EpiScan GPU: A CUDA-Based Tool for Identifying Epistasis

By: Lance Hartman

What is Epistasis?

- Epistasis can be simply defined as interaction between two different genes.
- It can also be used to find SNP (single nucleotide polymorphisms) association: two or more SNPs working together to cause a phenotype.
- There are two main ways that one can determine if epistasis exists between two genes or SNPs:
  - Physical Observation
  - Statistical Association
    - Machine Learning (feature selection/partial dependence)
    - Regressions
    - Time complexity: $O\left(\frac{n(n-1)}{2}\right)$

EpiScan R Package

(1) Partition the data set into a size of 2000 SNPs. Note that this number may increase or decrease depending on the number of individuals studied.
(2) Set up a two-level nested loop to apply the partition-based correlation for all possible SNP pairs for cases and controls separately.
(3) Compute the difference of correlation coefficients between cases and controls after each partition-based autocorrelation or cross-correlation is complete.
(4) Compute the *P*-values of each difference given that the distribution of the differences follows a Gaussian distribution (refer to the Results section).
(5) Retain only SNP pairs that show a *P*-value below a selected threshold.
(6) Repeat steps 3–5 across all partition pairs.
(7) Proceed to stage 2 by performing a logistic regression on the selected pairs.

Kam-Thong T, Czamara D, Tsuda K, Borgwardt K, Lewis CM, Erhardt-Lehmann A, Hemmer B, Rieckmann P, Daake M, Weber F, Wolf C, Ziegler A, Pütz B, Holsboer F, Schölkopf B, Müller-Myhsok B. EPIBLASTER-fast exhaustive two-locus epistasis detection strategy using graphical processing units. Eur J Hum Genet. 2011 Apr;19(4):465-71. doi: 10.1038/ejhg.2010.196. Epub 2010 Dec 8. PMID: 21150885; PMCID: PMC3060319.

What is CUDA?

- CUDA is a set of tools provided by Nvidia that allows one to interface with their GPU through Kernel code.
- CUDA can operate on Windows and Linux, compatible with the CHOP cluster.
- There are CUDA libraries that operate in C, C++, C#, Python, Java, and more.
- EpiScan GPU uses CUDA's C++ toolset.

