BIOINFORMATICS ORIGINAL PAPER Vol. 27 no. 11 2011, pages 1462–1465
doi:10.1093/bioinformatics/btr172

Genome analysis Advance Access publication April 6, 2011

EpiGPU: exhaustive pairwise epistasis scans parallelized on consumer level graphics cards

Gibran Hemani[1,*], Athanasios Theocharidis[1], Wenhua Wei[2] and Chris Haley[1,2]

[1]Division of Genetics and Genomics, The Roslin Institute and R(D)SVS, University of Edinburgh, Easter Bush, Midlothian, EH25 9RG and [2]MRC Human Genetics Unit, Institute of Genetics and Molecular Medicine, Western General Hospital, Crewe Road, Edinburgh EH4 2XU, UK

Associate Editor: Alex Bateman

ABSTRACT

Motivation: Hundreds of genome-wide association studies have been performed over the last decade, but as single nucleotide polymorphism (SNP) chip density has increased so has the computational burden to search for epistasis [for *n* SNPs the computational time resource is *O(n(n-1)/2)*]. While the theoretical contribution of epistasis toward phenotypes of medical and economic importance is widely discussed, empirical evidence is conspicuously absent because its analysis is often computationally prohibitive. To facilitate resolution in this field, tools must be made available that can render the search for epistasis universally viable in terms of hardware availability, cost and computational time.

Results: By partitioning the 2D search grid across the multicore architecture of a modern consumer graphics processing unit (GPU), we report a 92× increase in the speed of an exhaustive pairwise epistasis scan for a quantitative phenotype, and we expect the speed to increase as graphics cards continue to improve. To achieve a comparable computational improvement without a graphics card would require a large compute-cluster, an option that is often financially non-viable. The implementation presented uses OpenCL—an open-source library designed to run on any commercially available GPU and on any operating system.

Availability: The software is free, open-source, platform-independent and GPU-vendor independent. It can be downloaded from http://sourceforge.net/projects/epigpu/.

Contact: gib.hemani@roslin.ed.ac.uk

Received on October 29, 2010; revised on February 28, 2011; accepted on March 30, 2011

1 INTRODUCTION

The importance of epistasis (gene–gene interactions) in complex trait analysis is largely unknown, and computational difficulties have rendered this topic difficult to explore. Yet, efforts to identify the genetic factors that underlie traits of economic or medical importance have accelerated over the last decade. Indeed, in the first half of 2010 alone genome-wide association studies (GWASs) for 165 traits were published (http://www.genome.gov/gwastudies/, last accessed date October 12, 2010). Despite the scale of these studies, the proportion of genetic variance explained has been disappointing, and so has emerged the enigma of the missing heritability (Manolio *et al.*, 2009). But even with the availability of these data, the search for epistasis has been largely neglected (Carlborg and Haley, 2004; Phillips, 1998).

Phenotypic variance can be partitioned into several components:

$$V_{\text{phenotypic}} = V_{\text{add}} + V_{\text{dom}} + V_{\text{epistatic}} + V_{\text{env}} + \cdots \quad (1)$$

However, in the context of linkage mapping or GWAS, heritability estimates are generally limited to the narrow-sense, thus reducing the search to only independent additive genetic effects (Visscher *et al.*, 2008). In human and animal genetics in particular, estimation of genetic variance beyond the scope of purely additive effects (i.e. broad-sense heritability) is intractable in most cases, and so the overall contribution of epistasis remains unknown. There has been a lively debate for several years concerning the importance of the broad-sense heritability in complex traits, and in particular the contribution of epistasis (Frankel and Schork, 1996; Hill *et al.*, 2008; Moore, 2005), but there is still an absence of empirical results.

Epistasis is a recurring candidate for explaining the missing heritability, but in fact most epistatic patterns are unlikely to be detectable through marginal effects alone (Evans *et al.*, 2006). On the contrary, if epistasis was found to be prevalent in complex traits the major implication would be that significant genetic control exists beyond the extant estimates of narrow-sense heritability. Should it be the case that the phenotypic effect of one locus depends on the genotype at another locus, the impact upon such endeavours as personalized medicine, disease risk prediction, animal breeding and evolutionary genetics could be significant.

Several obstacles exist that make epistatic searches difficult. When searching for independent additive effects, each SNP is tested for association with the phenotype; but in order to most powerfully identify epistatic effects, the search must be increased to two dimensions (Evans *et al.*, 2006; Marchini *et al.*, 2005), testing each SNP against all other SNPs. For example, a 300k SNP chip would require $300\,000 \times 299\,999/2 \approx 4.5 \times 10^{10}$ independent tests, which is a massive computational undertaking. Currently, the cheapest way to run this type of analysis, using a desktop computer, could take weeks. However, the parallel decomposition of this problem is relatively straightforward, and the mainstream availability of multicore GPUs has paved the way for an efficient and inexpensive alternative. We provide software that dramatically reduces the computational time of an exhaustive search for two-locus epistasis on large-scale SNP data for continuous traits. We evaluate the performance of the software running on several types of GPUs against optimized software that runs serially on desktop computers, and against parallelized versions for multicore CPUs and large compute clusters.

