Published in final edited form as:

Neuroinformatics. 2018 October ; 16(3-4): 393–402. doi:10.1007/s12021-018-9376-y.

GPU Accelerated Browser for Neuroimaging Genomics

Bob Zigon,
Beckman Coulter, Indianapolis, IN 46268

Huang Li,
Department of Computer Science, Indiana University-Purdue University Indianapolis, IN 46202

Xiaohui Yao,
Department of BioHealth Informatics, Indiana University-Purdue University Indianapolis, IN 46202

Shiaofen Fang,
Department of Computer Science, Indiana University-Purdue University Indianapolis, IN 46202

Mohammad Al Hasan,
Department of Computer Science, Indiana University-Purdue University Indianapolis, IN 46202

Jingwen Yan,
Department of BioHealth Informatics, Indiana University-Purdue University Indianapolis, IN 46202

Jason H. Moore,
Department of Biostatistics, Epidemiology, & Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104

Andrew J. Saykin,
Department of Radiology and Imaging Sciences, IU School of Medicine, Indianapolis, IN 46202

Li Shen, and
Department of Biostatistics, Epidemiology, & Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104

for the Alzheimer's Disease Neuroimaging Initiative

Abstract

Neuroimaging genomics is an emerging field that provides exciting opportunities to understand the genetic basis of brain structure and function. The unprecedented scale and complexity of the imaging and genomics data, however, have presented critical computational bottlenecks. In this work we present our initial efforts towards building an interactive visual exploratory system for mining big data in neuroimaging genomics. A GPU accelerated browsing tool for neuroimaging genomics is created that implements the ANOVA algorithm for single nucleotide polymorphism

Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Information Sharing Statement

Both the source code and documentation are available at https://github.com/lheric/GPU-IGB.

Conflict of Interest: The authors declare that they have no conflict of interest.

(SNP) based analysis and the VEGAS algorithm for gene-based analysis, and executes them at interactive rates. The ANOVA algorithm is 110 times faster than the 4-core OpenMP version, while the VEGAS algorithm is 375 times faster than its 4-core OpenMP counter part. This approach lays a solid foundation for researchers to address the challenges of mining large-scale imaging genomics datasets via interactive visual exploration.

Keywords

GPU; Genomics; MRI; Alzheimer's disease; Data Mining; Versatile Gene Based Association Study

1 Introduction

Recent advances in multimodal brain imaging and high throughput genotyping and sequencing techniques provide exciting new opportunities to study the influence of genetic variation on brain structure and function. Research in this emerging field, known as neuroimaging genomics [3,16,17], holds great promise to better understand complex neurobiological systems, as well as brain structure, function and cognition [4,15,18–20,22].

The unprecedented scale and complexity of these data sets, however, have presented critical computational bottlenecks requiring new concepts and enabling tools. On one hand, it remains a major challenge to develop systematic data mining approaches for revealing complex relationships between the brain (e.g., 1 million voxels) and genome (e.g., 3 billion base pairs). Additional challenges include how to seamlessly integrate the data mining methods with prior knowledge to produce biologically meaningful findings, and how to translate the methods into user-friendly, interactive software tools that optimally combine human expertise and machine intelligence to enable novel contextually meaningful discoveries [6].

In order to address these challenges, using the study of Alzheimer's disease (AD) as a test bed, we've developed a GPU (graphics processing unit) accelerated computational method that enables visual exploratory browsing for interactive mining of complex neuroimaging genomics data. In this framework, a user-friendly heat map interface, coupled with a brain explorer and a genome explorer, is used to visualize high-dimensional research results while focusing on effective integration of techniques from data mining, interactive visualization, and big data analysis. Interactive data mining techniques have the potential to help people gain significant insights into a wide range of problems by means of iterative machine computation and visual exploration.

However, as datasets (e.g., those in neuroimaging genomics) are being generated in larger volumes, higher velocity, and greater variety, creating effective interactive data mining techniques becomes a more difficult task. The use of a GPU can potentially enable the user to freely 'wander' around the data and interactively analyze datasets at scale. In this work, we emphasize the value of the GPU by implementing a browsing system (or browser in short) using both a GPU approach and a multi-threaded, CPU approach. The 375 fold

improvement of the GPU over the 4-CPU version in this study clearly demonstrates the value of a GPU based implementation.

This paper presents the initial efforts for creating such an application. Currently, the focus is on mining imaging genetic associations between structural magnetic resonance imaging (MRI) scans and SNP (single nucleotide polymorphism) genotyping data. The goals include (1) the creation of an application that helps the user better understand the relationships between image and genomic data *at interactive rates*, and (2) to provide a user friendly evaluation scheme for further investigation of potential patterns. An ANOVA (analysis of variance) method was initially prototyped as an example strategy to identify imaging genetic associations at the SNP level. Additionally, the VEGAS (Versatile Gene Based Association Study) algorithm [9,11] was implemented to identify imaging genetic associations at the gene level. The team's contributions include: (1) an application that is accelerated by the GPU; (2) an ANOVA that runs at interactive rates; and, (3) the underlying Monte-Carlo simulation for the VEGAS algorithm runs in less than 1, 500 milliseconds for 10, 000 iterations.

2 Materials and Methods

The voxel-based imaging genetic analysis was applied to demonstrate the interactive application for efficient and effective discovery of imaging genetic associations in the study of Alzheimer's disease. We first describe the imaging and genotyping data, and then present the method for interactively calculating imaging genetic associations and visualizing these statistics from various perspectives.

2.1 Data and Materials

The proposed application was emiprically evaluated using the baseline structural magnetic resonance imaging (MRI) data and genotyping data obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu) [21]. One goal of ADNI has been to test whether serial magnetic resonance imaging, positron emission tomography, other biological markers and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early AD. For up-to-date information, see www.adni-info.org.