Pros and Cons of Working on a GPU

Pros

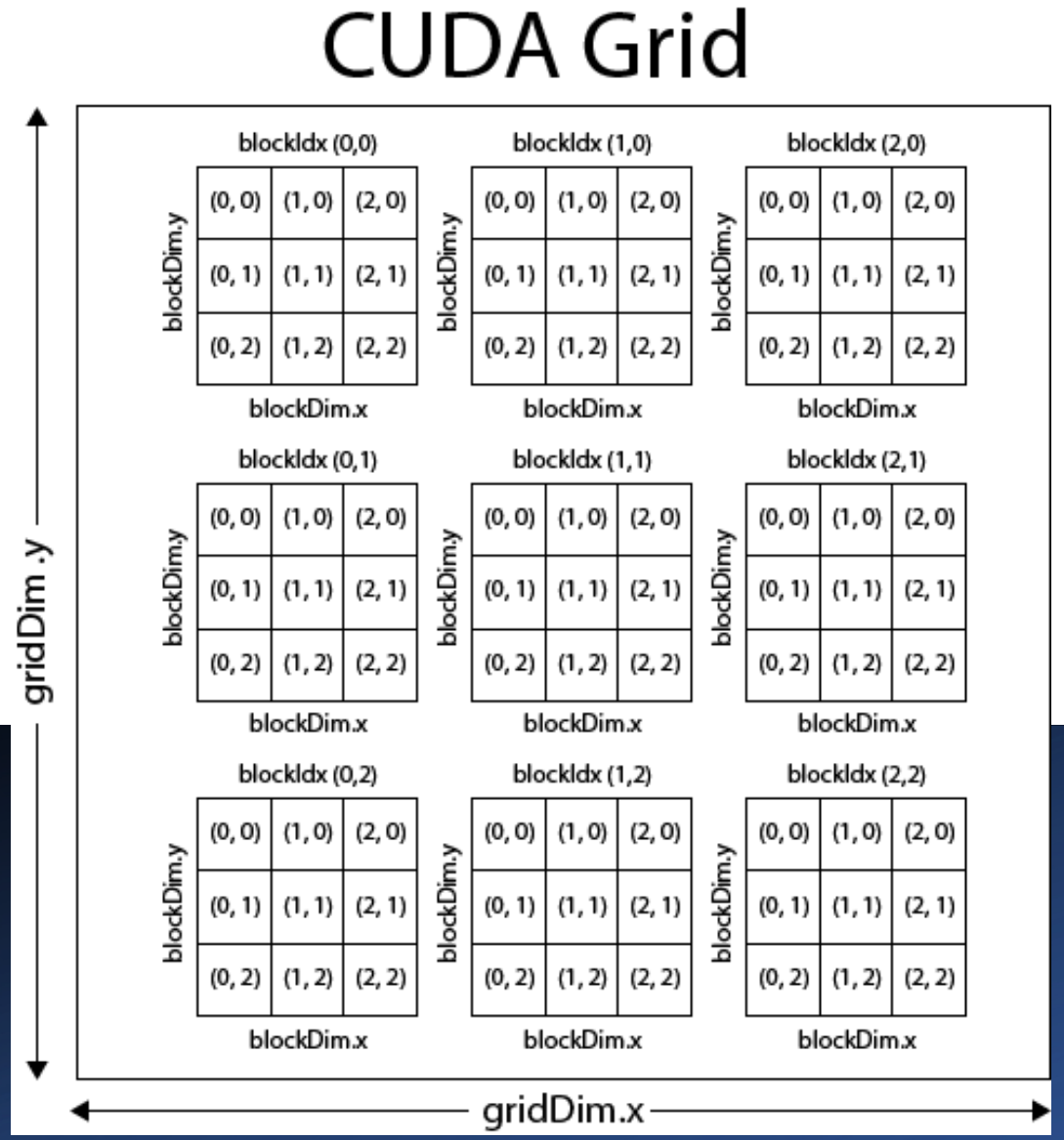
- Designed for throughput – can complete calculations at an extremely fast speed.
- GPU has its own heap memory - keeps user from using up all of their own memory.
- CUDA has recently gained the ability to write to an output stream from ”within” the GPU.

Cons

- Cannot use the standard libraries within CUDA code.
- Dynamic memory allocation is shaky – frequent memory problems.
- Sending lots of data to the GPU is the slowest part – large data analysis is tougher.
- Debugging is extremely difficult.

