CUDA-Accelerated Data-Mining for Putative Heteromeric Transcription Factors and Target Genes Using Microarray Gene Expression Profiles

Edward A. Salinas[1], Amitava Karmaker[2]
[1]Independent Researcher, Rockville, Maryland 20852, USA
[2]University of Wisconsin-Stout, Menomonie, Wisconsin 54751, USA

Abstract - *Understanding protein-protein and protein-DNA interactions is key to understanding the dynamics of gene regulation [3,17]. We here review a previously presented method[1,15,20], based on a variation of microarray expression profile correlation analysis, that seeks to identify interactions between a putative heteropolymeric transcription factor(TF) complex and DNA as well as some experimental results that bolster the argument for the method's validity. The method incorporates correlation coefficients between genes and transcription factors expression profiles, but also between genes and hypothetical TF co-factors, whose expression profiles are estimated by taking minima from constituent profiles. Second, we extend the technique to search for fourth-order protein interactions (k=4). Since a CPU-based analysis would require an execution time on the order of months, we have implemented the k=4 analysis on a CUDA-enabled NVIDIA GPU [16]. With CUDA, we achieved speedups of about 6-fold. Finally, we present the results of the higher order analysis and discuss those results as well as the implementation of the method using CUDA. To our knowledge CUDA has never been used to implement this particular algorithm for microarray gene expression profile analysis.*

Keywords: *Microarrays, Biological Data Mining, CUDA, correlation coefficients.*

1 Introduction

Since the sequencing of the human genome [2] has been completed, the interpretation and biological connotation of sequences and the annotation of functional elements of the genome have been of great interest to researchers. Although a large number of human genes have been identified, their complete regulatory mechanisms are not wholly known at the transcriptional level [3]. To understand gene regulation, we need to identify regulatory elements and the transcription factor complexes that can regulate gene expression, allowing the construction of transcription regulatory networks (TRNs). To control the expression of genes, TF proteins bind to cis-elements in promoter regions and either facilitate or inhibit gene expression. Simply stated, trans-elements can be considered to be "keys", cis-elements "locks", and together "opened doorways" to transcription. By establishing whole TRNs, we may be able to identify novel methods of gene regulation which could have applicability both in the laboratory and clinical settings.

In the post-genomic era, it has been a challenging task in functional genomics to construct TRNs from protein-DNA interactions. *In silico* discovery of transcription regulatory elements is quite effective for prokaryotes, like Escherichia coli [4], where genomes are more compact with many genes being regulated by a single operon. For higher multi-cellular eukaryotes, model-based approaches [3] that discover patterns among co-expressed genes with respect to regulating TFs have been proposed. The techniques involve finding over-represented DNA motifs and common transcriptional regulatory modules among co-expressed genes. A number of statistical and machine-learning algorithms have been used; they include position-weighted matrices, position-specific score matrices, Markov chains, artificial neural networks, and expectation maximization [5-11]. However, it has been reported that techniques incorporating a model-prediction-based approach are susceptible to a high false-positive prediction rate and that a majority of predicted TFBSs have no functional role *in vivo* [12].

Determining new ways to predict which proteins might participate in a heteromeric complex may facilitate the discovery of new TRNs. In this paper, we hypothesize that heteromeric TF complexes can be predicted *in silico* based on their constituent TF expression profiles. Using transcription factor expression profiles and gene expression profiles from microarray data, we review a technique that relies on combinations of TFs and correlation coefficients to predict TF-complexes [1,15,20]. Our dataset includes gene and TF expression profiles from a human female over 115 tissues samples [13]. The technique considers hypothetical TF-complex expression profiles in a given tissue which are estimated by taking minima from the constituent factors from the given tissue. By comparing these hypothetical profiles with each other and with the genuine expression profiles using correlation coefficients, we identify possible complexes. These proposed and hypothetical complexes are given a score based on the comparison. These scores are then compared with scores from other proposed and hypothetical complexes. This comparison leads to the identification of complexes that we believe are more likely to be genuine, and not hypothetical.

Our technique relies on a combinatorial approach selecting a gene, and tuples of TFs and computing many correlation coefficients. Due to extended program execution times, we decided to implement our algorithm using CUDA. We were able to achieve a speedup of approximately *6*-fold. As a result of our analyses, we have been able to perform some validation of our technique as well as identify possible hetero-tetrameric transcription factor complexes. In the following sections of this paper, we describe our technique, its implementation and validation, our findings, and conclude with a brief discussion.

