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

***Edward A. Salinas[1], Amitava Karmaker[2] (*corresponding author) (for BIOCOMP 2013)**
[1]Independent Researcher, Rockville, Maryland 20852, USA
[2]Univ. of WI-Stout, Menomonie, Wisconsin 54751, USA

Abstract – *Observing and interpreting intra-protein and protein-DNA interactions is critical to understanding the complexities of gene regulation [3, 16]. We here review a previously presented method [1, 15, 17, 26], using a variation of microarray expression profile correlation analysis, that mines microarray data to find interactions between putative heteropolymeric transcription factor (TF) complexes and target genes. The technique 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 revisit the technique and improve it with parametric calibration. Third, using the calibrated parameter, we adapt our algorithm and implement it to run on the Amazon EC2 cloud to achieve speedup and obtain results in a timely manner.*

Keywords: *Microarrays, Biological Data Mining, Amazon EC2 cloud, 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. Despite the fact that many genes have been catalogued, their complete regulatory interactions are not completely understood at the transcriptional level [3]. To know what orchestrates gene control, we must to discover regulatory elements and any interacting transcription factor (TF) complexes. Tuning the expression of genes, TF regulatory complexes may bind to cis-elements in promoter regions and either facilitate or inhibit gene expression [16]. Knowledge of such interactions would allow the construction of transcription regulatory networks (TRNs) and help researchers understand the dynamics of gene expression. By building and elucidating whole TRNs, we may be able to discover novel routes of gene regulation which may have applicability in many settings, for example the laboratory and the clinic.

It has been an arduous task in functional genomics to build TRNs from protein-DNA interactions. *In silico* data mining of transcription regulatory elements is quite effective for prokaryotes, like Escherichia coli [4], whose genomic landscapes are more compact with numerous genes being controlled by a single operon. For higher eukaryotes, model-based approaches [3] that find patterns among co-expressed genes with respect to regulating TFs have been proposed. The techniques include the finding of over-represented DNA motifs and common transcriptional regulatory modules among co-expressed genes. A variety of statistical and machine-learning algorithms have been employed; they include position-weighted matrices, position-specific score matrices, Markov chains, artificial neural networks, and expectation maximization [5-11]. Such techniques, though, incorporating model-prediction-based approaches have unfortunately been susceptible to high false-positive prediction rates and a majority of the predicted TFBSs have no functional role *in vivo* [12].

Discovering novel means to anticipate which proteins might cooperate in a heteropolymeric complex may aid in the discovery of new TRNs. Here, we hypothesize that heteropolymeric TF complexes of constituent members can be predicted *in silico* based on their constituent TF expression profiles. Using transcription factor activity profiles and gene activity profiles from microarray data, we review a technique that relies on combinations of TFs and correlation coefficients to predict TF-complexes [1, 15, 17, 26]. The dataset includes gene and TF expression profiles from a female human across 115 tissues samples [13]. The method supposes that a hypothetical TF-complex expression profile in a given tissue can be measured by taking minima from the component factors at the given tissue. By combining these values across tissues to create hypothetical TF complex profiles and by comparing and contrasting these profiles with each other and with the genuine expression profiles using correlation coefficients, we hope to discover novel complexes. The putative heteropolymeric complexes are assigned a score-value based on the analysis. These values are then sorted with values from other proposed and hypothetical complexes. This analysis may result in the identification of complexes that we believe are more likely to be genuine, and not hypothetical.

Our technique relies on a combinatorial scheme choosing a gene, tuples of TFs, and calculating correlation coefficients between the gene and TF profiles (both real and hypothetical). As our technique is parameterized, we explore parameter

calibration using known (true positive) TF-pair-gene data. Because timing studies indicated long execution times we modified our code to run on the Amazon EC2 cloud. Using the Amazon EC2 cloud, we obtained speedup and results in about a day, whereas, the local "terrestrial" implementation would have taken months to generate results.

2 Methods and Materials

For the project, we employed public microarray data [13]. The data come from a variety of human genes and transcription factors expressions across *115* tissue types (e.g. bladder, brain, stomach, and uterus). The data is atypical from other microarray data in that genomic DNA material is harnessed to approximate mRNA transcript levels. The dataset may be viewed as a table of transcript expression values with genes indexed by rows and tissues by columns; each cell in the data table thus quantifies a gene's activity in the indicated tissue. A portion of *3166* gene transcripts, from *2526* unique genes, was chosen. Also, *352* transcripts, based on entrez-gene and TRANSFAC databases [20, 21] were marked as transcription factors was also chosen. These two gene and TF datasets were used for all calculations and computations.

