26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

GPU-accelerated differential dependency network analysis

Gil Speyer
The Translational Genomics Research Institute
Phoenix, AZ, U.S.A.
gspeyer@tgen.org

Juan J. Rodriguez
The Translational Genomics Research Institute
Phoenix, AZ, U.S.A.
jrodriguez@tgen.org

Tomas Bencomo
The Translational Genomics Research Institute
Phoenix, AZ, U.S.A.
tbencomo@tgen.org

Seungchan Kim
Prairie View A&M University
Prairie View, TX, U.S.A.
sekim@pvamu.edu

***Abstract* — EDDY (Evaluation of Differential DependencY) interrogates transcriptomic data to identify differential genetic dependencies within a biological pathway. Through its probabilistic framework with resampling and permutation, aided by the incorporation of annotated gene sets, EDDY demonstrated superior sensitivity to other methods. However, this statistical rigor incurs considerable computational cost, limiting its application to larger datasets. The ample and independent computation coupled with manageable memory footprint positioned EDDY as a strong candidate for graphical processing unit (GPU) implementation. Custom kernels decompose the independence test loop, network construction, network enumeration, and Bayesian network scoring to accelerate the computation. GPU-accelerated EDDY consistently exhibits two orders of magnitude in performance enhancement, allowing the statistical rigor of the EDDY algorithm to be applied to larger datasets.**

Keywords— EDDY; differential dependency analysis; gene regulatory networks; biochemical pathways; GPU

I. Introduction

Computational analyses of large-scale genomic data generate a large number of associations and observations that can stand as testable hypotheses, but in powering these hypotheses, these approaches often incur considerable computational burden. The Evaluation of Differential Dependency (EDDY) discovers pathways that manifest their activities differently between two groups by computing and comparing two dependency network likelihood distributions from gene expression data. The method uses resampling to estimate network likelihood distributions, then calculates the divergence between them, with statistical significance assessed through permutation test [2]. EDDY has been successfully employed in the analysis of specific cancer types such as glioblastoma [1, 2] and adrenocortical carcinoma [3] as well as large panel of cancer cell lines [4]. The analysis, if applied to the much larger, TCGA pan-cancer dataset, promises compelling hypotheses on the heterogeneity of cancer. However, EDDY's computational demand on larger data set sizes has proven prohibitively large. Relief from this computational burden has presented itself in the form of parallel computing on the Graphical Processing Unit (GPU) architecture, as GPU's possess thousands of computational cores that can cope with EDDY's large but decomposable computational burden by processing calculations in parallel.

II. Methods

A. Evaluation of Differential Dependency

The original Java EDDY algorithm, workflow illustrated in Figure 1, begins with the construction of a pathway-specific distribution of networks for each of two conditions through an inner independence test loop over all possible edges and an outer resampling loop. These groups of Bayesian network structures are then distilled to a unique set and individually assessed as functions of the data using the Bayes Dirichlet (likelihood) equivalent uniform (BDeu) score [5]. The Jensen-Shannon divergence can then be employed to measure the difference of the likelihood distributions of graphs between two conditions [6]. An additional outer permutation loop around all of these steps is added to test for the significance of the divergence, quantified as a p-value, between the distributions of network scores.

Key details of the implementation: 1) Leave-one-out resampling was implemented, which generates roughly as many networks (before uniqueness filtering) as samples. 2) Jensen-Shannon divergences are modeled as a beta distribution, with its model parameters estimated from initial permutation tests, with the p-value evaluated from the

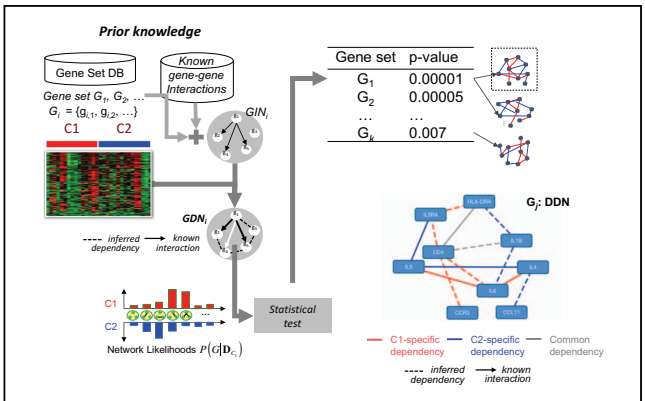

Fig 1. **Evaluation of Differential DependencY (EDDY) workflow.** For a set of samples, the quantized transcriptomic data within a gene set is used to construct a network. Through resampling, a distribution of networks can be generated. The differentiality between two sample groups can then be measured through the divergence of network score distributions, with p-value evaluated via permutation test [1]