2 Methods and Materials

To carry out the analyses, we used publicly available microarray expression data [13]. The dataset covers a number of human genes and transcription factors expressions across *115* tissue samples, from adrenal tissue to uterine tissue. The dataset is essentially a matrix of expression values with genes indexed by row indices and tissues by column indices; each entry in the data matrix thus represents a gene's expression in a specific tissue. The data is different from typical microarrays in that the genomic DNA is used to estimate mRNA transcript abundance. A subset of *3166* gene transcripts, representing *2526* unique genes, of the data was selected and set aside. Additionally, *352* transcripts, based on information in the entrez-gene and TRANSFAC databases [23, 24] were tagged as transcription factors and also set aside. These data were used for all analyses.

Initial experimental correlation coefficients led to a distribution of coefficients that were weak and centered around zero [1]. This led to the development of a gene data pre-processing step where each gene's expression value was transformed with the equation $y'=ye^{\alpha y}$ where α is a constant. For all experiments done for this paper, the value of α was set to *0.5*. The graph in figure 1 demonstrates the motivation for the transformation.

Given a dataset of *1* row of microarray data for a gene *g* and a set of rows of *N* transcription factors $TF_1,....TF_N$, our technique to asses the relationship between *g* and those transcription factors is as follows. First, the expression data for the gene is transformed with the previously described alpha transformation. Second, borrowing from previous techniques [12] *N* correlation coefficients are computed between the gene's expression values and the individual transcription factor expression values. The Pearson Correlation Coefficients are computed using the formula:

$$r_{xy}=\frac{n\sum x_i y_i-\sum x_i y_i}{\sqrt{n\sum x_i^2-(\sum x_i)^2}\sqrt{n\sum y_i^2-(\sum y_i)^2}} \quad (1)$$

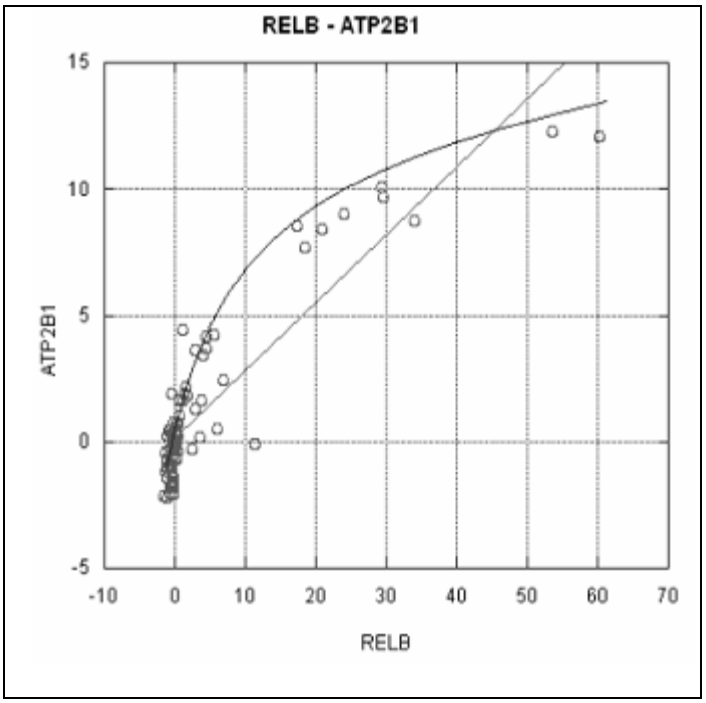

Fig. 1 Data such as depicted this chart helped motivate the α-transformation of the gene data.

Third, between all possible pairs, the hypothetical expression levels are computed and then as many correlation coefficients are computed. The hypothetical dimeric expression profiles are computed by taking the minimum expression value between the two constituent TFs expression values for a given tissue and assigning that value to the corresponding tissue expression for the hypothetical dimer. The same procedure is done for remaining *k=3,...,N* expression profile triplets, quadruplets, etc. of the corresponding hypothetical trimers, tetramers, etc.

For example, for a hypothetical tetramer, its expression at tissue j would be $min(TF1_j,TF2_j,TF3_j,TF4_j)$ where the TFx_j is the x^{th} constituent factor at the j^{th} tissue. This way, altogether, the sum of $C(N,k)$ ("*N choose k*"), for $k=1,2,...,N=k_{max}$ correlation coefficients are computed between the gene expression profile and the real and hypothetical expression profiles; *N* are real and the remaining are hypothetical.