*To whom correspondence should be addressed.

2 METHODS

2.1 Statistical tests

The program performs an exhaustive scan for pairwise interactions, such that each SNP is tested against all other SNPs for statistical association with the phenotype. Two different tests can be performed using the software, either treating the pairwise genotype classes as factor effects or parameterizing the class means to exclude any marginal effects thus testing for only interaction terms.

There are nine possible genotypes resulting from combining a pair of SNPs. By treating the genotype classes as a fixed effect, an 8 d.f. F-test can be performed that tests the following hypotheses:

$$H_0 : \sum_{i=1}^{3}\sum_{j=1}^{3}(\bar{x}_{ij}-\mu)^2 = 0; \quad (2)$$

$$H_1 : \sum_{i=1}^{3}\sum_{j=1}^{3}(\bar{x}_{ij}-\mu)^2 > 0; \quad (3)$$

where μ is the phenotype mean and $\bar{x}_{ij}$ is the pairwise genotype class mean for genotype i at locus A and genotype j at locus B. This type of statistical test does not parameterize for specific types of epistasis, rather it tests for the joint genetic effect at two loci, this having been demonstrated to be statistically more efficient when searching for a wide range of epistatic patterns (Evans *et al.*, 2006; Millstein *et al.*, 2006).

The software is, however, capable of reducing the test to 4 d.f., parameterizing for interaction terms only (Cordell, 2002). This is achieved by removing the marginal additive and dominance effects from each locus, testing the following hypotheses:

$$H_0 : \sum_{i=1}^{3}\sum_{j=1}^{3}(\bar{x}_{ij}-\bar{x}_i-\bar{x}_j+\mu)^2 = 0; \quad (4)$$

$$H_1 : \sum_{i=1}^{3}\sum_{j=1}^{3}(\bar{x}_{ij}-\bar{x}_i-\bar{x}_j+\mu)^2 > 0; \quad (5)$$

where $\bar{x}_i$ ($\bar{x}_j$) is the marginal class mean for genotype i (j) at locus A (B).

For efficiency, the program does not fit additional factors or covariates during analysis, instead it requires the normalized residual from the phenotype adjusted for other parameters to be used as the response variable, as in the GRAMMAR method for example (Aulchenko *et al.*, 2007). As in other implementations for specific epistatic parameterizations (Schüpbach *et al.*, 2010), the interaction parameterization assumes independence between SNPs for computational efficiency. Missing genotype values are ignored in the analysis, such that the denominator degrees of freedom is representative of the number of observed genotypes. The program is also capable of performing permutations, a potentially important function for generating thresholds (Churchill and Doerge, 1994).

2.2 OpenCL and general purpose graphics processing

Because the analysis problem can be managed with the single instruction multiple data (SIMD) model, the massively multicore architecture of consumer level graphics cards offers a viable option for searching for epistasis. While other architectures exist that may also be potentially viable, such as Cell or FPGA, we have focused on GPUs because of their commercial accessibility and increasing availability on HPC clusters.

Over the last few years, GPU devices have become programmable for non-graphics oriented applications through the CUDA API, but this has been restricted to Nvidia hardware only. OpenCL is a more recent API that is designed to be vendor independent, thus allowing software to run on any modern graphics card, including ATI/AMD (http://www.khronos.org/opencl/). We have opted to use OpenCL for its cross-vendor capability.

While most graphics cards have hundreds of cores, performance does not necessarily scale proportionally, and not all algorithms will benefit from GPU parallelization (e.g. Davis *et al.*, 2011). The main performance limiting factor is the I/O bandwidth between processing cores and video memory. The computational kernel that runs on the GPU cores restructures the regression algorithm, making efficient use of the video memory hierarchies and this was necessary for achieving significant speed improvements. Several steps were necessary to limit the kernel I/O operations, including converting naive sum of squares algorithms to on-line algorithms (Welford, 1962), storing frequently accessed genotype class means in local memory, vectorizing phenotype reads and delivery of genotypes to the kernel in bitpacked form.

The program uses a command line interface, and allows the analysis to be stopped and resumed by the user. The scale of the search space is such that storing all results would be impossible. Instead, only results that exceed a user-defined threshold are saved.

