BIOINFORMATICS APPLICATIONS NOTE Vol. 26 no. 11 2010, pages 1468–1469
doi:10.1093/bioinformatics/btq147

Genetics and population analysis

Advance Access publication April 7, 2010

FastEpistasis: a high performance computing solution for

Thierry Schüpbach[1,2], Ioannis Xenarios[1], Sven Bergmann[2,3] and Karen Kapur[2,3,*]

[1]Vital-IT Group, [2]Molecular Modeling Group, Swiss Institute of Bioinformatics, Lausanne and [3]Department of Medical Genetics, University of Lausanne, Rue du Bugnon 27, CH-1005 Lausanne, Switzerland

Associate Editor: David Posada

ABSTRACT

Motivation: Genome-wide association studies have become widely used tools to study effects of genetic variants on complex diseases. While it is of great interest to extend existing analysis methods by considering interaction effects between pairs of loci, the large number of possible tests presents a significant computational challenge. The number of computations is further multiplied in the study of gene expression quantitative trait mapping, in which tests are performed for thousands of gene phenotypes simultaneously.

Results: We present FastEpistasis, an efficient parallel solution extending the PLINK epistasis module, designed to test for epistasis effects when analyzing continuous phenotypes. Our results show that the algorithm scales with the number of processors and offers a reduction in computation time when several phenotypes are analyzed simultaneously. FastEpistasis is capable of testing the association of a continuous trait with all single nucleotide polymorphism (SNP) pairs from 500 000 SNPs, totaling 125 billion tests, in a population of 5000 individuals in 29, 4 or 0.5 days using 8, 64 or 512 processors.

Availability: FastEpistasis is open source and available free of charge *only for non-commercial users* from http://www.vital-it.ch/software/FastEpistasis

Contact: karen.kapur@unil.ch

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on February 24, 2010; revised on March 31, 2010; accepted on March 31, 2010

1 INTRODUCTION

Genome-wide association studies (GWASs) have been instrumental in identifying genetic variants associated with complex traits such as human disease or gene expression phenotypes (Hirschhorn *et al.*, 2005). While many GWAS results have been reported analyzing single nucleotide polymorphisms (SNPs) one-at-a-time, only recently have studies begun to extend analysis methods to consider interaction effects between pairs of loci (Cordell, 2009; Curtis, 2007; Emily *et al.*, 2009; Gayan *et al.*, 2008; Herold *et al.*, 2009).

Although interactions may yield new insight into the effect of genetics on complex traits (Manolio *et al.*, 2009), a major challenge to studying interactions is due to the large number of possible tests, which need to be considered. Examining all pairwise interactions between two SNP loci using a 500000 SNP chip equates to performing 125 billion tests. Additionally, carrying out permutation tests or studying epistasis in the context of quantitative trait mapping of gene expression, in which genetic variants are tested for association with each of thousands of phenotypes simultaneously (Franke *et al.*, 2009), further increases the number of epistasis tests.

Efficient software is needed to carry out the large number of tests of interaction using quantitative responses. Although several software programs have been proposed to search for interactions in case–control data (Greene *et al.*, 2010; Zhang *et al.*, 2009), few have been optimized to handle continuous responses. In this article, we describe FastEpistasis, an optimized software suite designed for quantitative responses, which extends PLINK (Purcell *et al.*, 2007) epistasis functionality. FastEpistasis uses a parallel algorithm that is capable of computing tests for all pairs of genome-wide SNPs and efficiently handles tests given multiple phenotypes.

2 METHODS

FastEpistasis, a software tool capable of computing tests of epistasis for a large number of SNP pairs, is an efficient parallel extension to the PLINK epistasis module. It tests epistatic effects in the normal linear regression of a quantitative response on marginal effects of each SNP and an interaction effect of the SNP pair, where SNPs are coded as additive effects, taking values 0, 1 or 2. The test for epistasis reduces to testing whether the interaction term is significantly different from zero.

FastEpistasis methods are briefly outlined, with further details provided in the Supplementary Material. The computations are optimized by splitting the analysis tasks into three separate applications: *pre-*, *core-* and *post-computation*. The pre-computation phase loads PLINK binary format data files, reformats the data for faster computations and reduces the number of conditions to check in the core phase. The core computational phase is designed to embarrassingly parallelize the computations, iterating through SNP pairs and efficiently carrying out the tests for epistasis. The computations are based on applying the QR decomposition to derive least squares estimates of the interaction coefficient and its standard error. The core computation software comes in several versions to take advantage of different high-performance architectures—a Symmetric Multiprocessing (SMP) version and a clustered Message Passing Interface (MPI) version. An optional post-computation phase is provided to aggregate results from each processor or core, include detailed SNP information, compute *P*-values from each test, and convert to text files.

We assessed the performance of our software using International HapMap Project genotypes (Frazer *et al.*, 2007) and random phenotypes (see supplementary material for details). Unless stated otherwise, results from all SNP pair epistasis tests are output.

3 RESULTS

We compared the performance of FastEpistasis and PLINK epistasis tests for several sets of SNP pairs, using a single core to enable

*To whom correspondence should be addressed.

