity among scientists of different fields. Main reasons are the availability of many packages for statistical analysis, the flexible programming syntax suited for tabular data, and the open source philosophy. One limitation of R may lie in the speed of computation and in the capacity of dealing with large datasets. Large amounts of data are common in many areas of science and technology. Examples are high-density marker panels and whole-genome sequences (e.g. 30-odd million SNPs in the sheep genome) in bioinformatics and genomics. Computational strategies can be devised to deal with big data, such as parallel computing (partitioning a larger computation problem into sub-problems on different processors), and the use of the high-performing graphical processing unit (GPU, designed to handle the complex operations and computation load of the processing and rendering of images). Both approaches are especially suited to deal with issues of computational speed.

We explored the use of parallel computing and of the GPU processor to analyse large datasets in R. As working example, we chose an application in genomics: with a simulated set of 1000 genetic markers (SNPs, single nucleotide polymorphisms) and 4000 individuals Jorjani (2009), we set up the matrix $G$ of genomic relationships between individuals VanRaden (2008). From this matrix we derived the SNP marker effects for a phenotype with mean 0 and standard deviation 10: this involves the calculation of the inverse of $G$, a computationally intensive operation whose processing time is known to increase quadratically with the number of individuals. Initial calculations started with 1000 individuals: the population was then increased in steps of 100 individuals until reaching 4000 individuals. We compared the relative performances, in terms of computation speed, of the standard serial use of R with parallel and GPU computing. All calculations were performed on the same platform (machine and settings). The chosen problem involved the manipulation, inversion and multiplication of matrices and vectors: some of these operations could be vectorised, thus allowing for parallel computation. Parallelization was implemented in R using the package parallel. Cuda architecture and Cublas libraries were used for calculations on the GPU. Functions for the binding on the GPU and for some matrix operations (e.g. Cholesky factorization and Gauss-Jordan elimination) were written in C and wrapped in R. First results showed that, compared with the standard R computations, the parallel R and GPU implementations were faster on average by 17.4%, and 40.9%, respectively. The difference in computation speed was almost negligible with small matrices, but increased progressively with matrix size: for the largest matrices, the parallel R and GPU implementations were both $\sim 75\%$ faster than standard R. Besides, for more than 3200 individuals the computations in standard R were no longer possible, whereas the parallel R and GPU implementations could both reach the full data-size (4000 individuals). On average, the GPU implementation was faster than parallel R computation on the CPU by 25.8%.

References
