

Fourth-Order Exhaustive Epistasis Detection for the xPU Era

Ricardo Nobre¹, Aleksandar Ilic¹, Sergio Santander-Jiménez² and Leonel Sousa¹

¹ INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

{ricardo.nobre,aleksandar.ilic,leonel.sousa}@inesc-id.pt

² University of Extremadura (UNEX), Cáceres, Spain

sesaji@unex.es



Association Studies Identify Correlation w/ Genotype

cases
have trait

controls
do not have trait



markers (e.g. SNPs)
most correlated w/ trait

*output used in prevention / treatment,
reducing spread of infectious diseases*

Single Nucleotide Polymorphism

variation of single nucleotide in
genome of $\geq 1\%$ of a population

sample X	...	A	C	C	A	G	A	T	...
		T	G	G	T	C	T	A	
	...								
sample Y	...	G	T	A	G	T	A	G	...
		C	A	T	C	A	T	C	

Epistasis Detection as a Combinatorial Problem

Some phenotypes depend on high-order interactions^{1,2}

Total num comb. of SNPs:

$$\frac{M!}{k! (M - k)!}$$

Embarrassingly parallel workload: repeat 3 steps

1 Count genotype freq.

2 Compute scores

3 Reduce scores

2 alleles per SNP

major allele (X)
highest frequency

minor allele (x)
lowest frequency



3 Possible Genotypes

Homozygous major: XX (0)

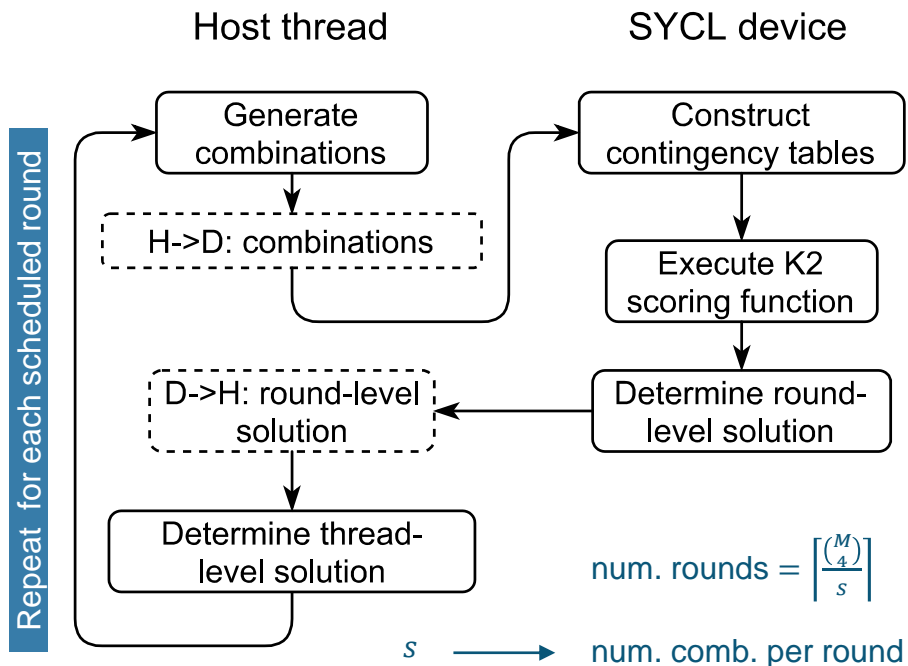
Heterozygous: Xx (1)

Homozygous minor: xx (2)

[1] J. Sun, Hidden risk genes with high-order intragenic epistasis in Alzheimer's disease. J Alzheimers Dis, 2014

[2] C. Im, Genome-wide search for higher order epistasis as modifiers of treatment effects on bone mineral density in childhood cancer survivors. Eur J Hum Genet, 2018