Baseline structural MRI scans of both ADNI-1 and ADNI-GO/2 cohorts were downloaded from LONI (adni.loni.usc.edu) and processed with voxel-based mor-phometry (VBM) in SPM8 [13]. Briefly, scans were aligned to a T1-weighted template image, segmented into gray matter, white matter and cerebrospinal fluid maps, and then normalized to the Montreal Neurological Institute space. The gray matter density (GMD) maps were extracted and smoothed with an 8mm FWHM kernel. The resulting GMD images have a dimension of 182 × 218 × 182 (i.e., containing 7,221,032 voxels with 1 × 1 ×1 mm^3 voxel size).

Genotyping data of both ADNI-1 and ADNI-GO/2 [14,15,19] were also obtained from LONI, quality controlled, imputed and combined as described in [8]. In the experiment, 100 non-Hispanic Caucasian subjects were randomly selected. In addition, all of the SNPs

located within 253 genes, which were reported in publications using the ADNI genetics data, were extracted. As a result, there were a total of 22,888 SNPs included in the experiment.

To summarize, the test data set contains 100 subjects, 22, 888 SNPs (from 253 genes), and 7.2 million voxels. These values are selected so that all the data can be loaded onto the GPU in this initial implementation. A future topic is to study whether these size restrictions can be removed by designing an effective memory allocation strategy for larger scale data sets.

2.2 Methodological Background

The use of the GPU for computing in the neuroimaging genomics domain appears to be unique. Earlier researchers typically used tools like Matlab or R that executed on several CPUs. As a result, they frequently limited the number of variables involved due to computational challenges. For example, Kim [7] built a browser for neuroimaging genomic data that implemented the ANOVA (analysis of variance) and ANCOVA (analysis of covariance) algorithms in Matlab. They performed analyses in targeted regions including the hippocampus, amygdala, and the entire temporal lobe. They also allowed for the processing of 137 SNPs. Even with these restrictions it still took 2 to 4 seconds to calculate one statistical map of p-values.

In a similar vein, though genome-wide association studies (GWAS) [5,12,23] have been actively performed, it remains a challenging issue to relate high throughput genotyping data to large scale image data. The reduction in the size of these data types limits a researcher's ability to identify important relationships in an unbiased manner.

It is important to note that GPUs have been applied successfully to various forms of MRI data. Gembris [2] produced the first work involving fMRI analysis on the GPU with the purpose of accelerating the calculation of correlations between voxel time series, a technique used for identifying functional brain networks. Liu [10] used the GPU to accelerate correlation analysis. Eklund [1] used the GPU to create an interactive interface, with 3D visualization, for exploratory functional connectivity analysis.

2.3 Our Approach

To begin, the upper bound on the number of brain regions to consider is increased. For example, the Automated Anatomical Labeling (AAL) atlas subdivides the brain into 116 regions of interest (ROIs) (see the left panel of Figure 1), and many prior studies focus on analyzing the summary statistics of these ROIs instead of measures on millions of voxels. These boundaries have been eliminated and now permit the user to investigate up to several million voxels generated by the MRI (e.g., our test data containing 7.2 million voxels; see the right panel of Figure 1).

On the genomic end, in this study, we allow the user to explore tens of thousands of SNPs (e.g., our test data set contains 22,888 SNPs). In these experiments, the number of subjects is limited to 100 so that all of this data can be loaded into the RAM of the GPU. In the future, strategies will be explored on how to remove this constraint via effective memory management.

Figure 2 shows the primary components of the user interface called BECA (Brain Explorer for Connectomic Analysis). In the lower left hand corner, a 3-dimensional model of a reference brain is displayed with 7.2 million voxels, color-mapped with the p-value of the association between each voxel and the current SNP or gene. At the top of the user interface is the SNP or gene explorer. This region displays the $-log_{10}$(p-value) of the association between each SNP or gene and the current voxel.

The focal point of the user interface is the heat map. The user first selects either the ANOVA or VEGAS algorithm. If the ANOVA is selected, then the y-axis corresponds to the voxels and the x-axis corresponds to the SNPs. The intersection of a given voxel row and SNP column contains the ANOVA p-value of the association between the corresponding voxel and SNP.

If, on the other hand, the user selected the VEGAS algorithm, the y-axis corresponds to the voxels and the x-axis corresponds to the genes. The intersection of a given voxel row and gene column contains the VEGAS p-value (the details will be explained later) of the association between the corresponding voxel and gene.

In both cases, the heat map is a *window* of 80 voxels by 80 SNPs (or 80 genes) into the larger matrix with at most 7.2 million voxels along the y-axis and 22, 888 SNPs across the x-axis for the ANOVA, or 253 genes in the case of the VEGAS algorithm. Using a technique similar to *Google Earth*, the cells of the heat map are computed as the mouse moves about. The mouse wheel can be used to zoom in or zoom out. While the mouse is hovering over a given voxel-SNP (or voxel-gene) location, all of the p-values associated with column are mapped onto the brain explorer. At the same time, the p-values associated with the row are mapped to the genome explorer. The GPU performs approximately two TFLOPS (trillion floating point operations per second) to achieve this level of interactivity.

If we had naively instanced the entire voxel-snp matrix for the ANOVA, it would have required 22,888 × 7,200,000 × 4 = 628 gigabytes of RAM and approximately 85 hours to compute on the GPU. Instead, we used our window based approach and computed everything required to be displayed, in 0.012 seconds. In this case, the RAM requirements were merely 22,888 × 100 × 1 byte = 2.1 megabytes for the snp-subject matrix and 7,200,000 × 100 × 4 bytes = 2,746 megabytes for the voxel-subject matrix. It is unreasonable to expect a researcher to wait 85 hours to compute the entire matrix when they are only interested in examining small subsets that are relevant to their work.