0-7695-6380-5/18/$31.00 ©2018 IEEE
DOI 10.1109/PDP2018.2018.00072

estimated models. 3) EDDY also incorporates prior knowledge of gene-gene interactions mined from the Pathway Commons 2 database (http://www.pathwaycommons.org) [1]. Partial weighting of known edges allowed for data to determine the condition specificity. Probabilistic and gene-set assisted approaches together contribute to significantly higher sensitivity and specificity of EDDY, compared to other methods, such as Gene Set Co-expression Analysis and Gene Set Enrichment Analysis [2].

In practice, multiple biological pathways are interrogated to identify rewired pathways between conditions, with statistical significance (p-value < 0.05), as our focus is to generated a ranked list of biological pathways of interest. The multiple testing correction can be added as a post-processing if desired. While the probabilistic framework (via likelihood distributions) and permutation tests result in convoluted computational load, this nested loop is easily decomposed and distributed to multiple nodes in a multiprocessor environment. Hence, the Java implementation has been deployed effectively in cluster environments in the analysis of medium-sized RNAseq datasets over large sets of biological pathways.

B. *Acceleration on GPU*

EDDY-GPU, a GPU-accelerated EDDY algorithm, utilizes the ample and independent computation coupled with manageable memory footprint of the algorithm, broken into three main computational kernel groups, as shown in Fig. 2. RNAseq data for a set of samples, as well as classification information of the samples into two condition groups, are transferred to the GPU. The pair-wise independence tests between all genes in a gene set used to construct graph edges for each condition consume an ample chunk of EDDY's Java computation. Implemented as a kernel, independent threads corresponding to all possible edges, and scaling with the square of the number of genes in a pathway, can execute concurrently. Only a small, relevant subset of the input expression data is required for this computation, demanding a relatively small data footprint. A two-dimensional array, indexed by the pairs of nodes, is used to store the edge probabilities, comparable to that implemented in the Java version. Bootstrapped resamplings, while further scaling the computation, leverages this same data. Thus, parallel cohorts of independence tests process edges for each resampled network, with each network calculated in a concurrent block of concurrent threads, and return binary edge arrays assessed to a preset threshold.

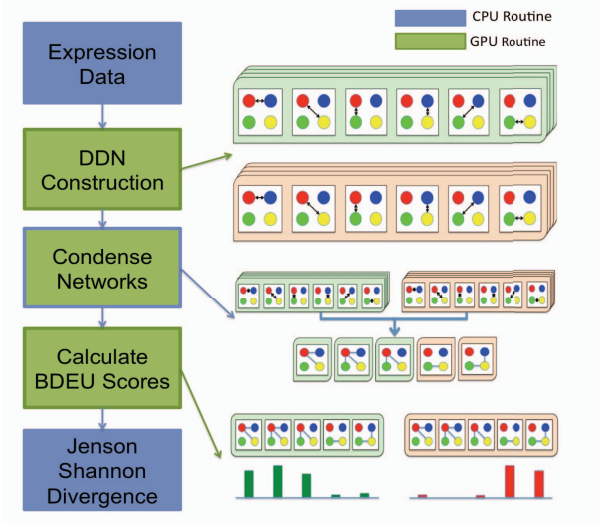

Fig. 2. **Scheme of EDDY-GPU.** After transfer of data to GPU, pairwise comparisons of co-expression within gene sets are run for both conditions over multiple resamplings. Restructuring of edge assessments into condensed graph representation allows efficient parallel pairwise comparison of networks for a minimum unique set. Once distributions of Bayesian scores are computed across networks for both conditions, they are transferred to CPU for divergence calculation.

A second group of kernels reshapes the data for network assessment kernels by first distilling the binary edge arrays into the node and edge lists for the condensed graph representation [7]. A kernel first determines the size of the edge array for allocation. Then, a subsequent kernel assigns an independent thread to each node index within a network block, counting the number and annotating the endpoints of the edges determined for its node. This new data representation can now be exploited by another pairwise comparison, but this time between networks, through a third filter kernel that determines the unique networks from the set determined by the resampling. Results from parallel comparisons of node and edge lists populate a global array, which is then referenced to winnow out the unique networks from the master list on the CPU. In the Java version, a similar Bayesian network data structure had also been employed, storing lists of edges by their sources and targets. However, uniqueness of networks was tested as each network was serially determined.