Suitable for Heterogeneous Parallel Architectures



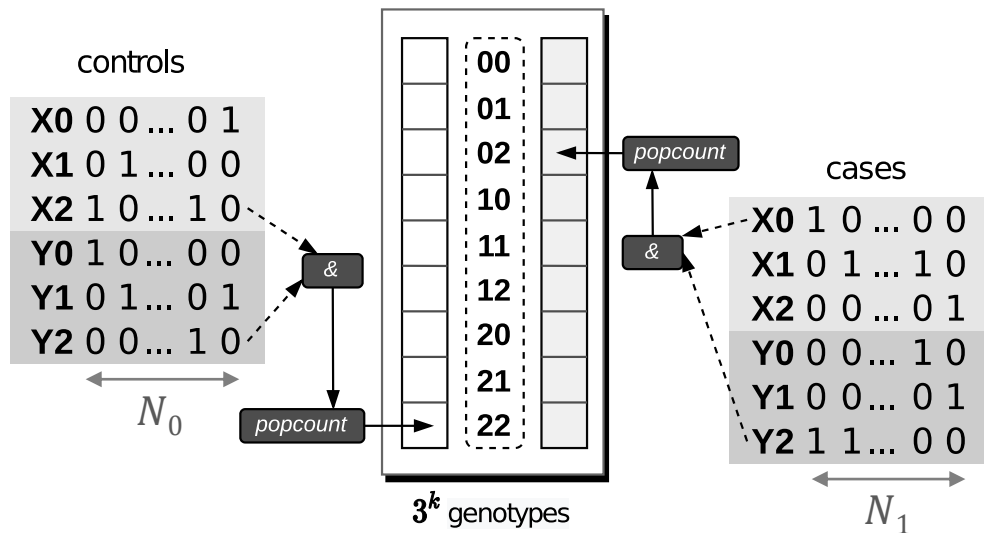
SoA for accelerators focuses on 2/3-way searches and is usually tied to proprietary technologies

Proposed solution:

- Implem. in SYCL for HW interop.
- Specialized to fourth-order explor. with support for GPU accelerators

CPU (Host) + GPU (Device) is a suitable pairing for achieving high epistasis detection throughput

Contingency Table Construction (Example for 2-way)



Bitwise AND and POPC (count num. of set bits) instr. in CPUs¹ and GPUs²

POPC in SYCL through `sycl::popcount()`

Multiple samples processed per POPC instruction

$3 \times (N_0 + N_1)$ bits per SNP to represent information for all cases and controls

Num. POPC inst. $\rightarrow \binom{M}{4} \times 81 \times \left(\left\lfloor \frac{N_0}{32} \right\rfloor + \left\lfloor \frac{N_1}{32} \right\rfloor \right)$

In 4-way Num. of bits per instr.

[1] X. Wan, BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. AJHG, 2010

[2] C. Ponte-Fernández, Fast search of third-order epistatic interactions on CPU and GPU clusters. IJHPCA, 2020

Optimizing Calculations in Fourth-Order Searches

Knowledge specific to input data can be used to reduce amount of instr. exec.

Remaining frequencies derived using simple arithmetic operations from:

- 16 genotypes (w/ SNPs of type 0 or 1)
- Precalculated freq. for third-order comb.

Close to $5 \times$ less inst. executed (from $\frac{81}{16}$) when processing challenging datasets

32 genotypes (4 x 8)

CALC_A_B_C(a,b,c, A_idx, B_idx, C_idx)
 CALC_A_B_D(a,b,d, A_idx, B_idx, D_idx)
 CALC_A_C_D(a,c,d, A_idx, C_idx, D_idx)
 CALC_B_C_D(b,c,d, B_idx, C_idx, D_idx)



24 genotypes (6 x 4)

CALC_A_B_C(a,b,2, A_idx, B_idx, C_idx)
 CALC_A_B_C(a,2,c, A_idx, B_idx, C_idx)
 CALC_A_B_C(2,b,c, A_idx, B_idx, C_idx)
 CALC_A_B_D(a,2,d, A_idx, B_idx, D_idx)
 CALC_A_B_D(2,b,d, A_idx, B_idx, D_idx)
 CALC_A_C_D(2,c,d, A_idx, C_idx, D_idx)



8 genotypes (4 x 2)

