High Performance Linkage Disequilibrium: FPGAs Hold the Key

Nikolaos Alachiotis and Gabriel Weisz

Department of Electrical & Computer Engineering
Carnegie Mellon University
Outline

- Motivation
- “Generic” algorithm for Linkage Disequilibrium (LD)
- Reconfigurable accelerator architecture and hardware generation
- Design space exploration and performance comparison
- Conclusion
Population genetics investigate...

... the adaptation of species in an environment by studying the genetic composition of a population (same species).

A finch collection

The shape of the beak is controlled by genes.
What is Linkage Disequilibrium (LD)

- LD is the non-random association between alleles (different forms of a gene) at different locations in a genome.
What is Linkage Disequilibrium (LD)

LD is the non-random association between alleles (different forms of a gene) at different locations in a genome.
What is Linkage Disequilibrium (LD)

- LD is the non-random association between alleles (different forms of a gene) at different locations in a genome.

DNA sequencing

Multiple Sequence Alignment (MSA)

Outgroup
ACCCGCACT
ACCCGCACT
ACCTCAACCT
ACCCAACCT
CCTCAACCT

Ingroup
ACCCGCACT
ACCCGCACT
ACCTCAACCT
ACCCAACCT
CCTCAACCT

Population
What is Linkage Disequilibrium (LD)

- LD is the non-random association between alleles (different forms of a gene) at different locations in a genome.
### Pairwise LD computation

<table>
<thead>
<tr>
<th>SNP A</th>
<th>Vec A</th>
<th>Vec B</th>
<th>SNP B</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
<td>1</td>
<td>C</td>
</tr>
</tbody>
</table>

**Alleles:** C and T
- **Ancestral:** C *(outgroup)*
- **Derived:** T

**Alleles:** C and A
- **Ancestral:** A
- **Derived:** C
Pairwise LD computation

Derived allele frequencies

\[
p_1 = \frac{\text{popcnt}(A)}{N}
\]

\[
q_1 = \frac{\text{popcnt}(B)}{N}
\]

Haplotype frequency

\[
x_{11} = \frac{\text{popcnt}(A \cap B)}{N}
\]

\(N\): number of genomes

<table>
<thead>
<tr>
<th>SNP A</th>
<th>Vec A</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
</tr>
</tbody>
</table>

Alleles: C and T

Ancestral: C (outgroup)  →  0

Derived: T  →  1

<table>
<thead>
<tr>
<th>Vec B</th>
<th>SNP B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
</tr>
</tbody>
</table>

Alleles: C and A

Ancestral: A  ←  0

Derived: C  ←  1
Pairwise LD computation

Derived allele frequencies

\[ p_1 = \text{popcnt}(A) \div N \]
\[ q_1 = \text{popcnt}(B) \div N \]

Haplotype frequency

\[ x_{11} = \text{popcnt}(A \cap B) \div N \]

N: number of genomes

Squared Pearson correlation coefficient

\[ r^2_{AB} = \frac{(x_{11} - p_1 q_1)^2}{p_1 q_1 (1 - p_1) (1 - q_1)} \]

Alleles: C and T

Ancestral: C (outgroup) \[ \rightarrow 0 \]
Derived: T \[ \rightarrow 1 \]

Alleles: C and A

Ancestral: A \[ \leftarrow 0 \]
Derived: C \[ \leftarrow 1 \]

SNP A Vec A

<table>
<thead>
<tr>
<th>SNP A</th>
<th>Vec A</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
</tr>
</tbody>
</table>

Vec B SNP B

<table>
<thead>
<tr>
<th>Vec B</th>
<th>SNP B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
</tr>
</tbody>
</table>

Vec A

<table>
<thead>
<tr>
<th>Vec A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Vec B

<table>
<thead>
<tr>
<th>Vec B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
What is Linkage Disequilibrium (LD)

- LD is the non-random association between alleles (different forms of a gene) at different locations in a genome.

DNA sequencing → Multiple Sequence Alignment (MSA)

Outgroup: A C C C G C A C T
Ingroup: A C C C G C A G T
A C T C A C C C T
A C C C A C C C T
C C C A A C C C T

Population

Single Nucleotide Polymorphism (SNP)
What is Linkage Disequilibrium (LD)

- LD is the non-random association between alleles (different forms of a gene) at different locations in a genome.