A final kernel scores each network for each of the two conditions, thus preparing two arrays for the divergence calculation on the CPU. The BDeu score can be easily decomposed into sums of logged gamma functions over states, parents and samples, only requiring that the graph be directed acyclic, which is easily enforced in the graph construction kernel. The score arrays, upon transfer back to the CPU, are used to compute the Jensen-Shannon divergence to assess the difference of the likelihood distributions of graphs between two conditions.

An outer loop for permutation testing is employed to assess the significance of the divergence score. Scrambling the class labels on the samples, the above routine is repeated to yield a null distribution of divergence scores to estimate statistical significance. A final, outermost loop cycles through entire annotated pathway databases, such as REACTOME [8] and BIOCARTA, further iterating the computation. Additional features from the CPU implementation of EDDY (in Java code), such as asymptotic approximation, as well as prior knowledge of gene relationships, were also implemented to emulate the original Java code.

III. Results

A. *Verification and Performance*

EDDY-GPU was verified against a dataset run with the Java version, with Jensen-Shannon scores agreeing to an

accuracy of four decimal places. Performance comparisons were made between Java EDDY run on an Intel Xeon 2.3 GHz with 33GB of RAM and EDDY-GPU run on a NVIDIA Quadro K6000 902 MHz with 12 GB DDR5, with the performance accelerated up to 550 times. Both the number of samples and the number of genes in the gene set affect the performance. The scaling behavior of the program depends on which of these dimensions is increased, and the CPU and GPU implementations exhibit different behavior to this scaling.

Figure 3 shows several run time trends for the CPU and GPU implementations. The x-axis indicates gene set size, while the y-axis indicates the logarithm of run time in ms. Four traces are plotted for each implementation varying the number of samples. The slopes of the CPU runtime traces increase slightly as gene set size increases, but the distance between these traces reveal a greater performance sensitivity to sample size doubling. The slopes of the GPU runtime traces are steeper than those for the CPU runtimes, revealing greater sensitivity to geneset size. The distances between GPU runtime traces, in response to sample size doubling, appear smaller than those for the CPU runtimes, but appear to fan out as gene set size increases.

The gene set size, *g,* scales the independence test and scoring routines as g^2. The performance with sample size is also quadratic in the independence test routine with the nested contingency table and resampling loops. In profiling these codes, the majority of the compute time for the CPU implementation was spent in the independence test, while in the GPU implementation the majority of the compute time was spent in the scoring kernel. Thus, the greater spacing between the CPU runtime traces. For the GPU, the independence test kernel distributes the processing to one thread per edge, which, until the dataset size consumes the compute resources, mitigates the quadratic scaling. Hence, the greatest acceleration is seen with large sample size and small gene set size. The scoring kernel, however, decomposes the computation across the individual nodes. As gene set size increases, the compute on these nodes increases quadratically, accounting for the slope of the traces. Other factors, such as different memory performance issues between the architectures likely contribute to the performance differences.

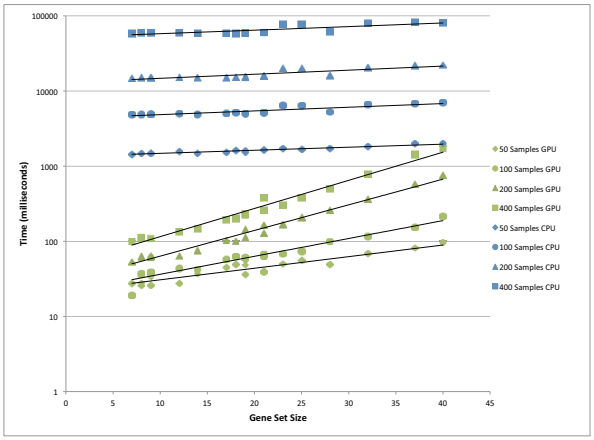

Fig 3. **Performance of of EDDY-CPU and EDDY-GPU.** Logarithmic time (ms) versus gene set size, with four traces per implementation where sample size is repeatedly doubled.

B. *Analysis of Larger Datasets*

1) TCGA Pan-cancer dataset

In addition to computational speed up, EDDY-GPU also enabled the analysis of a large pan-cancer dataset from TCGA of 4,754 samples, an order of magnitude larger than previously possible with the original Java implementation, identifying significantly rewired pathways between samples classified as PIK3CA mutation (465 samples) versus wild type (4,289 sample) [9]. Runtime performance mirrored that seen for GPU traces in Figure 3.