2.3 Performance testing

Exhaustive pairwise scans for epistasis were performed on a dataset comprising 300 000 SNPs, evenly spaced across 20 chromosomes, with 1% missing values and 1000 individuals, for association with a random normally distributed phenotype. Analyses were performed using the full 8 d.f. test, although the speed with the 4 d.f. test is almost identical. Other parameters, such as the 'iteration size' were chosen as recommended in the software documentation. All timings reported refer to user time.

epiGPU is adapted from a CPU version that performs the same analysis, *episcan* (Hemani,G. and Wei,W. 2010, unpublished data, software). As with *epiGPU*, the CPU version is written in C, and compiled using GCC version 4.3.4 with -O2 optimization. To test the efficiency of *episcan*, we compared its performance against the FastEpistasis module in PLINK (Schüpbach *et al.*, 2010). On an Intel i7 970 3.2 GHz processor (using a single core), *episcan* performs ~128 201 tests per second, which is over 3.5x faster than *FastEpistasis* (~36 180 tests per second as reported in Schüpbach *et al.*, 2010).

In addition to running serially, *episcan* can also parallelize across multicore CPUs. It uses the OpenMP API, and when hyperthreading is enabled it achieves near linear speed improvements with the number of processing cores. For multinode compute clusters, an extended version of the software called *epiMPI* was used (Hemani,G. 2010, unpublished data, software). It geometrically parallelizes the search space using the OpenMPI API (Gabriel *et al.*, 2004), also achieving almost linear scaling with the number of nodes.

All CPU implementations use single precision floating point operations. When compared with double precision the speed was identical, as were F value calculations to five decimal places. The GPU version also uses single precision, and maintained precision to four decimal places.

Open source code is available for both CPU implementations at https://sourceforge.net/projects/epigpu/files/misc/.

3 RESULTS

We produced software that geometrically parallelizes exhaustive searches for pairwise epistatic associations with quantitative traits. We performed large-scale analyses, typical of those that would be expected based on GWASs already published, on several different software and hardware systems. Our tests show that against the baseline system (serial code running on a modern CPU) graphics cards can perform the same analysis almost two orders of magnitude faster and at minimal expense (Table 1), such that an analysis that would take over 4 days could be performed in just over an hour by using software utilizing a graphics card. It is demonstrable that to achieve comparable speeds using CPU cores would require a large compute cluster, for which the cost to acquire and administer could be prohibitively expensive.

Table 1. Performance and cost comparison

Parallelization	Hardware	Cost / £[a]	Time / min[b]	Relative speed[c]	Cost benefit[d]
None	Baseline CPU[e]	–	5860	1	–
Multicore CPU	6-core CPU[f]	760	986	5.9	1.6
	8-core CPU[g]	1600	763	7.7	1.0
CPU cluster[g]	16-core cluster	–	398	14.7	–
	32-core cluster	–	195	30.0	–
	64-core cluster	–	96	61.0	–
GPU	Nv Fermi GTX580	367	63	91.6	51.9
	ATI Radeon 6970	300	86	68.1	47.2
	Nv Tesla S1070	960	146	40.1	9.0
	Nv GTX285	230	145	40.1	36.2
	Nv 8800GT	72	613	9.6	27.7

[a]Approximate cost for equipment above baseline. Cost estimates for large compute clusters are too subjective for realistic comparisons; [b]Total user time to complete the analysis (300 000 SNPs, 1000 individuals); [c]Time relative to baseline time; [d]Cost benefit calculated as speed/cost, figures shown are adjusted relative to the cost of the best-performing desktop CPU alternative (8-cores); [e]Baseline equipment, Intel i7 970 3.2 GHz, running in serial; [f]Intel i7 970 3.2 GHz; [g]Dual Intel Xeon E5472 3.0 GHz.

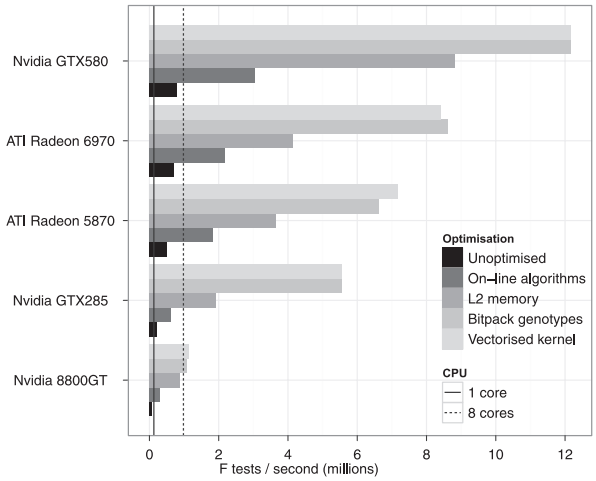

Fig. 1. Incremental improvements in performance by incorporating different GPU optimization methods. For reference, the CPU speeds are shown as vertical lines (serial and parallelized on Dual socket Intel Xeon E5472). Speeds are for calculating 8 d.f. *F*-tests with 1000 individuals.