2.4 ANOVA Functionality

The purpose of the ANOVA functionality in the application is to understand the relationship between the gray matter density (GMD) value from the MRI data and the SNP value from the genotyping data. The resulting p-value helps answer the question *'Given a voxel and a SNP, are the mean GMD values at the voxel location different across the SNP genotype groups?'* If the group means are significantly different, that indicates the genetic variation at the SNP location has an effect on the phenotypical GMD value at the voxel location.

The reported p-value for a given voxel-SNP pair across all the subjects is

$$\text{p-value} = F^{-1}\left(\frac{MS_{bg}}{MS_{wg}}\right), \quad (1)$$

where MS stands for Mean Square, the bg subscript refers to 'between group', the wg subscript refers to 'within group', and F^{-1} is the inverse F distribution function.

Table 1 summarizes the intermediate computations required to compute the MS_{bg} and MS_{wg}, where k represents the number of SNP genotype groups, N is the total number of subjects, n_j is the number of subjects within each genotype group, X is an individual observation, $\bar{X}_j$ is the sample mean of the j^{th} group, and $\bar{X}$ is the overall sample mean.

2.5 VEGAS Functionality

The VEGAS algorithm was first described by Liu et al. in 2010 [9]. This gene-based approach considers the association between a trait and all the markers (usually SNPs) within a gene rather than each marker individually. By combining the effects of all SNPs in a gene into a test statistic and correcting for linkage disequilibrium (LD), the gene-based test might be able to detect a stronger collective effect from multiple SNPs than that from each individual SNP. Like the ANOVA, the resulting p-value helps answer a similar question '*Given a gene and a voxel, do the genetic variations within the gene have a collective effect on the GMD measure at the voxel?*' On the surface it might appear that this approach would require fewer computations than the ANOVA, but the underlying Monte-Carlo simulation quickly negates any of those benefits.

The algorithm assumes that for a given gene with n SNPs, an n-element multivariate normally distributed vector with mean $\mu = 0$ and covariance Σ is simulated. The $n \times n$ covariance matrix Σ representing the pairwise LD values is formed by computing all the pairwise Pearson correlation coefficients. The simulation (as described by [9]) is achieved by drawing n values from $\mathcal{N}(0, 1)$ and then multiplying the vector by the Cholesky decomposition of Σ. The result, $C = Chol(\Sigma)$, is a lower triangular matrix such that $CC^t = \Sigma$.

The trouble with this approach is that the Cholesky decomposition of Σ is only defined if Σ is symmetric, positive definite. However, covariance matrices are classified as symmetric, positive semi-definite, which means the Cholesky decomposition can fail due to singularity of Σ. A more robust solution is to extract $[U, S, V^t] = SVD(\Sigma)$, where SVD is the singular value decomposition of its argument. With U orthogonal, S a diagonal matrix of singular values, and V^t orthogonal, we have

$$\sum = U \times S \times V^t. \quad (2)$$

But Σ is symmetric, so we will have

$$\sum = U \times S \times U^t, \quad (3)$$

which means that

$$C = U \times \sqrt{S}, \quad (4)$$

where C is now typically dense. With CC^t as an approximation to the covariance matrix Σ (due to rounding and truncation error caused by numerical computation), the vector $Y^n \in \mathbb{R}^n$ can be generated from a distribution with covariance Σ, by

$$Y^n = C^{n \times n} \times Z^n \quad (5)$$

with $Z^n \sim \mathcal{N}_n(0, 1)$ and $Z = (z_1, z_2, \ldots z_n)$. This new random vector Y^n will have a multivariate normal distribution $Y^n \sim \mathcal{N}_n(0, \sum)$. Y^n is then transformed into a vector of correlated chi-squared components, each with 1 degree of freedom, such that $Q^n = (q_1, q_2, \ldots q_n)$ with $q_i = y_i^2$. The gene based test statistic is now $T = \sum_{i=1}^{n} q_i$ and it will have the same approximate gene-based statistic under the null hypothesis.

When a large number of multivariate vectors (like Q^n) are simulated, the empirical gene-based p-value is the proportion of simulated test statistics that exceed the observed gene-based test statistic. In this case, the observed gene-based test statistic is computed by first executing the ANOVA for all of the SNPs associated with a given gene. The observed statistic is then $\sum_{i=1}^{n} X^{-1}(p_i)$, where p_i are the p-values for each of the n SNPs associated with a gene.

2.6 OpenMP Implementation

In order to understand the benefit of the GPU acceleration, the ANOVA and VEGAS algorithms were first implemented on the CPU using OpenMP. OpenMP is an application programming interface that supports shared memory multiprocessing in C, C++ and Fortran (our code was written in C++). In short, it uses a portable, scalable model that gives programmers a simple and flexible interface for developing parallel programming applications.

First, the OpenMP implementation of the ANOVA algorithm was generated. The pseudo code is shown in Algorithm 1, along with Equation (1) and Table 1. Normally each voxel would be processed sequentially, but, by prefixing the first *For Each* loop with the **#pragma omp parallel for** preprocessor directive, the compiler will generate C++ code that executes in parallel across all of the available CPU cores. The VoxelSNP matrix is the matrix of p-values that are computed from the matrix of VoxelSubject values (derived from the MRI) and the SNPSubject values (supplied by the genotyping data).