Over the 479 REACTOME pathways, EDDY-GPU identified 79 rewired pathways with statistical significance. One result of the increased power due to sample size is that the number of pathways is larger than had previously been found through hundreds of runs across smaller sample sizes. This will be further discussed in the next section.

Of the 79 significant pathways, the pathway with the lowest p-value, TGF β receptor signaling in epithelial to mesenchymal transition, reveals a dramatic rewiring between PIK3CA wild type and mutation sample groups. The differential dependency network (DDN) and its constituent condition-specific networks (CDNs) for this cancer-associated pathway are shown in Figure 4. In the DDN in the lower left, edges determined through independence test for each of the two sample groups are identified by color. For clarity, the DDN at top left only shows condition-specific edges.

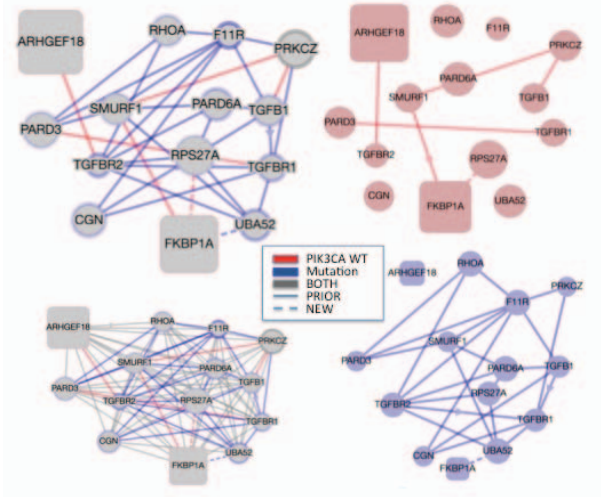

Fig 4. **Differential Dependency (left) and Condition-specific Networks (right) for TGF β receptor signaling in epithelial to mesnechymal transition (EMT) pathway.** In this identified network, edges specific to samples from either PIK3CA wild type or mutation are colored. All edges shown in lower left. For greater clarity, shared edges are not shown in DDN at top left and CDNs. The networks reveal a dramatic signaling shift between PIK3CA wild type and mutation samples in a pathway associated with cancer progression. Known interactions are indicated by solid line edges, while dependencies identified solely through independence test are dashed. Square nodes indicate mediators, genes with important roles in the network.

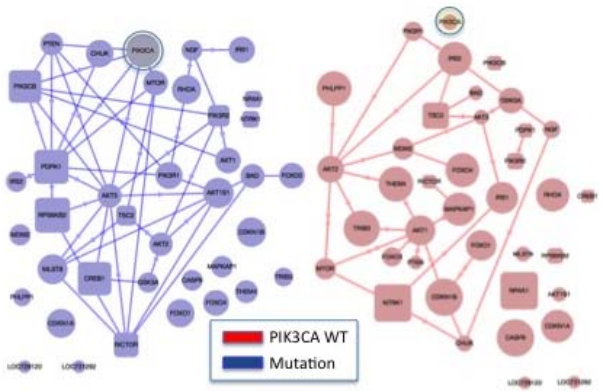

Fig 5. **PIK3CA mutation (left) and wild type (right) condition-specific networks for the PI3K AKT Activation Pathway.** In these CDNs, several mediators are identified, but PIK3CA (highlighted), whose mutation status determined the two sample groups, is not one of them. Shared and purely data-derived edges not shown.

Once the graph is constructed, graphical analysis is employed to identify nodes that are "critical" for the network, denoting those genes as *mediators*. For example, betweenness centrality can identify hubs in a network corresponding to essentiality of those genes in the biological pathway. In our analysis, we split the network up into PIK3CA wild type and mutation networks, calculated betweenness centrality for all nodes, and then identified nodes that had the greatest betweenness centrality *difference* between the two networks, denoting them essentiality mediators. Other nodes with significant rewiring of condition-specific edges were labeled specificity mediators. In Figure 4, square nodes indicate mediators. The wild type CDN features hub mediators such as ARHGEF1B and FKBP1A, while the mutation samples show a more fully connected TGF β-mediated network.