The use of graphics cards as a tool for scientific research is a rapidly emerging industry that has manifested staggering improvements in performance over the last few years. However it is still in its infancy, and as reflected in Figure 1, the level of manual optimization required by developers to harness this power is considerable. Furthermore, while a very heterogeneous array of devices can be used for OpenCL applications, differences in their architectures inevitably results in different responses to optimization strategies. Figure 1 shows that without careful optimization, even the most recent GPUs will appear to offer little to no advantage over CPU implementations.

4 CONCLUSION

Quantitative genetics has long been occupied with the theoretical contribution of genetic variants to complex traits. The last decade has seen a global effort to start investigating this empirically on a large scale, yet epistasis remains largely unexplored. Computing exhaustive pairwise epistatic scans is an important step in making tractable the understanding of non-additive genetic effects in complex traits. We show that this can be achieved efficiently by using consumer level graphics cards, an established technology that is cheap and widely available. In its current implementation, *epiGPU* is limited to performing linear regression on quantitative traits, but the parallel decomposition framework is sufficiently generic to allow its extension to other pairwise statistical analyses relatively easily, such as chi-square testing for case–control data.

Another central problem with epistasis scans is the heavy multiple testing penalty incurred by stringent significance thresholds. Computationally straightforward methods such as the Bonferroni correction are likely to penalize for an overestimated number of independent tests, and this is particularly problematic with epistasis where the dimensionality of the search is increased. However, with the growing availability of GPU clusters (Fan *et al.*, 2004), it is now becoming feasible to perform 2D genome-wide permutation analyses to generate more accurate estimates of family-wise false discovery rates (Churchill and Doerge, 1994), a potentially critical step toward understanding the contribution of epistasis toward complex traits.

ACKNOWLEDGEMENT

We thank Dr Sara Knott for her comments on the manuscript.

Funding: Biotechnology and Biological Sciences Research Council; Medical Research Council; Biosciences KTN; Newsham Choice Genetics.

Conflict of Interest: none declared.

REFERENCES

Aulchenko,Y.S. *et al.* (2007) Genomewide rapid association using mixed model and regression: a fast and simple method for genomewide pedigree-based quantitative trait loci association analysis. *Genetics*, **177**, 577–585.

Carlborg,O. and Haley,C.S. (2004) Epistasis: too often neglected in complex trait studies? *Nat. Rev. Genet.*, **5**, 618–625.

Churchill,G.A. and Doerge,R.W. (1994) Empirical threshold values for quantitative triat mapping. *Genetics*, **138**, 963–971.

Cordell,H.J. (2002) Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Hum. Mol. Genet.*, **11**, 2463–2468.

Davis,N.A. *et al.* (2011) Real-world comparison of CPU and GPU implementations of SNPrank: a network analysis tool for GWAS. *Bioinformatics*, **27**, 284–285.

Evans,D.M. *et al.* (2006) Two-stage two-locus models in genome-wide association. *PLoS Genet.*, **2**, e157.

Fan,Z. *et al.* (2004) GPU cluster for high performance computing. In *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*, SC '04. IEEE Computer Society, Washington, DC, USA, pp. 47–59.

Frankel,W.N. and Schork,N.J. (1996) Who's afraid of epistasis? *Nat. Genet.*, **14**, 371–373.

Gabriel,E. *et al.* (2004) Open MPI: goals, concept, and design of a next generation MPI implementation. In *11th European PVM/MPI Users' Group Meeting, Budapest, Hungary, Proceedings*. Vol. 3241 of *Lecture Notes in Computer Science*, Springer, pp. 97–104.

Hill,W.G. *et al.* (2008) Data and theory point to mainly additive genetic variance for complex traits. *PLoS Genet.*, **4**, e1000008.

Manolio,T.A. *et al.* (2009) Finding the missing heritability of complex diseases. *Nature*, **461**, 747–753.
Marchini,J. *et al.* (2005) Genome-wide strategies for detecting multiple loci that influence complex diseases. *Nat. Genet.*, **37**, 413–417.
Millstein,J. *et al.* (2006) A testing framework for identifying susceptibility genes in the presence of epistasis. *Am. J. Hum. Genet.*, **78**, 15–27.
Moore,J.H. (2005) A global view of epistasis. *Nat. Genet.*, **37**, 13–14.
Phillips,P.C. (1998) The language of gene interaction. *Genetics*, **149**, 1167–1171.
Schüpbach,T. *et al.* (2010) FastEpistasis: a high performance computing solution for quantitative trait epistasis. *Bioinformatics*, **26**, 1468–1469.
Visscher,P.M. *et al.* (2008) Heritability in the genomics era–concepts and misconceptions. *Nat. Rev. Genet.*, **9**, 255–266.
Welford,B. (1962) Note on a method for calculating corrected sums of squares and products. *Technometrics*, **4**, 419–420.