Fourth, the highest-order coefficient (k_{max}), where the *minima* of *N* values for a given tissue was taken is compared with the remaining, lower-order coefficients. The value *a*, which we call the absolute improvement score is computed with the formula:

$$\min_{y \neq k_{\max}} \left(\left| \rho_{k_{\max}} - \rho_y \right| \right) \quad (2)$$

where the minimal absolute value between the highest order correlation and all other correlations is taken. This score we believe helps to distinguish any transcription regulatory signal from the highest-order hypothetical TF from the others. Fifth,

this procedure is carried out for all genes and for all *k-tuples* of transcription factors. In total,

$$c = g\left(\sum \binom{N}{k}_{k=1}^{k_{max}}\right) \quad (3)$$

where g is the number of genes, N is the number of transcription factors coefficients are computed, k is the different numbers of combinations of factors chosen, and k_{max} represents the highest-order polymerization under consideration. For the CFOS/CJUN example later, k_{max} is *2*; for data-mining for heterotetramers, k_{max} is *4*. Note that the sum over combinations is used in Eq. 3 because an analysis requires the computation of lower-order coefficients in the formula for computing the absolute improvement score. Finally, we rank the complexes by their scores. Figure 2 gives a schematic giving an overview of the technique.

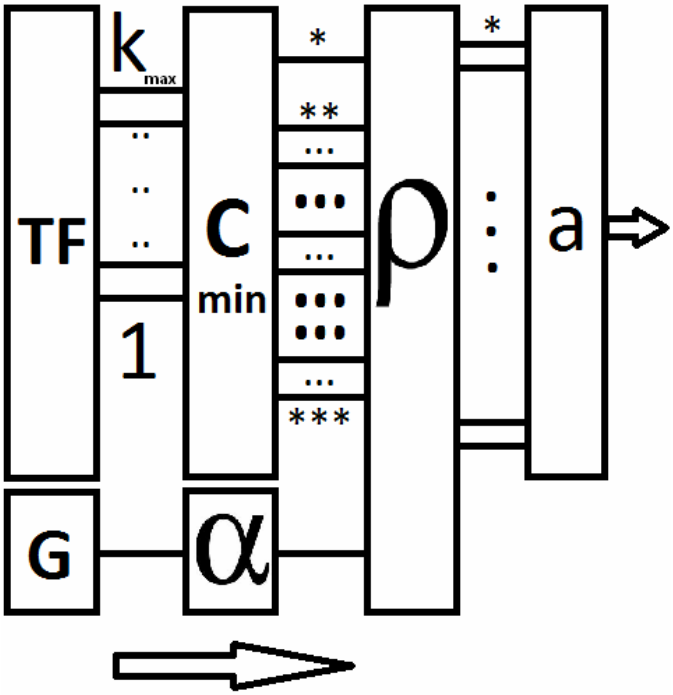

Fig. 2. A schematic shows data-flow and operations of the algorithm. TFs are chosen (*k* in total); a gene is chosen (box "*G*") and then subjected to the alpha transformation (box "α"); 1-*tuples, 2-tuples,…(k_{max}-1)-tuples,* and k_{max}-*tuples* of TFs are chosen and minima are taken to form hypothetical expression profiles (boxes labeled "TF" & "C_{min}"). Finally, correlations are computed between the gene and all of the TF profiles (box "ρ") (both genuine and hypothetical) and compared to generate an absolute improvement score for the highest-order putative heteropolymeric TF complex (box "a"). The scores are used for ranking hypothetical TFs as being likely transcription factor complexes. Legend: The "*" represents the highest-order coefficient, "**", intermediates, and "***" the lowest.

When a gene shares a name with any of the transcription factors, or if any pair of the transcription factors share a name, then the corresponding coefficients and absolute improvement value are not computed. Such analyses are not carried out because we do not wish to consider polymerization involving self-regulating genes or any degree of homo-polymerization.

Central hypotheses of this project are that by taking the minima at a given tissue across expression profiles that we find the hypothetical expression profile of the corresponding polymeric TF and also that the computation and subsequent sorting of the absolute improvement scores may identify and distinguish a transcription regulatory signal from the transcription factors and their hypothetical joining to regulate the corresponding gene by binding to transcription factor binding sites on DNA.

All analyses were completed with a custom-written C/C++ computer program running on a 64-bit Ubuntu/Linux platform with an Intel core i7-960 processor. Perl and bash scripts played a role in loading data into our program as well. Our dataset was not free of missing values. Missing values were indicated with the value *(-18)*. In computing the correlation coefficients, columns (tissues) with missing values were ignored and skipped over. In computing the hypothetical expression profiles, if any single component TF profile had a missing value in a given column, then the hypothetical profile was defined to also have a missing value in that column.