A DDN for another pathway, PI3K AKT Activation, is shown at the bottom of Figure 5. For the 35-node network, 622 edges of a possible 1,225 were found, but, for clarity, the shared and purely data-driven (previously unknown) edges in the figure are not shown. Six nodes (NTRK1, NR4A1, CREB1, PDPK1, RPS6KB2 and PIK3CB) are essentiality mediators and three (PIK3R2, RICTOR and TSC2) are specificity mediators. We note that PIK3CA, the very gene whose mutation status determined the sample groups, is not a mediator in this network. Moreover, most of the 79 identified networks do not even contain the PIK3CA gene. Nevertheless, EDDY-GPU was able to detect subtle differences resulting from downstream or associated effects of the mutation across the samples.

2) Single-cell RNAseq datasets

Single-cell RNA sequencing (scRNAseq) addresses several shortcomings of the population-based average RNA expression from bulk tissue analyte collection. In isolating the specific RNA profile of individual cells, subtle changes in biological behavior are brought into sharp focus. This allows for new research leveraging this data to explore biological mechanisms such as microevolution, dynamic RNA processes and rare disease biology. Moreover, with the ability to characterize

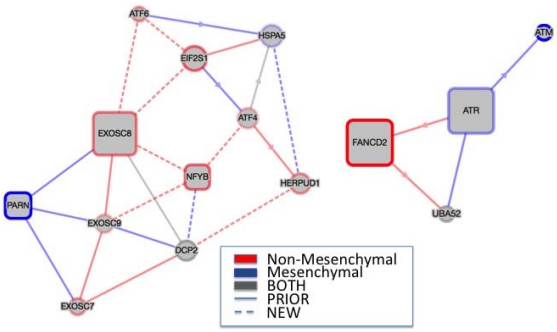

Fig 6. **Differential Dependency Networks from analysis of a single-cell RNAseq dataset.** Two significant pathways, PERK regulated gene expression (left) and Regulation of the Fanconi anemia (right), contrast mesnechymal and non-mesenchymal cell transcriptomes taken from the same tumor sample.

tissues from a single patient within reach, medicine moves ever closer to personalized therapies. The vast amount data (thousands of cells from a single patient) from scRNAseq provides a unique opportunity for EDDY-GPU.

Employing publicly available scRNAseq profiles of samples from glioblastoma (GBM) patients with tumors of one predominant subtype, differential networks could be used to reveal subtle biological distinctions with the remaining cells in the sample, shedding light on mechanisms within intratumoral heterogeneity [10]. Patel and colleagues profiled 430 cells from five primary gliobastomas, revealing that subtype classifiers varied across cells in individual tumors. While the number of samples was smaller than currently achievable with scRNAseq, the dataset structure typified what could be processed at larger scale. Upon download, this expression data was log transformed and quantized for input to EDDY-GPU.

Figure 6 presents two pathways yielded by differential dependency analysis for a predominantly mesenchymal patient sample. In the Regulation of the Fanconi anemia pathway, the FANCD2 gene is identified as an essentiality mediator. FANCD2 is an important protein in DNA double strand break repair. It stays connected to ATR in non-mesenchymal cells. In the mesenchymal cells, FANCD2 disconnects from this network, but now ATR rewires with ATM, which has a similar activity to ATR. This is suggestive of a switch in the type of DNA repair mechanisms and possible lack of DNA double strand repair in mesenchymal subtype.

IV. Discussion

The analysis of these large datasets provided preliminary data for characteristics of EDDY's approach to the statistical analysis of differential dependency at larger scale. Table 1 presents a table of run statistics on multiple datasets to show scaling trends at each of three nested loops, with quadratic fit lines on the logged data. The first panel shows the average number of possible edges per network identified through the innermost independence test kernel, which increases slightly with larger datasets, reflecting the increased power of increasing the number of samples. In the second panel, the number of unique networks found, especially as a proportion of

possible networks, decreases substantially in comparison to that proportion in smaller datasets, reflecting the diminished influence of a single sample using leave-one-out resampling. Nevertheless, the absolute number of networks is comparable to those found with the smaller datasets, indicating that a distribution of network scores can still be generated. The characteristics described above suggest a potential issue with scaling, motivating investigations into alternative sampling strategies for larger datasets. In the last panel, as mentioned above, the proportion of significant pathways found for this dataset slightly exceeds the proportion found for smaller datasets, suggesting an enhanced sensitivity due to the statistical scaling.