LD scores based on Pearson's coefficient of correlation
Applications of LD

- Identification of traces of positive selection

GENETICS
Linkage Disequilibrium as a Signature of Selective Sweeps
Yuseob Kim, Rasmus Nielsen
GENETICS July 27, 2004 vol. 167 no. 3 1513-1524; DOI: 10.1534/genetics.103.025387

Letters to Nature
Nature 411, 199-204 (10 May 2001) | doi:10.1038/35075590; Received 11 December 2000; Accepted 13 March 2001

Linkage disequilibrium in the human genome
David E. Reich1, Michele Cargill1,2, Stacey Bolk1, James Ireland1, Pardis C. Sabeti3, Daniel J. Richter1, Thomas Lavery1, Rose Kouyoumjian1, Shelli F. Farhadian1, Ryk Ward3 & Eric S. Lander1,4
So what are the computational issues with LD?
SNPs increase (almost) proportionally with the genomes

- Computational demands increase linearly with the number of genomes
- ... but quadratically with the number of SNPs
In need for high performance
Moore's law

From single-core to multi-core processors

1970 – 2005

2005 – ...

big core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core

little core
Moore's law

From scalar to vector processing

add r3 r1 r2

add.vv v3 v1 v2
Derived allele frequencies

\[ p_1 = \frac{\text{popcnt}(A)}{N} \]
\[ q_1 = \frac{\text{popcnt}(B)}{N} \]

Haplotype frequency

\[ x_{11} = \frac{\text{popcnt}(A \cap B)}{N} \]

Squared Pearson correlation coefficient

\[ r^2_{AB} = \frac{(x_{11} - p_1 q_1)^2}{p_1 q_1 (1 - p_1)(1 - q_1)} \]
Derived allele frequencies

\[ p_1 = \frac{\text{popcnt}(A)}{N} \]
\[ q_1 = \frac{\text{popcnt}(B)}{N} \]

Haplotype frequency

\[ x_{11} = \frac{\text{popcnt}(A \cap B)}{N} \]

Squared Pearson correlation coefficient

\[ r_{AB}^2 = \frac{(x_{11} - p_1 q_1)^2}{p_1 q_1 (1 - p_1)(1 - q_1)} \]

Hardware support (intrinsic instruction) for population count in processors but only on regular registers.
“Generic” LD approach
“Generic” LD approach

Multiple Sequence Alignment
“Generic” LD approach

Multiple Sequence Alignment
"Generic" LD approach

Multiple Sequence Alignment

alignment columns

SNPs

SNP groups
(fixed number of SNPs)
"Generic" LD approach

Multiple Sequence Alignment

alignment columns

SNPs

SNP groups (fixed number of SNPs)
"Generic" LD approach

Multiple Sequence Alignment

- Alignment columns
- Genomes
  - Region A
  - SNP groups
    - SNP group List A
  - SNP groups
    - SNP group List B
- Genomes
  - Region B
  - SNP groups (fixed number of SNPs)
“Generic” LD approach

Multiple Sequence Alignment

alignment columns

genomes

Region A

SNPs

genomes

Region B

SNP groups (fixed number of SNPs)

SNP-group List A

SNP-group List B

Compute group

All pairwise LD computations between two SNP groups
“Generic” LD approach

- No excessive memory requirements and computations when long range LD computations are conducted.

- Offloading to accelerators (FPGAs, GPUs) regardless of available memory capacity on the accelerator platform.

- Better scalability than other parallel algorithms for increasing number of cores in a processor, due to better computation-to-synchronization ratios.
System overview
System overview

Host Processor

Data Processor

LD Processor

DRAM Memory

Accelerator platform
LD processor
LD processor

LD Core Grid
Allele Frequency Calculator
Data Memory

SNP group A data bus
SNP group B data bus
LD score data bus

LD scores to Data processor
LD processor

SNP group B data from Data processor

AFC 0

AFC 1

SNP group A data bus

SNP group B data bus

LD Core Grid

Allele Frequency Calculator

Data Memory

LD Core Grid

LD scores to Data processor

LD Core Grid

LD Core 00
LD Core 01
LD Core 02
LD Core 03

LD Core 10
LD Core 11
LD Core 12
LD Core 13

LD Core 20
LD Core 21
LD Core 22
LD Core 23

LD Core 30
LD Core 31
LD Core 32
LD Core 33

AFC 0
AFC 1
AFC 2
AFC 3

Bank 0
Bank 1
Bank 2
Bank 3

Dual-port Multi-bank Data Memory for SNP group A

LD score data bus

Output Control

LD scores to Data processor
LD processor

SNP group B data from Data processor

AFC 0
AFC 1

AFC arrays for both SNP groups

LD Core Grid

Allele Frequency Calculator

Data Memory

SNP group A data bus
SNP group B data bus
LD score data bus

LD scores to Data processor
Parallel retrieval of scores from multiple LD Cores

LD processor

SNP group B data from Data processor

AFC 0

AFC 1

LD Core Grid

Allele Frequency Calculator

Data Memory

SNP group A data bus

SNP group B data bus

LD score data bus

LD scores to Data processor
LD processor: Type S (Size) parameters

SNP group B data from Data processor

LD Core 00
LD Core 01
LD Core 10
LD Core 11
LD Core 20
LD Core 21
LD Core 30
LD Core 31