2.1 Methods Validation

To explore the validity of our technique we selected two well-known heterodimer-forming transcription factors CFOS and CJUN [26] from our dataset and applied our algorithm. The two transcription factors together form AP-1. Using the TRANSFAC and ENCODE [24, 25] databases we identified a total of 4 known target genes of the AP-1 TF complex in our gene dataset: TIMP1, GJA1, HMGA1, and MAP4K5. A perfect data-mining technique to identify TFs and their target genes would identify at least these four known target genes for AP-1. As described in the METHODS section, using every pair of transcripts in our dataset belonging to CFOS and CJUN, we carried out a $k_{max}=2$ analysis and computed correlation coefficients, hypothetical expression profiles, absolute improvement scores, and then sorted. After sorting our list and discounting the reported target genes CFOS, and CJUN (the components of AP-1 itself), we found two of the known target genes (HMGA1 and MAP4K5) among the top ten rows of the sorted list of absolute improvement scores and corresponding genes and TFs. Using the hypergeometric distribution, similarly as elsewhere [21, 22], based on the null hypothesis that the four known positives are distributed in the list of 2526 genes at random, we computed that there is a P-value of $8.4 \cdot 10^{(-5)}$ for finding *2* or more of the known target genes in the top 10 of the list sorted by the absolute improvement scores. This indicates that we may reject that null hypothesis, H_0, that the four target genes are randomly distributed in the sorted list at the $\alpha=1\%$ significance threshold. The results are displayed in table 1.

Table 1. Genes and correlations (between CFOS, CJUN and the supposed but genuine AP-1 complex. Known targets of the AP-1 complex are underlined. AI is the absolute improvement score, used for ranking.

	Gene	C1	C2	CC	AI
1	VARS2	0.43	-0.32	0.07	0.36
2	RNU3IP2	0.50	-0.23	0.15	0.35
3	ZFX	-0.40	0.37	-0.06	0.34
4	AP2S1	0.46	-0.21	0.12	0.34
5	LRP6	-0.37	0.37	-0.05	0.32
6	MAP4K5	-0.35	0.32	-0.04	0.32
7	LOC56902	0.42	-0.22	0.09	0.31
8	ERG1	-0.04	0.61	0.30	0.31
9	HMGA1	-0.25	-0.25	0.05	0.30
10	TAPBP	-0.22	-0.22	0.08	0.30

2.2 Data Mining for Heterotetrameric Transcription Factors

To search for putative heterotetrameric transcription factors, we decided to carry out our algorithm with $k_{max}=4$; we wrote a *C/C++* computer program and ran four time trials. Using a quad-core i7 Pentium processor and the OpenMP API for multi-threaded computer programming, our $k_{max}=4$ analysis was over a single gene running *1, 2, 3,* and *4* OpenMP threads at a time. Our time trials were done not to analyze the results, but solely to acquire execution-time data. With four essentially identical time-trials with *1…4* OpenMP threads we saw average execution times of *3090, 1550, 1036,* and *780* seconds. For analyzing all *3166* gene transcripts (including loading the data and printing results), this would be about *113, 57, 38,* and *29* days. Preferring shorting execution times, we deemed such running times too long; in fact a previous analysis never completed [20]. Figure 3, along with some power curves generated with Excel, shows the timing data for the time trials of a single gene.

For these reasons we decided to explore computing the correlation coefficients using *C/C++* and NVIDIA's CUDA architecture. CUDA is a specialized GPU parallel computing architecture implemented on NVIDIA GPUs. CUDA-enabled NVIDIA GPUs allow the parallel execution of threads on the GPU within logically organized grids. The organization is known as an execution configuration. The precise parameters for the execution configuration are set up by the program. A complete description of CUDA is beyond the scope of this paper, but it suffices to say that it enables programs that run on the GPU, called "kernels" to run many threads in a parallel at a time and that CUDA is optimized for arithmetically intense compute-bound programs which have a high ratio of computation operations to I/O operations. More information can be found about CUDA elsewhere [16, 19].