# samples	100-200	200-300	300-400	4754
Average fraction of possible edges				0.248
min	0.056	0.154	0.164	
mean	0.195	0.211	0.224	
max	0.34	0.306	0.288	
Average # of networks				22.89
min	9.33	10.56	16.88	
mean	24.86	34.58	35.7	
max	66.89	59.06	61.78	
# significant pathways				79
min	1	6	6	
mean	6.5	13.4	19.9	
max	25	27	42	

Table 1. **Statistical implications for differential dependency analysis at large-scale**. This table present summary statistics (rows) for EDDY runs of varying numbers of samples (columns). The rightmost column shows the data for the single run on the large TCGA pan-cancer dataset from Section III B for comparison. While the proportion of significant edges increases slightly, the average number of networks decreases, especially when considered as a proportion of possible networks. The number of significant pathways increases, reflecting an enhanced sensitivity with larger sample size.

V. Conclusion

This work has presented a GPU-accelerated EDDY, demonstrating its acceleration and application to large datasets. With the decomposition of the independence test kernel to one thread per edge over multiple resamplings, the most time-consuming routine was accelerated. However, in the GPU-accelerated implementation, the scoring kernel then became the time-limiting module. While there may be means to address this with a more efficient kernel, the goal of processing a much larger dataset was achieved. Indeed, despite the serialization of the outer loop over pathways that could be distributed in a cluster environment, the acceleration of the inner loops makes EDDY-GPU analysis feasible on a desktop. Examination of edge and network counts at large-scale showed behavior of the differential dependency approach consistent with the smaller datasets, while a greater count of discovered pathways demonstrated the advantage of greater sensitivity with larger sample counts.

VI. Availability

EDDY-GPU open-source CUDA and C code is freely available through github under openBSD license (https://github.com/dolchan/eddy-gpu).

Acknowledgment

This work was supported in part by the National Cancer Institute, National Institutes of Health [1U01CA168397] (SK, GS), the NVIDIA Foundation Compute the Cure Initiative and the Silicon Valley Community Foundation (SK, GS), and the Helios Education Foundation through the Helios Scholars at TGen Summer Internship Program (JJR – 2015, TB – 2016) at the Translational Genomics Research Institute (TGen) in Phoenix, AZ.

The authors would like to thank Jeff Kiefer, Harshil Dhruv and Michael Berens for helpful discussions.

References

[1] G. Speyer, J. Kiefer, H. Dhruv, M. Berens, and S. Kim, "Knowledge-Assisted Approach to Identify Pathways With Differential Dependencies," *Pacific Symposium on Biocomputing 2016,* vol. 21, pp. 33-44, 2016.

[2] S. Jung and S. Kim, "EDDY: a novel statistical gene set test method to detect differential genetic dependencies," *Nucleic Acids Research,* vol. 42, p. e60, 2014.

[3] S. Zheng, A. D. Cherniack, N. Dewal, R. A. Moffitt, L. Danilova, B. A. Murray*, et al.*, "Comprehensive Pan-Genomic Characterization of Adrenocortical Carcinoma," *Cancer Cell,* vol. 29, pp. 723-36, May 9 2016.

[4] G. Speyer, D. Mahendra, H. J. Tran, J. Kiefer, S. Schreiber, P. Clemon*, et al.*, "Differential pathway dependency discovery associated with drug response across cancer cell lines," *Pacific Symposium on Biocomputing 2017,* vol. 22, pp. 497-508, 2017.

[5] W. Buntine, "Theory refinement on Bayesian networks," *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence. UAI '92,* pp. 52-60, 1991.

[6] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Trans Inform Theory,* vol. 37, pp. 145-151, 1991.

[7] G. M. Merrill D, and Grimshaw A, "Scalable GPU graph traversal," *Proceedings of the 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP '12,* pp. 117-128, February 2012 2012.

[8] A. Fabregat, K. Sidiropoulos, P. Garapati, M. Gillespie, K. Hausmann, R. Haw*, et al.*, "The Reactome pathway Knowledgebase.," *Nucleic Acids Research,* vol. 44, pp. D481-D487, 2016.

[9] T. C. G. A. Network, J. Weinstein, E. Collisson, G. Mills, K. Shaw, B. Ozenberger*, et al.*, "The Cancer Genome Atlas Pan-Cancer Analysis Project," *Nature Genetics,* vol. 45, pp. 1113-1120, 2013.

[10] A. P. Patel, I. Tirosh, J. J. Trombetta, A. K. Shalek, S. M. Gillespie, H. Wakimoto*, et al.*, "Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma," *Science,* vol. 344, pp. 1396-1401, 2014.