CALC_A_B_C(a,2,2, A_idx, B_idx, C_idx)
 CALC_A_B_C(2,b,2, A_idx, B_idx, C_idx)
 CALC_A_B_C(2,2,c, A_idx, B_idx, C_idx)
 CALC_B_C_D(2,2,d, B_idx, C_idx, D_idx)



1 genotype (1 x 1)

CALC_A_B_C(2,2,2, A_idx, B_idx, C_idx)

Experimental Setup and Performance Metric

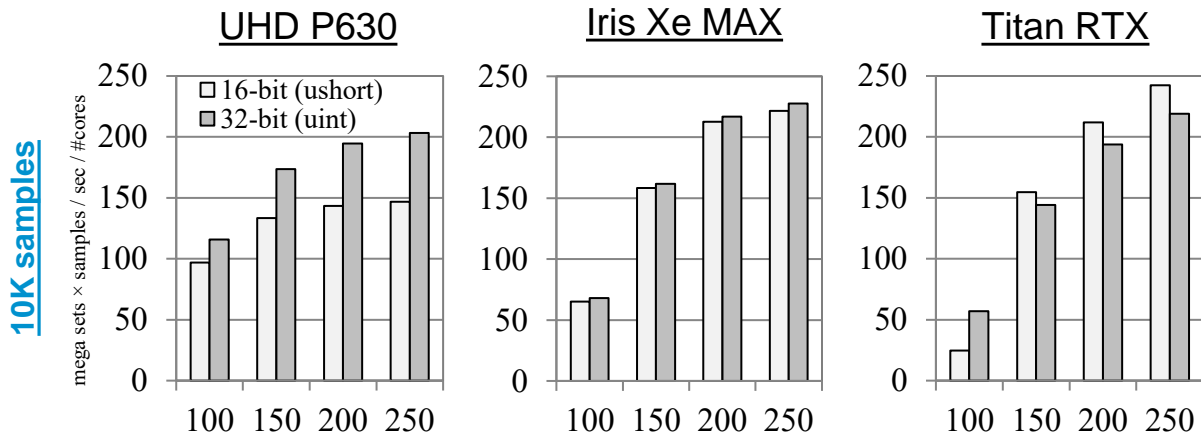
Systems	Intel CPU (arch.) #cores freq.	iGPU / dGPU (arch.) # stream cores (freq.) memory (band.)	DRAM	Operating System
S1	(2×) Xeon Gold 6128 (Skylake) 2×6 3.4–3.7GHz		192GB DDR4	Ubuntu 18.04
S2	Xeon E-2176G (Coffee Lake) 6 3.7–4.7GHz	Intel UHD Graphics P630 (Gen9.5) 192 (1200MHz) system shared	64GB DDR4	Ubuntu 20.04
S3	Core i9-10920X (Cascade Lake) 12 3.5–4.6GHz	Intel Iris Xe MAX (Gen12) 768 (1650MHz) 4GB (68.26 GB/s)	32GB DDR4	Ubuntu 20.04
S4	Core i9-10980XE (Cascade Lake) 18 3.0–4.6GHz	NVIDIA Titan RTX (Turing) 4608 (1770MHz) 24GB (672.0 GB/s)	128GB DDR4	CentOS 7.8

SYCL source code compiled with the Intel OneAPI DPC++ compiler

Runs performed with synthetic datasets with half cases/controls

Performance metric is the number of combinations of SNPs processed per second (scaled to sample size)

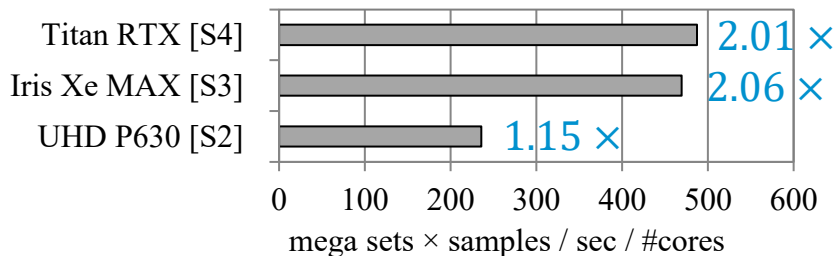
Experimental Results w/ GPU Acceleration