2.3 CUDA-accelerated Data Mining for Heterotetrameric Transcription Factors

Our *C/C++* CUDA-based implementation of the $k_{max}=4$ analysis incorporated *42,875* grid blocks (a *35x35x35* cube of blocks) with each block composed of *1000* threads (a *10x10x10* cube of threads) for its kernel execution configuration. This way, each CUDA-kernel invocation led to the execution of $(10*35)^3=350^3=42{,}875{,}000$ CUDA threads. For a given kernel invocation, the gene is fixed. In the large grid-cube of threads, the row, column, and height indices correspond to row indices in our dataset table of transcription factors. This way, at cell *x, y, z* in the cube, the correlation coefficient with the hypothetical trimeric transcription factor made of the three transcription factors indexed by *x, y,* and, *z* is computed by a thread; control flow, partitioning the cube in two by the inequality $x>y>z$, prevents redundant computation of some coefficients however. If any of the indices are equal, then the correlation is between the gene and either a dimer (if two are equal) or a monomer (if all three are equal). This is because

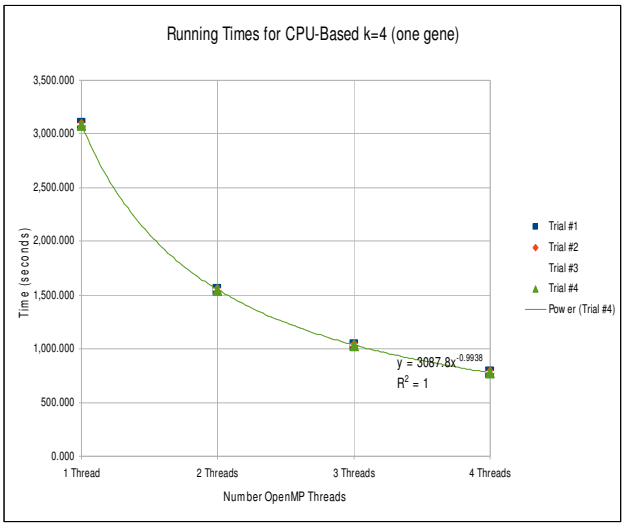

Fig. 3. Four essentially indistinguishable execution time data and power curves for a *k=4* analysis with one gene using *1,2,3, & 4* OpenMP threads

of the property of the minimum function: $min\ (a_1, a_2,...a_N)=a_1$ *if* $a_1=a_2=...=a_N$. This way, a single kernel invocation computes correlations with *k=1, k=2,* and *k=3* which yielded great computational efficiency.

We chose the dimension *350* because of the maximal *1000* threads per block limitation of the CUDA compute capability of the NVIDIA GTX 590 GPU we employed. Because of the *352* TFs in the dataset, the *350x350x350* grid could not accommodate the computation of all the coefficients. To address this issue we introduced grid offsets into our code. This way, the grid always computes the aforementioned *42* million coefficients, but across different indices. By changing the offsets we are essentially moving a *3-D* window that offers views into a *3-D* correlation space. To compute correlations for *k=4*, we also held an index for one of four TFs constant in addition to holding a gene's index constant.

By varying a gene index, a single transcription factor index, and adjusting the kernel's grid offsets we were able to compute a total of *2,013,884,773,648* (≈*2 trillion*) correlation coefficients. Both the CUDA/GPU-based and CPU-based

analyses were done using single-precision floating-point numbers and calculations. Our architecture uses *4* bytes of memory to store a single-precision floating point number; the total number of correlation coefficients computed therefore corresponds to about *8* terabytes of data. Because we do not have such memory capacity available, as we analyzed combinations of genes and TFs, we recorded combinations (as well as the correlations) that had improvement scores above an arbitrary threshold of *0.35*. The top *65,536* combinations were recorded; the remaining data points were not recorded.

CUDA kernels, because of the specialized GPU architecture on which they execute, run faster with fewer control flow statements. Our code could have computed a hypothetical expression profile conditioned on checks that none of the component expression profiles had missing values by looking to see if any of the values was missing (*-18*). To avoid such control flow statements, a *boolean* flag was created from logical AND and in-equality testing operations on the values. In the code, the flag was used as an indicator whose value was interpreted as zero or one. The indicator was incorporated into computations within a *for* loop; intermediate values and counters were adjusted accordingly. During development, this change led to a dramatic speedup.

3 Results

Our *C/C++ k=4* CUDA-based analysis led to two results: *a)* putative heterotetrameric TF complexes and target genes along with the corresponding coefficients sorted by their improvement scores and *b)* timing data for comparison with the CPU-based implementation.

Table 2 presents the top *10* genes and putative TF-tetratmers of our analysis results. The CUDA-accelerated program ran in approximately *4.6* days. Figure 4 displays GPU speedup against estimated OpenMP thread running times (*1, 2, 3,* and *4* OpenMP threads).