Our method contains a genetic profile pre-processing procedure where a gene's activity level may be adjusted with the formula $y'=ye^{\alpha y}$ where α has a constant parametric setting for the algorithm. For all experiments done for this paper, the value of α was set to *0.26*. The graph in figure 1 demonstrates the motivation for the transformation. We later describe a calibration procedure we used to arrive at this parametric setting.

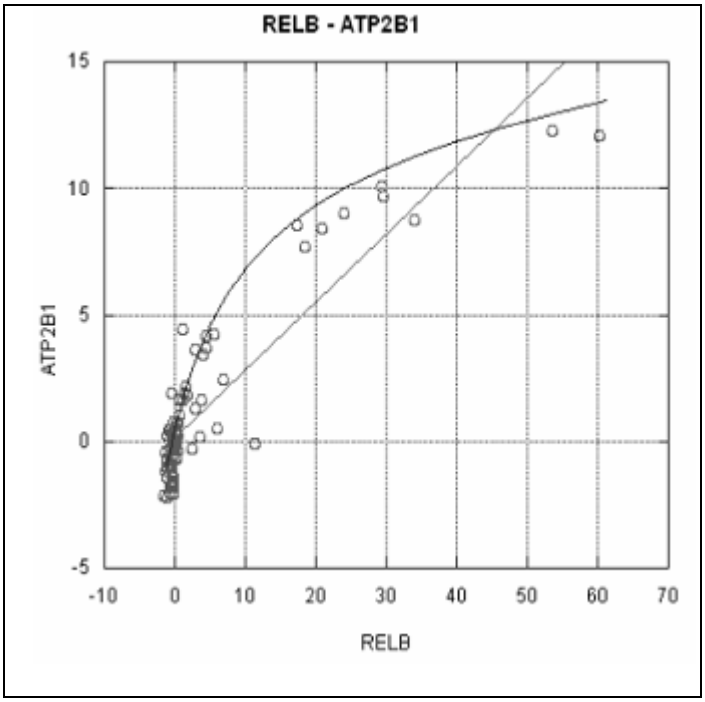

Fig. 1 Data such as depicted this chart helped motivate the α-transformation of the gene data.

Given *1* row (profile) of microarray data for a gene *g* and a set of *N* rows (profiles) of transcription factors $TF_1,....TF_N$, our method to asses the regulatory dynamics between *g* and the *N* transcription factors as a complex is as follows. First, the expression data for the gene is transformed with the previously described alpha transformation. Second, as we have done before [17, 26] *N* correlation coefficients are calculated between the gene's transformed expression profile and the individual transcription factor expression profiles. The Pearson Correlation Coefficients are computed using the formula:

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

Third, between each of the possible pairs, the hypothetical expression levels are computed and then as many correlation coefficients are calculated. 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 TFx_j is the x^{th} constituent factor expression data at the j^{th} tissue. This way, altogether, the sum of $C(N,k)$ ("*N choose k*"), for *k=1,2,...,N* 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 (the k_{max}^{th} coefficient), 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 isolate and reveal any transcription regulatory signal out of the highest-order hypothetical TF from among the others. If this procedure is carried out for all genes and all *k-tuples* of transcription factors, then in total,

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

correlation coefficients are computed. In the formula, g is the number of genes, N is the number of transcription factors, k represents the different numbers of combinations of factors chosen (singletons, pairs, triples, etc.), and k_{max} represents the highest-order polymerization under consideration. For example, for the CFOS/CJUN example we discuss later, k_{max} is *2*; in 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* presents a schematic providing an overview of the technique.