Algorithm 1

OpenMP pseudo code for ANOVA on the CPU

```
1 #pragma omp parallel for
2 foreach voxel V in parallel do
3 |   foreach snp S sequentially do
4 |   |   VoxelSNP_{V,S} = ANOVA(V, S,
  |   |     VoxelSubject, SNPSubject)
5 |   end
6 end
```

Now that the ANOVA algorithm has been implemented on the CPU, the focus is turned to the VEGAS algorithm. The pseudo code is described in Algorithm 2 and consists of 3 components. First, lines 2–8 perform an ANOVA calculation for all the SNPs associated with a given gene. Then, lines 11–15 compute the observed p-values for those voxel-gene pairs. Finally, lines 18–31 perform a Monte-Carlo simulation to generate the empirical p-values. It is important to note that Algorithm 2 is called *every time the user moves the mouse over the heat map.* It is also important to point out that by inserting the OpenMP pragma on lines 1, 10, and 17, the algorithm is effortlessly parallelized across the cores in a workstation.

2.7 GPU Implementation

With the CPU implementations complete, the GPU logic is now reviewed. The GPU in use is the GeForce GTX Titan X. It has 12gb of RAM and 3,072 cores. Given the embarrassingly parallel nature of the ANOVA algorithm, Algorithm 3 demonstrates that the p-value can be computed for each voxel V and each SNP S *simultaneously* on the GPU.

The VEGAS algorithm, however, is more difficult to implement efficiently. The first attempt executed in the same amount of time as the CPU version. The final attempt was hundreds of times faster. The following paragraphs describe why this occurred.

Algorithm 2

OpenMP pseudo code for VEGAS on the CPU

```
1  #pragma omp parallel for
2  foreach voxel V in parallel do
3  |  foreach gene G sequentially do
4  |  |  foreach snp S in gene G sequentially do
5  |  |  |  VoxelSNP_{V,S} = ANOVA(V, S,
   |  |  |    VoxelSubject, SNPSubject)
6  |  |  end
7  |  end
8  end
9
10 #pragma omp parallel for
11 foreach voxel V in parallel do
12 |  foreach gene G sequentially do
13 |  |  ObservedPValues_{V,G} =
   |  |    ComputeObserved(VoxelSNP_{V,G})
14 |  end
15 end
16
17 #pragma omp parallel for
18 foreach voxel V in parallel do
19 |  foreach gene G sequentially do
20 |  |  count = 0
21 |  |  for i = 1 to K do
22 |  |  |  Draw n values from N(0,1) giving X^n
23 |  |  |  Y^n = C^{n×n} × X^n
24 |  |  |  T = Σ_{i=1}^{n} y_i^2
25 |  |  |  if (T ≥ ObservedPValues_{V,G}) then
26 |  |  |  |  count = count + 1
27 |  |  |  end
28 |  |  end
29 |  |  VoxelGeneSim_{V,G} = count/K
30 |  end
31 end
32
```

Assume for a moment that you are asked to implement a 1 dimensional Monte-Carlo simulation. Algorithm 4 demonstrates a typical approach. First a random value X is drawn from an $\mathcal{N}(0,1)$ distribution. X is then used as the argument of a function C that returns some value Y. Finally, in line 4, some sort of decision is made with regard to Y and the process continues for K iterations.

Algorithm 3

GPU pseudo code for ANOVA

```
1 foreach voxel V in parallel do
2 |   foreach snp S in parallel do
3 |   |   VoxelSNP_{V,S} = ANOVA(V, S,
  |   |     VoxelSubject, SNPSubject)
4 |   end
5 end
```

Algorithm 4

1-Dimensional draw for Monte-Carlo Simulation

```
1 for i = 1 to K do
2 |   Draw X from N(0, 1)
3 |   Y = C(X)
4 |   Make decision about Y
5 end
```

In the case of the VEGAS algorithm, the variable Z^n is drawn from a normal distribution. This variable Z^n corresponds to the same variable in Equation (5). Algorithm 5 demonstrates a reasonable approach to the generation of Z^n. Line 2 draws n values from $\mathcal{N}(0, 1)$ and assembles them into Z^n. Line 3 shows the function C operating on Z^n to yield the response variable Y^n. Finally, a decision is made with regard to Y^n and the process continues for K iterations. The first attempt at implementing the VEGAS algorithm on the GPU followed this pattern. The pseudo code is shown in Algorithm 6. You can see how the Monte-Carlo simulation is executed *in parallel for every (voxel, gene) pair.*

Algorithm 5

Serial Monte-Carlo for N-Dimensional draw on the GPU

```
1 for i = 1 to K do
2 |   Draw n values from 𝒩(0, 1) giving Z^n
3 |   Y^n = C(Z^n)
4 |   Make decision about Y^n
5 end
```

At this point it's necessary to understand how the performance of logic on a GPU is measured. The rate at which memory is accessed and the number of floating point operations per second (FLOPS) are two critical parameters. On the GeForce GTX Titan X, memory can be accessed at a rate of 336 gb/sec, and the maximum number of giga FLOPS (GFLOPS) is 6144. Algorithm 5 was accessing memory at a rate of 20 gb/sec and the computation rate was 20 GFLOPS. The solution to the poor performance problem is described in Algorithm 7.

Algorithm 6

Slow N-Dimensional VEGAS for GPU

```
1  foreach voxel V in parallel do
2  |   foreach gene G in parallel do
3  |   |   count = 0
4  |   |   for i = 1 to K do
5  |   |   |   Draw n values from 𝒩(0, 1) giving X^n
6  |   |   |   Y^n = C^{n×n} × X^n
7  |   |   |   T = Σ_{i=1}^{n} y_i^2
8  |   |   |   if (T ≥ ObservedPValues_{V,G}) then
9  |   |   |   |   count = count + 1
10 |   |   |   end
11 |   |   end
12 |   |   VoxelGeneSim_{V,G} = count/K
13 |   end
14 end
```

On line 1 a *stream* of random numbers of length $n \times K$ is generated in parallel by the GPU. This corresponds to the number of n-dimensional values that would be generated by K iterations of the Monte-Carlo loop. Line 2 then takes advantage of the structure of our problem. Recall that Equation (5) transforms the normally distributed variable Z^n into a new variable $Y^n \sim C^{n \times n} \times Z^n$ through multiplication by the C matrix. On line 2 a matrix

multiplication between the matrix $C^{n\times n}$ and $X^{n\times K}$ is performed. Finally, line 4 is where some sort of decision is made about the matrix $Y^{n\times K}$. The approach in Algorithm 7 leverages the massive memory bandwidth and computational bandwidth of the GPU in an optimal fashion. Algorithm 8 shows the final implementation and Table 2 shows the performance improvement between Algorithm 5 and Algorithm 7.