Table 2. The top-scoring genes and hypothetical transcription factors from the CUDA-based k=4 analysis. Legend: AI "Abs. improvement"

	AI	GENE	TF1	TF2	TF3	TF4
1	0.73	SERPINA6	EPC1	PLAGL1	WT1	ZNF10
2	0.73	FGB	IRF1	MGA	PAPOLA	SNAPC3
3	0.73	FGB	PRKAR1A	TWISTNB	ZNF155	ZNF83
4	0.72	FGB	EPC1	HMGB2	ITGB3BP	SP110
5	0.72	FGB	EPC1	ITGB3BP	PAPOLA	SP110
6	0.71	AFP	BCL6	ID4	SIAH2	ZNF212
7	0.70	FGB	E2F5	MHGB2	MGA	SP110
8	0.70	FGB	HMGB2	MGA	SP110	ZNF83
9	0.70	FGB	TWISTNB	ZNF155	ZNF198	ZNF83
10	0.70	FGB	E2F5	HMGB2	ITGB3BP	SP110

Interestingly, we note that in the top *10* results from the CUDA-based *k=4* analysis that the FGB gene is seemingly overrepresented as well as the SP110 transcription factor. FGB forms the beta portion of fibrinogen. The protein helps form blood clots. The SP110 transcription factor plays a role forming a part of a leukocyte-specific nuclear-body [14, 23]. We submit these top results to the body of scientific literature as candidates for subjects of further research and inquiry. In addition, the complete list of over *65,000* putative target genes, correlations, and tetrameric TF complexes, dataset and source code are available from the corresponding author of this paper as well.

4 Discussion

We here discuss the efficacy of our algorithm, the role of CUDA in it, its execution, and ways to possibly improve it by parameter adjustment and tuning. We also discuss further ways to test the technique. Finally, we discuss its role of the in a greater bioinformatics context.

Regarding efficacy we note how the program detected two out of four known target genes for the AP-1 complex in the top ten listed target genes (out of *3166* transcripts total). This outcome suggests that the algorithm has some value, but that to be more precise, it needs to be improved. To further explore the algorithm's efficacy, other known dimers and their target genes could be considered and the program's output could be analyzed similarly as was done in this paper.

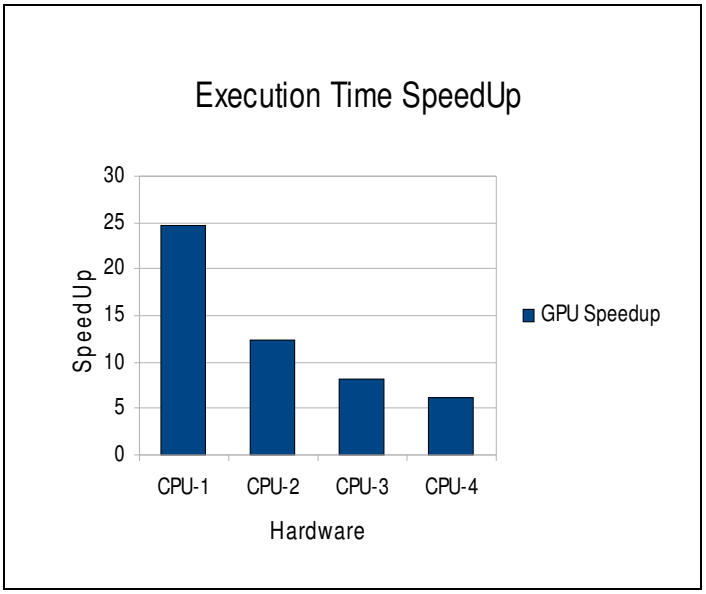

Fig. 4 The speedup (GPU vs. CPU) achieved by the CUDA-based implementation of our algorithm compared with OpenMP threads (1…4).

All of the gene expression profiles were subjected to an alpha transform. The parameterization of alpha leaves a place for experimentation, adjustment, and hopefully improvement. In our analyses for this paper, alpha(α) was set to *0.5*. Perhaps, known heterodimers and their target genes could be set to vary, so that alpha, on a per-dataset basis, could be variable and calibrated or optimized to reveal the most known heteropolymeric transcription factors and their target genes as possible.

Our program computed all of the correlation coefficients with the GPU. Their computations and subsequent comparisons of the absolute improvement score for sorting were carried out by the CPU. The majority of the program's execution time was spent doing such things. This suggests that having the GPU carry out such calculations presents a future avenue to expand the use of the GPU and further contract the running time of the program. Such use of the hardware will require further and continued use of the CUDA API to coordinate kernel calling and data transfers between GPU memory and host memory.