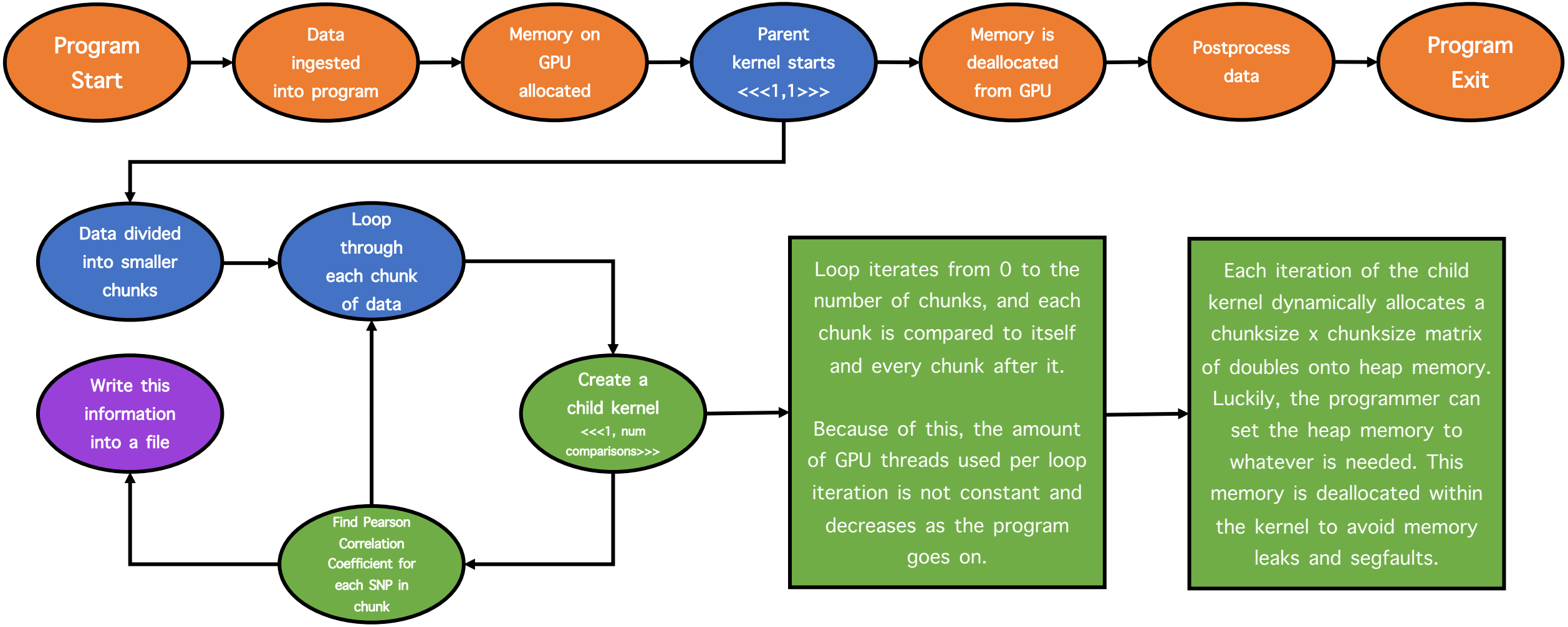
CUDA Parallelization

In addition to completing calculations faster, the design of a GPU is built on parallelization. CUDA maximizes this parallelization.



Using CUDA to Identify Epistasis

Orange: CPU
Blue: GPU (Parent Kernel)
Green: GPU (Child Kernel)
Purple: stdout stream



*Similar workflow to the EpiScan R package described earlier.

Results

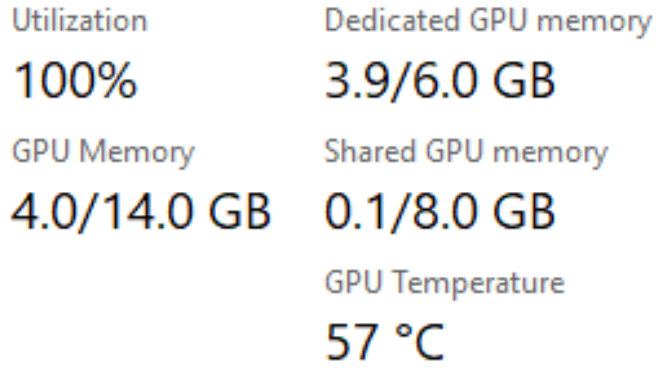
Compared to the R package EpiScan, results are similar. Share most entries, but GPU package has many more false positives.

Still want to compare to other proven epistasis methods as well as simulated datasets. Both of these are being tested currently (once running on the cluster).

Time Improvements

Method	ALVM Data – 13562 SNPs x 711 Samples	ASD Data – 28,089 SNPs x 711 Samples
EpiScan R -	12 min 8 sec	51 min 29 sec
EpiScan R (Parallelized) -	10 min 7 sec	44 min 27 sec
EpiScan GPU -	3 min 27 sec	9 min 38 sec

Note* R times do not include data ingestion, GPU does



Potential Problems

Pearson Correlation Coefficient only identifies linear relationships between SNPs.

Maximum threads allowed in a CUDA block is 1024.

Using the standard output stream as a way to write to a file sometimes causes buffer overflow.

Data must be formatted in a very specific way.

For the Future

1. Begin considering scale of epistasis using the Maximal Information Coefficient rather than the Pearson Correlation Coefficient so that not just linear relationships but all types of relationships are taken into consideration.
2. Make sure that the math is correct and more accurately verify that epistasis is being predicted.
3. Begin seeing capabilities on cluster – stronger GPU but Linux (runs ~10% slower). More memory on cluster GPUs than the GPU the code has been tested on.
4. Start running on larger datasets.
5. Make more user friendly – instead of adding multiple constants, allow user to manipulate code through command line flags.

Sources

Original EpiScan R package can be found here: https://github.com/cran/episcan

My parallelized version of the EpiScan R package can be found here: https://github.com/LanceGuy5/episcan-parallelized

The source code for EpiScan GPU can be found here: https://github.com/LanceGuy5/EpiScan-gpu