Best param. depends on the targeted GPU:

- 16-bit improved perf. on Titan RTX
- Perf. drop on UHD P630 / Iris Xe MAX

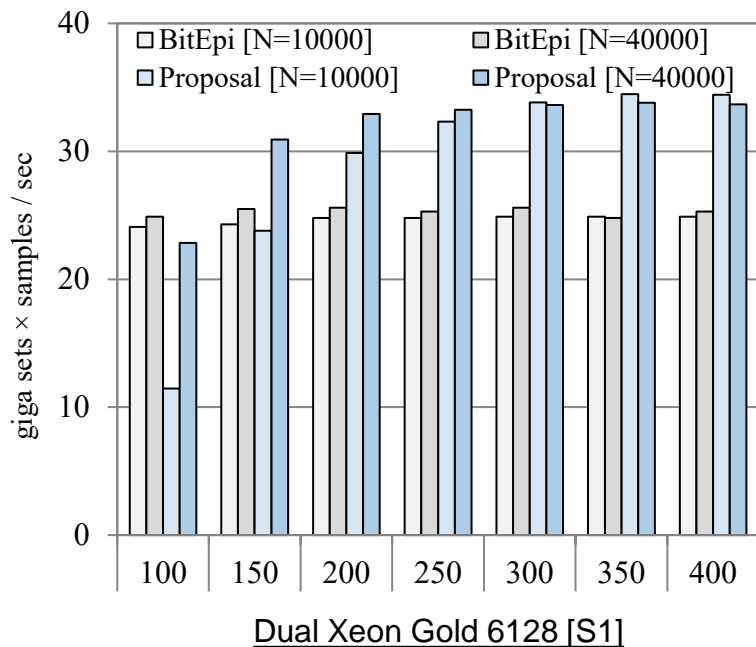
250 SNPs x 80K samples



Similar perf. per stream core on Gen12 and Turing GPU architectures

Up to 2245 giga ($\times 10^9$) sets of SNPs per second on system S4 (Titan RTX)

Performance Using CPU as the SYCL Device



Overall higher perf. than BitEpi¹
 $1.33 \times$ (Xeon Gold) and $1.13 \times$ (Core i9)

BitEpi uses a data repres. and calculation method specialized for fast execution on modern CPUs

Higher performance on CPUs while achieving efficient execution on GPU devices

[1] A. Bayat, BitEpi: A Fast and Accurate Exhaustive Higher-Order Epistasis Search. bioRxiv, 2020

Conclusions

(evaluated on 4 systems)

**Implem. in SYCL for
high HW interoperability**



(1.33 × faster on Xeon Gold server)

**Higher perf. on CPU
than related art**



(65 × speedup w/ GPU)

**Use of GPUs provides
significant perf. increase**

Ongoing work



**reduce memory requirements
of the proposed algorithm**



**add support for heterogeneous
clusters w/ additional architectures
(e.g. custom HW on FPGAs)**



DEFINING TECHNOLOGY

Thank you!

FCT Fundação
para a Ciência
e a Tecnologia



EUROPEAN UNION
European Regional Development Fund

This work was supported by the FCT (Fundação para a Ciência e a Tecnologia, Portugal), ERDF (European Regional Development Fund, EU) and EuroHPC Joint Undertaking through the projects UIDB/50021/2020, LISBOA-01-0145-FEDER-031901 (PTDC/CCI-COM/31901/2017, HiPERBio) and grant agreement No 956213 (SparCity). We would also like to thank Intel Corporation for DevCloud and Iris Xe MAX GPU early access.

Related work by the same authors:

[IPDPS 2020] R. Nobre, A. Ilic, S. Santander-Jiménez, and L. Sousa. Exploring the Binary Precision Capabilities of Tensor Cores for Epistasis Detection.

[JSSPP 2020] R. Nobre, S. Santander-Jiménez, L. Sousa, and A. Ilic. Accelerating 3-way Epistasis Detection with CPU+GPU processing.

[TPDS 2021] R. Nobre, A. Ilic, S. Santander-Jiménez, and L. Sousa. Retargeting Tensor Accelerators for Epistasis Detection.