The counter intuitive aspect of Algorithm 8 lies in the structure of the two *ForEach* loops. The voxel- gene pairs are processed *sequentially*, but the Monte- Carlo simulation is performed in *parallel*. An 800 fold improvement in the Monte-Carlo simulation is observed when the architecture of the GPU is considered while implementing the logic.

Algorithm 7

Parallel Monte-Carlo for N-Dimensional draw on the GPU

1 In parallel, generate $n \times K$ values from $\mathcal{N}(0,1)$ giving $X^{n\times K}$
2 $Y^{n\times K} = C^{n\times n} \times X^{n\times K}$
3 In parallel, decide about $Y^{n\times K}$

Algorithm 8

Fast N-Dimensional VEGAS for GPU

1 **foreach** *voxel V sequentially* **do**
2 **foreach** *gene G sequentially* **do**
3 In parallel, generate $n \times K$ values from $\mathcal{N}(0,1)$
4 $Y^{n\times K} = C^{n\times n} \times X^{n\times K}$
5 $T_{V,G,:} = \sum_{i=1}^{n} y_{i,:}^2$
6 $count = \sum_{j=1}^{K} (T_{V,G,j} \geq ObservedPValues_{V,G})$
7 $VoxelGeneSim_{V,G} = count/K$
8 **end**
9 **end**

3 Results

The user interface for the application was written in C++ using QT 5.7. The interface logic between the QT code and GPU code was written in C++ using Visual Studio Professional 2015. The GPU code utilized CUDA 8.0 and device driver version number 382.33. CUDA (Compute Unified Device Architecture) is the parallel computing platform and application programming interface that was created by NVIDIA to give application programmers access to the massively parallel hardware of the GPU. All of the host and GPU code is 64 bit. The

accuracy of our VEGAS algorithm for both the CPU and GPU was validated against a targeted genetic association analysis containing 24 genes and 1 phenotype. Our implementation yielded the same result when running the same analysis on the VEGAS website http://vegas2.qimrberghofer.edu.au/.

Figure 3 shows the timing results for the ANOVA implementation. The GeForce Titan X is approximately 110 times faster than the 4-core OpenMP version, and approximately 440 times faster than the 1-core CPU version. It's interesting to note that the 1-core version is faster than the 4-core version up to a window size of 100^2. This suggests that the overhead associated with OpenMP can play a dominant role in execution time for 'small problems'.

Figure 4 shows the timing results for the VEGAS implementation. It can be seen that the GeForce Titan X is nearly 375 times faster than the 4-core OpenMP version. The 4-core version is nearly 4 times faster than the 1-core version. All of this suggests that the GPU version is nearly 1, 500 times faster than the 1-core CPU version.

We have also made a video (see Supplemental Materials) to demonstrate the functionality and real-time interaction performance of the GPU-accelerated imaging genomics browser. The video was recorded on a PC running Windows 10/64 with one Intel i7-7700k CPU, one GTX 1080 GPU and 32GB RAM.

4 Conclusion

This paper has demonstrated how a GPU can be used to accelerate a big data mining application for neuroimaging genomics. With response times on the order of 1 second during interactive calculation of ANOVA and VEGAS heat map p-values, researchers are given unprecedented insight into basic characteristics their data. The 375 fold acceleration of the GPU based VEGAS algorithm is important because it enables the computation of Monte-Carlo simulations at interactive rates. It has been shown how the obvious implementation of the simulation leads to poor response times. Then, an effective arithmetic transformation (that is better suited to the architecture of the GPU) is demonstrated to enable a huge performance increase.

Although this work focused primarily on execution time, there are opportunities for further exploration with regards to implementation. For example, all of the computations were performed using single precision floating point format. We believe double precision should be investigated.

In addition, the user interface communicates to the computing engine using TCP-IP based sockets in a client-server architecture. The overhead of the network bandwidth was never explored because the computing engine was never deployed to a server class machine. It could be that those tests would reveal further opportunities for improvements.

This work implemented the ANOVA and VEGAS algorithms. There are many other algorithms used in imaging genomics analyses such as those implemented in PLINK (http://www.cog-genomics.org/plink2). There are few changes that need to be made to the user interface to include them. The bulk of the work requires implementation of the computations

in OpenMP on the CPU and on the GPU, which is time-consuming and non-trivial. However, the two existing algorithms can serve as a template when implementing new algorithms. We will gradually expand the functionality of this tool by implementing these additional algorithms.

Finally, work is in progress to implement a user-friendly and dynamic graphical interface containing the heat map panel, the brain explorer, and the genome explorer; as well as functionalities to enable visual analytics via interactive exploration. Given that the critical computational challenge is successfully addressed by the GPU-based approach presented in this work, we aim to expand this prototype imaging genomic browser into an integrative system that optimally combines machine intelligence, human intelligence and domain knowledge to advance the scientific discovery in high dimensional brain imaging genomics.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

Acknowledgements Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; Bio-Clinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Funding: This work was funded in part by the National Institutes of Health (NIH) grants R01 EB022574, R01 LM011360, U01 AG024904, P30 AG10133, R01 AG019771 and IUPUI ITDP Program.