Table 1. Epistasis tests per second completed by FastEpistasis core computation phase for several population sizes, using eight cores

Individuals	10^3 tests (s)	Individuals	10^3 tests (s)
60	1393.14 (82.7)	1000	289.44 (3.7)
100	1214.15 (38.4)	3000	81.00 (0.7)
500	538.59 (3.9)	5000	45.56 (0.4)

Averages are taken over 10 runs with SDs in parentheses. SNP pairs are derived from disjoint sets *A*, *B* containing 19 999 and 2596 SNPs.

a fair comparison. FastEpistasis ran almost 15 times faster than PLINK, completing 81376 epistasis tests per second compared to 5696 tests per second computed by PLINK (see Supplementary Table 1). In the event that only SNP pair results below a *P*-value threshold are needed, requiring a negligible time for post-computation, FastEpistasis computes about 120 000 epistasis tests per second, ∼20 times faster than PLINK (also see below for output size effect in multiple phenotype analysis). However, the gain in performance depends on the number of individuals in the population as shown in Table 1 and Supplementary Figure 1. With the exception of *Not A Number* PLINK output, all FastEpistasis results agree perfectly with PLINK.

The speed of FastEpistasis scales linearly with the number of processors at 93% asymptotical efficiency, using either SMP or MPI architecture (see Supplementary Fig. 2). At this rate, the computational time required to test all pairs of 500000 SNPs, totaling 125 billion tests, using a population of 5000 individuals is about 29, 4 or 0.5 days using 8, 64 or 512 MPI-bound processors.

FastEpistasis is capable of analyzing several different phenotypes simultaneously, using the same genotypes. By performing the QR decomposition of the covariate matrix once and applying the result to several phenotypes, the total number of computations is reduced compared to carrying out the computations separately for each phenotype. Although we observe a significant speed-up with multiple phenotypes, the performance reaches a peak and then collapses, and becomes a penalty as the number of phenotypes grows (Supplementary Fig. 3). The problem occurs during the core-computation phase and is due to the size of the results. The processors are able to compute the test statistics faster than the results can be buffered and transferred to the hard drive. Completely omitting to output the results removes the performance collapse. The reduction in computational time analyzing several phenotypes simultaneously depends on several factors including the speed of the epistasis tests (which in turn depends on the number of individuals) and the number of results to be output. For example, using 8 processors, a population size of 171 MKK individuals, 10 phenotypes, and outputting all epistasis results, the computations are 1.06 times faster than analyzing each phenotype separately whereas outputting results for $P < 0.01$, $\sim 1\%$ of tests, the computations are 4.77 times faster. Therefore, restricting the output to *P*-values below a relatively small threshold, or increasing storage throughput using a striped disk RAID array, for example, can decrease computational demands when analyzing multiple phenotypes.

4 DISCUSSION

Epistasis is fundamental to understanding the structure and function of genetic pathways (Phillips, 2008). Recent studies have reported epistatic effects that confer susceptibility to common diseases (Emily *et al.*, 2009; Wu *et al.*, 2010). Genetic interactions may also be able to explain a larger proportion of phenotypic variance for common diseases or related traits (Manolio *et al.*, 2009) or reveal information about gene function (Franke *et al.*, 2009). FastEpistasis is capable of computing fast tests of epistasis for quantitative phenotypes, enabling researchers to study interaction effects of pairs of genetic loci.

ACKNOWLEDGEMENTS

The authors would like to thank the Vital-IT team as all calculations were performed on the Vital-IT high-performance computing facility of the Swiss Institute of Bioinformatics. We thank Frédéric Schütz and Toby Johnson for helpful statistics advice.

Funding: Swiss Institute of Bioinformatics service grant; Swiss National Science Foundation grant #3100AO-116323/1 (to S.B.).

Conflict of Interest: none declared.

REFERENCES

Cordell,H.J. (2009) Detecting gene-gene interactions that underlie human diseases. *Nat. Rev. Genet.*, **10**, 392–404.

Curtis,D. (2007) Allelic association studies of genome wide association data can reveal errors in marker position assignments. *BMC Genet.*, **8**, 30.

Emily,M. *et al.* (2009) Using biological networks to search for interacting loci in genome-wide association studies. *Eur. J. Hum. Genet*., **17**, 1231–40.

Franke,L. *et al.* (2009) eQTL analysis in humans. *Meth. Mol. Biol*., **573**, 311–328.

Frazer,K.A. *et al.* (2007) A second generation human haplotype map of over 3.1 million SNPs. *Nature*, **449**, 851–861.

Gayan,J. *et al.* (2008) A method for detecting epistasis in genome-wide studies using case-control multi-locus association analysis. *BMC Genomics*, **9**, 360.

Greene,C.S. *et al.* (2010) Multifactor dimensionality reduction for graphics processing units enables genome-wide testing of epistasis in sporadic ALS. *Bioinformatics*, **26**, 694–695.

Herold,C. *et al.* (2009) INTERSNP: genome-wide interaction analysis guided by a priori information. *Bioinformatics*, **25**, 3275–3281.

Hirschhorn,J.N. *et al.* (2005) Genome-wide association studies for common diseases and complex traits. *Nat. Rev. Genet.*, **6**, 95–108.

Manolio,T.A. *et al.* (2009) Finding the missing heritability of complex diseases. *Nature*, **461**, 747–753.

Phillips,P.C. (2008) Epistasis–the essential role of gene interactions in the structure and evolution of genetic systems. *Nat. Rev. Genet.*, **9**, 855–867.

Purcell,S. *et al.* (2007) PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, **81**, 559–575.

Wu,J. *et al.* (2010) Screen and clean: a tool for identifying interactions in genome-wide association studies. *Genet. Epidemiol.*, **34**, 275–285.

Zhang,X. *et al.* (2009) FastChi: an efficient algorithm for analyzing gene-gene interactions. *Pac. Symp. Biocomput.*, 528–539.