Our dataset set included a total of *44,886* missing values (*40,080* in the gene dataset, *4806* in the TF dataset). Thus, with a total of *3166+352=3518* expression profiles, there is an average of approximately *11.4* missing values per expression profile. From this point of view, every composite hypothetical expression profile of two TFs would have approximately at least that many missing values. Thus, for a given tissue (of *115* total), the probability that its expression value is missing is about *9.93%*. Using the binomial distribution, for a hypoethical dimer there is a nearly *19%* = $P(X \geq 1 | n=2; p=0.0993)$ chance that a given tissue's data is missing. For three TFs this value is just over *25%*. For four, it is nearly *35%*. The more TFs that are under consideration for a given tissue, then the more likely that at least one component TF expression value is missing increases at that tissue. Thus, for higher order composite expression profiles, many tissue expression values would be missing. Thus, for $k=4$, any results should perhaps be used with some caution. To make such analysis more meaningful, missing values could be estimated, but any results from analyses with such imputed values we believe should similarly be used with some, but perhaps less caution. In addition, as k increases, because there are more missing values, the signal-to-noise ratio also increases and that is a further reminder for using results from higher-order analyses with some caution.

Such ideas cause us to remember the fact that the "gold standard" techniques to definitively tell whether or not two or more proteins heteropolymerize include standard "wet lab" molecular biology techniques. Such techniques include crystallography and co-immunoprecipitation (co-IP) [30]. Crystallography [29] involves actual structures, crystallized and examined as 3D structures; co-IP extracts protein-protein-DNA complexes from a solution using antibodies. Such techniques however, are relatively time-consuming and expensive. Moreover, as the number of combinations of proteins whose polymerization is considered increases, more experiments and procedures are necessary to determine whether they bind or not. This means more time and money is needed to make such determinations. Thus our technique explored in this paper may have some value in saving time and money.

To our knowledge, CUDA has never been used to implement this particular technique for microarray data-mining for TF complexes. A somewhat related work for microarray analysis, the TSP algorithm has also been ported to CUDA [27]. Another exists as well that computes correlations and is integrated into the R package for statistical computing [28]. We note that our CUDA kernels' correlation coefficients here are distinct from other CUDA kernels' coefficients in that here minima are taken.

5 Conclusion and Summary

In conclusion, we have presented a set techniques used to analyze a microarray dataset by computing correlation coefficients between gene expression profiles and transcription factor expression profiles across tissues. Its goal is to find multiple transcription factors that bind together and have a target gene whose transcription is modulated. The technique involves hypothetical heteromeric transcription factor profiles whose expressions are estimated by taking minima for each tissue. A scoring function based on a comparison among the correlation coefficients is used to sort and prioritize combinations of genes and transcription factors. The higher scoring combinations are though to be more likely to form transcription factor complexes for the gene. We presented some test data showing the efficacy of our program; it gave interesting results in revealing some *2* out of *4* true positives with a *P-value* of $8.4 \cdot 10^{(-5)}$. To consider *4* TFs at a time, the computational demands are high, so we explored using CUDA-enabled NVIDIA GPUs to speed up the computations. We achieved speedups of about *6x*. For analyzing whether or not four TFs bind, we completed an analysis and have presented some the results from that analysis. Finally, we discussed some of the strengths and weaknesses of the algorithm and our CUDA-implemented technique to speed it up; we also mentioned some ways that the technique could be further improved.

6 Acknowledgements

We acknowledge Dr. Michael Allan for providing ideas for validating the technique and biological insights too. We also acknowledge Dr. Stephen Kwek (Medio Systems) for guidance in implementing the algorithm. All programming was done by Edward A. Salinas.

Funding: All funding for the computer hardware was provided by Edward A. Salinas.

7 References

[1] A. Karmaker, E. Salinas, S. E. Harris and S. Kwek, *Identifying Correlations between Genes and Transcription Co-factors using Expression Profile.*, JCIS, 2007.

[2] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, et al., *Initial sequencing and*

analysis of the human genome, Nature, 409, pp. 860-921, 2001.

[3] J. W. Fickett and W. W. Wasserman, *Discovery and modeling of transcriptional regulatory regions*, Curr Opin Biotechnol, 11, pp. 19-24, 2000.

[4] L. A. McCue, W. Thompson, C. S. Carmack and C. E. Lawrence, *Factors influencing the identification of transcription factor binding sites by cross-species comparison*, Genome Res, 12, pp. 1523-32, 2002.