References

1. Eklund A, Friman O, Andersson M, Knutsson H. A gpu accelerated interactive interface for exploratory functional connectivity analysis of fmri data. 2011 18th IEEE International Conference on Image Processing. 2011; :1589–1592. DOI: 10.1109/ICIP.2011.6115753
2. Gembris D, Neeb M, Gipp M, Kugel A, Männer R. Correlation analysis on gpu systems using nvidia's cuda. Journal of Real-Time Image Processing. 2011; 6(4):275–280. URL . DOI: 10.1007/s11554-010-0162-9 DOI: 10.1007/s11554-010-0162-9
3. Glahn D, Thompson P, Blangero J. Neuroimaging endophenotypes: strategies for finding genes influencing brain structure and function. Human brain mapping. 2007; 28(6):488–501. DOI: 10.1002/hbm.20401 [PubMed: 17440953]
4. Hariri A, Drabant E, Weinberger D. Imaging genetics: perspectives from studies of genetically driven variation in serotonin function and corticolimbic affective processing. Biological psychiatry. 2006; 59(10):888–97. DOI: 10.1016/j.biopsych.2005.11.005 [PubMed: 16442081]

5. Hirschhorn JN, Daly MJ. Genome-wide association studies for common diseases and complex traits. Nat Rev Genet. 2005; 6(2):95–108. URL . DOI: 10.1038/nrg1521 DOI: 10.1038/nrg1521 [PubMed: 15716906]

6. Kim S, Shen L, Saykin A, West J. Data synthesis and tool development for exploring imaging genomic patterns. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology proceedings. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology 2009. 2009; :298–305. DOI: 10.1109/CIBCB.2009.4925742

7. Kim S, Shen L, Saykin A, West J. Visual exploration of genetic association with voxel-based imaging phenotypes in an mci/ad study. Conference proceedings: … Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference 2009. 2009; :3849–52. DOI: 10.1109/IEMBS.2009.5332570

8. Kim S, Swaminathan S, Inlow M, Risacher S, Nho K, Shen L, Foroud T, Petersen R, Aisen P, Soares H, Toledo J, Shaw L, Trojanowski J, Weiner M, McDonald B, Farlow M, Ghetti B, Saykin A. Influence of genetic variation on plasma protein levels in older adults using a multi-analyte panel. PloS one. 2013; 8(7):e70,269.doi: 10.1371/journal.pone.0070269

9. Liu J, McRae A, Nyholt D, Medland S, Wray N, Brown K, Hayward N, Montgomery G, Visscher P, Martin N, Macgregor S. A versatile gene-based test for genome-wide association studies. American journal of human genetics. 2010; 87(1):139–45. DOI: 10.1016/j.ajhg.2010.06.009 [PubMed: 20598278]

10. Liu W, Zhu P, Anderson JS, Yurgelun-Todd D, Fletcher PT. Spatial regularization of functional connectivity using high-dimensional markov random fields. Med Image Comput Comput Assist Interv. 2010; 13(02):363–370. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4214154/. 20879336[pmid]. [PubMed: 20879336]

11. Mishra A, Macgregor S. Vegas2: Software for more flexible gene-based testing. Twin research and human genetics: the official journal of the International Society for Twin Studies. 2015; 18(1):86–91. DOI: 10.1017/thg.2014.79 [PubMed: 25518859]

12. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira M, Bender D, Maller J, Sklar P, de Bakker P, Daly M, Sham P. Plink: a tool set for whole-genome association and population-based linkage analyses. American journal of human genetics. 2007; 81(3):559–75. DOI: 10.1086/519795 [PubMed: 17701901]

13. Risacher S, Saykin A, West J, Shen L, Firpi H, McDonald B. Baseline mri predictors of conversion from mci to probable ad in the adni cohort. Current Alzheimer research. 2009; 6(4):347–61. [PubMed: 19689234]

14. Saykin AJ, Shen L, Foroud TM, Potkin SG, Swaminathan S, Kim S, Risacher SL, Nho K, Huentelman MJ, Craig DW, Thompson PM, Stein JL, Moore JH, Farrer LA, Green RC, Bertram L, Jack CRJ, Weiner MW, Alzheimer's Disease Neuroimaging, I. Alzheimer's disease neuroimaging initiative biomarkers as quantitative phenotypes: Genetics core aims, progress and plans. Alzheimers Dement. 2010; 6(3):265–73. URL https://www.ncbi.nlm.nih.gov/pubmed/20451875. DOI: 10.1016/j.jalz.2010.03.013 [PubMed: 20451875]

15. Saykin AJ, Shen L, Yao X, Kim S, Nho K, Risacher SL, Ramanan VK, Foroud TM, Faber KM, Sarwar N, Munsie LM, Hu X, Soares HD, Potkin SG, Thompson PM, Kauwe JS, Kaddurah-Daouk R, Green RC, Toga AW, Weiner MW, Alzheimer's Disease Neuroimaging, I. Genetic studies of quantitative mci and ad phenotypes in adni: Progress opportunities, and plans. Alzheimers Dement. 2015; 11(7):792–814. URL https://www.ncbi.nlm.nih.gov/pubmed/26194313. DOI: 10.1016/j.jalz.2015.05.009 [PubMed: 26194313]

16. Seshadri S, DeStefano A, Au R, Massaro J, Beiser A, Kelly-Hayes M, Kase C, D'Agostino R, Decarli C, Atwood L, Wolf P. Genetic correlates of brain aging on mri and cognitive test measures: a genome-wide association and linkage analysis in the framingham study. BMC medical genetics. 2007; 8(Suppl 1):S15.doi: 10.1186/1471-2350-8-S1-S15 [PubMed: 17903297]

17. Shen L, Cooper LA. Imaging genomics. Pac Symp Biocomput. 2017; 22:51–57. URL https://www.ncbi.nlm.nih.gov/pubmed/27896961. [PubMed: 27896961]