AFC 0
AFC 1
AFC 2
AFC 3

Pop count
Accum

GRID WIDTH
GRID HEIGHT
LATENCY
WIDTH
DEPTH
GRID WIDTH

LD scores to Data processor

Output Control
Design space exploration
Design space exploration: Tuning S parameters

Virtex 7 VX980T-2 – post Place And Route results
Throughput (million LDs/second)
10,000 SNPs – 2,504 genomes

LD processor evaluation with constant popcount size
(m=64, t=6)

LD Processor Evaluation for constant LD Core Grid size
(x=8 and y=1)
Design space exploration: Tuning R parameters

Virtex 7 VX980T-2 – post Place And Route results
Throughput (million LDs/second)
10,000 SNPs – 2,504 genomes

LD Processor Evaluation for various Type R configurations
(x=8, y=1, m=512, t=9)
Design space exploration: Refining S parameters

Virtex 7 VX980T-2 – post Place And Route results
Throughput (million LDs/second)
10,000 SNPs – 2,504 genomes

LD Processor Evaluation for various Type R configurations
(x=8, y=1, m=512, t=9)

LD processor evaluation for Configuration #5
(increasing number of LD cores)
Performance comparison
Performance comparison

Platform: Intel Xeon E5-2630 6-core processor at 2.60 GHz and 32 GBs main memory

Data: 10,000 SNPs – 2,504 human genomes (real, 1000Genomes project)
       10,000 SNPs – 10,000 sequences (synthetic)
       10,000 SNPs – 100,000 sequences (synthetic)
## Performance Comparison

### 2,504 Human Genomes (1000Genomes Project), 10,000 SNPs

<table>
<thead>
<tr>
<th>Threads</th>
<th>PLINK 1.9 Exec. time (sec)</th>
<th>mLD/sec</th>
<th>FPGA LD Proc. Speedup (X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.3</td>
<td>4.1</td>
<td>200.2</td>
</tr>
<tr>
<td>2</td>
<td>9.6</td>
<td>5.2</td>
<td>157.8</td>
</tr>
<tr>
<td>4</td>
<td>5.9</td>
<td>8.4</td>
<td>97.7</td>
</tr>
<tr>
<td>8</td>
<td>3.8</td>
<td>13.0</td>
<td>63.1</td>
</tr>
<tr>
<td>12</td>
<td>3.0</td>
<td>16.4</td>
<td>50.1</td>
</tr>
</tbody>
</table>

- **FPGA** 820.9 mLD/sec

### D.1: 10,000 Sequences, D.2: 100,000 Sequences, 10,000 SNPs

<table>
<thead>
<tr>
<th>Threads</th>
<th>PLINK 1.9 Exec. time (sec)</th>
<th>mLD/sec</th>
<th>FPGA LD Proc. Speedup (X)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D.1</td>
<td>D.2</td>
<td>D.1</td>
</tr>
<tr>
<td>1</td>
<td>41.1</td>
<td>389.1</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>31.4</td>
<td>297.6</td>
<td>1.6</td>
</tr>
<tr>
<td>4</td>
<td>19.2</td>
<td>180.2</td>
<td>2.6</td>
</tr>
<tr>
<td>8</td>
<td>11.3</td>
<td>109.4</td>
<td>4.4</td>
</tr>
<tr>
<td>12</td>
<td>9.9</td>
<td>88.3</td>
<td>5.0</td>
</tr>
</tbody>
</table>

- **FPGA**
  - D.1: 205.7 mLD/sec
  - D.2: 20.4 mLD/sec
Conclusion
Lack of a vectorized population count operator in processors does not permit the implementation of a high performance microkernel.

FPGAs allow very wide and deep bit-counting pipelines, which is a key for performance as more genomes are sequenced.

Hardware generation software to explore the design space and get high performance for a particular dataset size.
Thank you