[5] M. Defrance and H. Touzet, *Predicting transcription factor binding sites using local over-representation and comparative genomics*, BMC Bioinformatics, 7, pp. 396, 2006.

[6] A. E. Kel, E. Gossling, I. Reuter, E. Cheremushkin, O. V. Kel-Margoulis and E. Wingender, *MATCH: A tool for searching transcription factor binding sites in DNA sequences*, Nucleic Acids Res, 31, pp. 3576-9, 2003.

[7] M. C. Frith, M. C. Li and Z. Weng, *Cluster-Buster: Finding dense clusters of motifs in DNA sequences*, Nucleic Acids Res, 31, pp. 3666-8, 2003.

[8] C. T. Workman and G. D. Stormo, *ANN-Spec: a method for discovering transcription factor binding sites with improved specificity*, Pac Symp Biocomput, pp. 467-78, 2000.

[9] M. C. Frith, U. Hansen, J. L. Spouge and Z. Weng, *Finding functional sequence elements by multiple local alignment*, Nucleic Acids Res, 32, pp. 189-200, 2004.

[10] K. Ellrott, C. Yang, F. M. Sladek and T. Jiang, *Identifying transcription factor binding sites through Markov chain optimization*, Bioinformatics, 18 Suppl 2, pp. S100-9, 2002.

[11] W. Ao, J. Gaudet, W. J. Kent, S. Muttumu and S. E. Mango, *Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR*, Science, 305, pp. 1743-6, 2004.

[12] W. B. Alkema, O. Johansson, J. Lagergren and W. W. Wasserman, *MSCAN: identification of functional clusters of transcription factor binding sites*, Nucleic Acids Res, 32, pp. W195-8, 2004.

[13] R. Shyamsundar, Y. H. Kim, J. P. Higgins, K. Montgomery, M. Jorden, A. Sethuraman, et al., *A DNA microarray survey of gene expression in normal human tissues*, Genome Biol, 6, pp. R22, 2005.

[14] *Entrez Gene http://www.ncbi.nlm.nih.gov/entrez/http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=gene*,

[15] E. Salinas, A. Karmaker, BioComp 2009 Analysis of Correlations between Genes and Triads of Transcription Factors Using
Microarray Expression Profiles.

[16] The NVIDIA CUDA Programming Guide http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf

[17] Watson, et. al., Mol. Biology of the Gene, 6th Edition, 2008
Microarray Expression Profiles.

[18] Lee, et. al., Coexpression Analysis of Human Genes Across Many Microarray Data Sets, Genome Res. 2004 June; 14(6): 1085–1094 .

[19] Farber, Rob; CUDA Application and Development, MK Press, 2011

[20] E. Salinas, A. Karmaker, Analysis of Correlations Between Genes and Tetrads of Transcription Factors Using Microarray Expression Profiles, Proc. Of BioComp 2010, Las Vegas, NV, USA

[21] S. Falcon and R. Gentleman Using GOstats to test gene lists for GO term association Bioinformatics (2007) 23(2): 257-258

[22] W. Ewens, G Grant, Statistical Methods in Bioinformatics, an Introduction, 2nd Edition, Springer, 2005

[23] Sayers et. al., Database Resources of the National Center for Biotechnology Information, Nucleic Acids Res. (2009) 37(suppl 1): D5-D15

[24] E. Wingender, P. Dietze, H. Karas, and R. Knüppel, TRANSFAC: A Database on Transcription Factors and Their DNA Binding Sites, Nucl. Acids Res., (1996) 24(1): 238-241

[25] D. Thomas, et al., The ENCODE Project at UC Santa Cruz, Nucl. Acids Res.(2007) 35(suppl 1): D663-D667

[26] Halazonetis TD et al., CJUN Dimerizes with CFOS, Forming Complexes of different DNA Binding Affinities, Cell. 1998 Dec. 2; 55(5):917-924

[27] Magis A., et al., Graphics processing unit implementations of relative expression analysis algorithms enable dramatic computational speedup Bioinformatics (2011) 27(6): 872-873

[28] Buckner, et. al., The gputools package enables GPU computing in R Bioinformatics (2010) 26(1): 134-135

[29] Park, Young-Jun, et. al., Crystal structure of a heterodimer of editosome interaction proteins in complex with two copies of a cross-reacting nanobody; Nucl. Acids Res. (2011) doi: 10.1093/nar/gkr867

[30] Zhang L., et. al., Successful co-immunoprecipitation of Oct4 and Nanog using cross-linking, Biochem Biophys Res Commun. 2007 September 28; 361(3): 611–614