18. Shen L, Kim S, Risacher S, Nho K, Swaminathan S, West J, Foroud T, Pankratz N, Moore J, Sloan C, Huentelman M, Craig D, Dechairo B, Potkin S, Jack C, Weiner M, Saykin A. Whole genome

association study of brain-wide imaging phenotypes for identifying quantitative trait loci in mci and ad: A study of the adni cohort. NeuroImage. 2010; 53(3):1051–63. DOI: 10.1016/j.neuroimage.2010.01.042 [PubMed: 20100581]

19. Shen L, Thompson P, Potkin S, Bertram L, Farrer L, Foroud T, Green R, Hu X, Huentelman M, Kim S, Kauwe J, Li Q, Liu E, Macciardi F, Moore J, Munsie L, Nho K, Ramanan V, Risacher S, Stone D, Swaminathan S, Toga A, Weiner M, Saykin A. Genetic analysis of quantitative phenotypes in ad and mci: imaging, cognition and biomarkers. Brain imaging and behavior. 2014; 8(2):183–207. DOI: 10.1007/s11682-013-9262-z [PubMed: 24092460]
20. Viding E, Williamson D, Hariri A. Developmental imaging genetics: challenges and promises for translational research. Development and psychopathology. 2006; 18(3):877–92. [PubMed: 17152405]
21. Weiner MW, Veitch DP, Aisen PS, Beckett LA, Cairns NJ, Cedarbaum J, Green RC, Harvey D, Jack CR, Jagust W, Luthman J, Morris JC, Petersen RC, Saykin AJ, Shaw L, Shen L, Schwarz A, Toga AW, Trojanowski JQ, Alzheimer's Disease Neuroimaging, I. 2014 update of the alzheimer's disease neuroimaging initiative: A review of papers published since its inception. Alzheimers Dement. 2015; 11(6):e1–120. URL https://www.ncbi.nlm.nih.gov/pubmed/26073027. DOI: 10.1016/j.jalz.2014.11.001 [PubMed: 26073027]
22. Yao X, Yan J, Liu K, Kim S, Nho K, Risacher SL, Greene CS, Moore JH, Saykin AJ, Shen L. Alzheimer's Disease Neuroimaging, I. Tissue-specific network-based genome wide study of amygdala imaging phenotypes to identify functional interaction modules. Bioinformatics. 2017. URL https://www.ncbi.nlm.nih.gov/pubmed/28575147
23. Zondervan K, Cardon L. Designing candidate gene and genome-wide case-control association studies. Nature protocols. 2007; 2(10):2492–501. DOI: 10.1038/nprot.2007.366 [PubMed: 17947991]

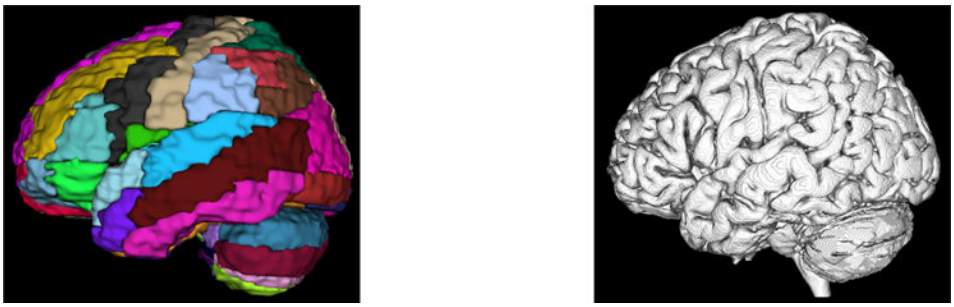

Fig. 1.

Surface rendering of the brain template used in our voxel-based analysis, including 182 × 218 × 182 = 7,221,032 voxels with 1 × 1 × 1 mm^3 voxel size: ROIs from the AAL atlas are shown on left, and a voxel-based isosurface is shown on right.

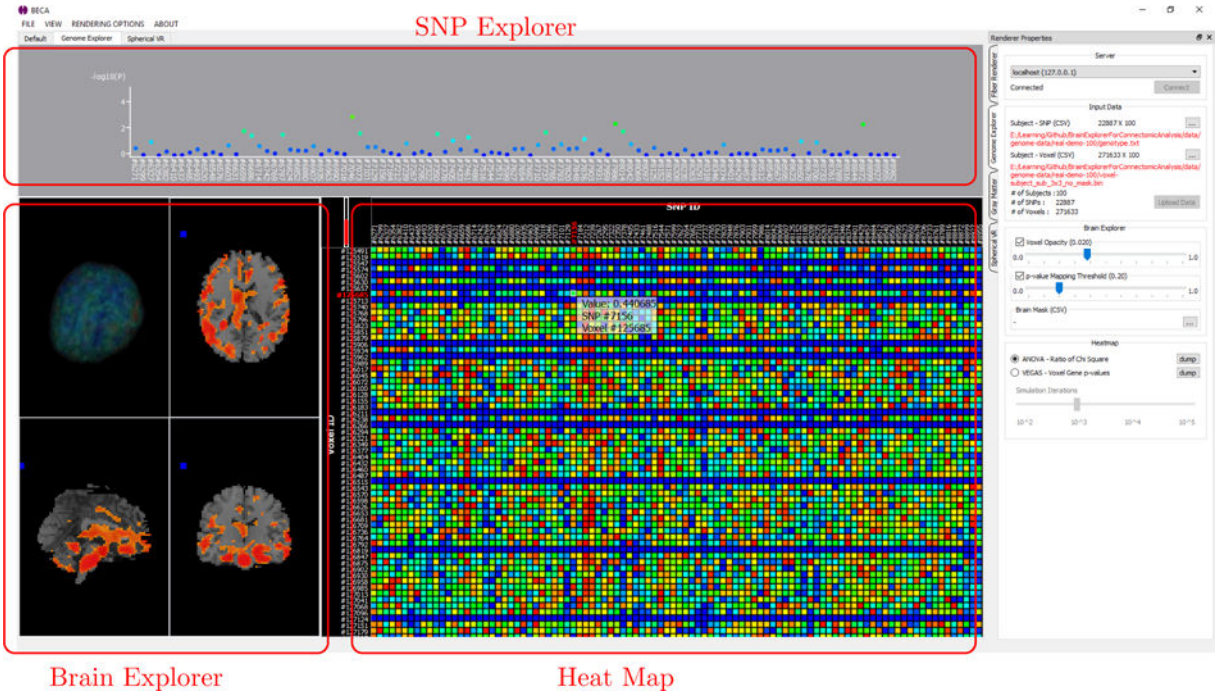

Fig. 2.

Prototype user interface for the Neuroimaging Genomic Browser. Work is in progress to polish the interface and implement functionality to enable visual analytics via interactive exploration.

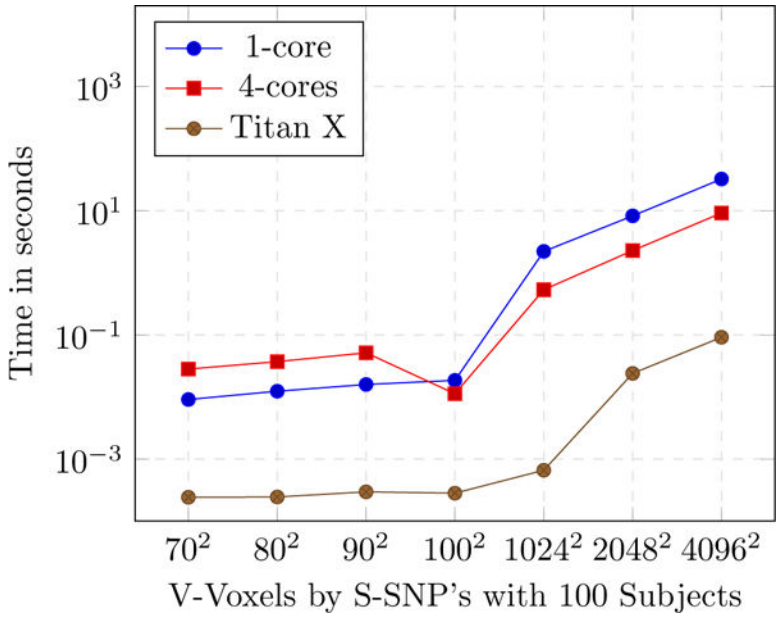

Fig. 3.

Execution times for 1 ANOVA heat map: Time is plotted against the window size (i.e., (# of voxels) × (# of SNPs)).

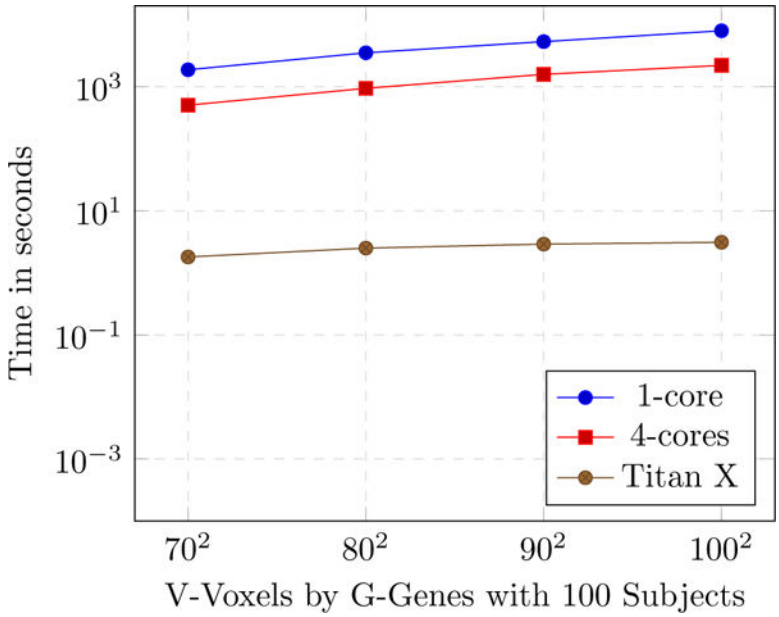

Fig. 4.

Execution times for 1 VEGAS heat map with K=10,000 Monte-Carlo iterations: Time is plotted against the window size (i.e., (# of voxels) × (# of genes)).

Table 1

ANOVA intermediate computations: k represents the number of SNP genotype groups, N is the total number of subjects, n_j is the number of subjects within each genotype group, X is an individual observation, $\bar{X}_j$ is the sample mean of the j^{th} group, and $\bar{X}$ is the overall sample mean.

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square
Between Group	$SS_{bg} = \sum n_j(\bar{X}_j - \bar{X})^2$	$k - 1$	$MS_{bg} = \frac{SS_{bg}}{k-1}$
Within Group	$SS_{wg} = \sum\sum(X - \bar{X}_j)^2$	$N - k$	$MS_{wg} = \frac{SS_{wg}}{N-k}$

Table 2

Comparison of Monte-Carlo implementations on the GPU. On the GeForce GTX Titan X used in our experiments, memory can be accessed at a rate of 336 gb/sec, and the maximum number of giga FLOPS (GFLOPS) is 6,144.

	Serial GPU (Algorithm 5)	Parallel GPU (Algorithm 7)	Theoretical
Memory Bandwidth (gb/sec)	20	155	336
GFLOPS	20	2,000	6,144