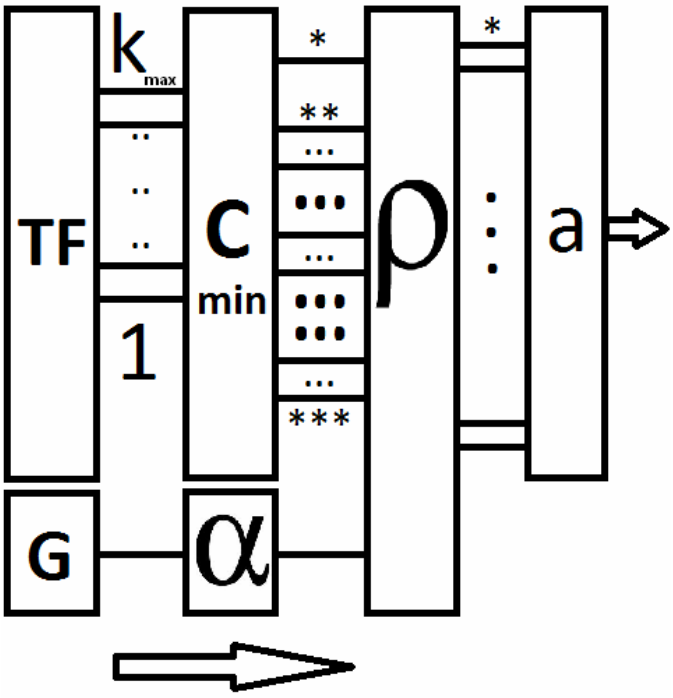

Fig. 2. A schematic shows data-flow and operations of the algorithm. TFs are chosen (k_{max} in total); a gene is chosen (box "*G*") and then subjected to the alpha transformation (box "α"); 1-*tuples, 2-tuples,…(k_{max}-1)-tuples,* and k_{max}-*tuples* of TFs are chosen and minima are taken to form 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 possible regulatory transcription factors, or if any pair of the transcription factors share a name, then the corresponding coefficients and absolute improvement scores are not calculated. This is because we do not aim to consider polymerization involving self-regulating genes or any extent of homo-polymerization.

The 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 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.

All local "terrestrial" analyses were done with a custom-written *C/C++* 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 marked 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 have a missing value in that column as well.

2.1 Validation and Parametric Calibration

To explore the validity of our technique we selected two well-known heterodimer-forming transcription factors CFOS and CJUN [23] from our dataset and applied our algorithm. The two transcription factors together form AP-1. Using the TRANSFAC and ENCODE [21, 22] 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 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. We simultaneously allowed the α parameter to vary from 0 to 1. Looking at the data generated across α–values, data with α in the range 0.25 to *0.27* resulted in the greatest accumulation of true positive target genes in the top-10 list of target genes sorted by the absolute improvement score. From that, we set alpha to *0.26* which is the median (and mean) of the values *0.25*, *0.26*, and *0.27* listed above.

Having calibrated α, we sorted our list of target genes and discounted reported target genes CFOS, and CJUN (the components of AP-1 itself). In the resulting list we found 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 hyper-geometric distribution to carry out a non-parametric statistical test, similarly as elsewhere [18, 19], based on the null hypothesis that the known positives are randomly distributed in the list of *2526* genes, 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 the null hypothesis, H_0, that the 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 hypothesized yet genuine AP-1 complex). Known targets of the AP-1 complex are starred ("*"). AI is the absolute improvement score, used for ranking.

	Gene	C1	C2	CC	AI
1	VARS2	*0.46*	*-0.39*	*0.07*	*0.39*
2	EGR1	*-0.07*	*0.72*	*0.33*	*0.38*
3	HMGA1*	*0.44*	*-0.34*	*0.04*	*0.37*
4	AP2S1	*0.46*	*-0.30*	*0.06*	*0.35*
5	ZFX	*-0.41*	*0.35*	*-0.07*	*0.35*
6	EGR1	*-0.07*	*0.64*	*0.29*	*0.35*
7	LRP6	*-0.33*	*0.37*	*0.01*	*0.34*
8	MAP4K5*	*-0.36*	*0.33*	*-0.02*	*0.34*
9	DPYSL3	*-0.16*	*0.61*	*0.18*	*0.34*
10	RNU3IP2	*0.511*	*-0.25*	*0.17*	*0.34*

2.2 Data Mining for Heterotetrameric Transcription Factors

To data-mine for possible hetero-tetrameric TF complexes, we implemented our algorithm with $k_{max}=4$; we coded a *C/C++* computer program and ran exactly 4 time trials. Employing a quad-core i7 Pentium processor and the OpenMP API for multi-threaded programming, our $k_{max}=4$ calculations were over a single gene running *1, 2, 3,* and *4* OpenMP threads. The trials were carried out not to analyze the results, but simply to obtain execution-time data. From four essentially identical trials 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. Desiring shorter execution times, we deemed such running times too long; in fact a previous analysis never completed [17]. Figure *3*, along with some power curves made with Excel, shows the timing data for the time trials of a single gene.

For these reasons we decided to explore using the Amazon (Elastic Compute Cloud) EC2 cloud to carry out a complete $k_{max}=4$ analysis.

2.3 Cloud-accelerated Data Mining for Heterotetrameric Transcription Factors

The Amazon Elastic Compute Cloud (EC2) is "a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers." [27] Having such a resource and the ability to use and control it would enable the rapid calculations we sought.

For all of our cloud resource requirements and needs, we used the MIT StarCluster tools package [28]. The software's name, STAR, is an acronym standing for "Software Tools for Academics and Researchers". StarCluster helps enable users and developers to programmatically acquire and allocate Amazon EC2 cloud resources, virtualized servers, and set them up as a High-Performance Computing (HPC) cluster

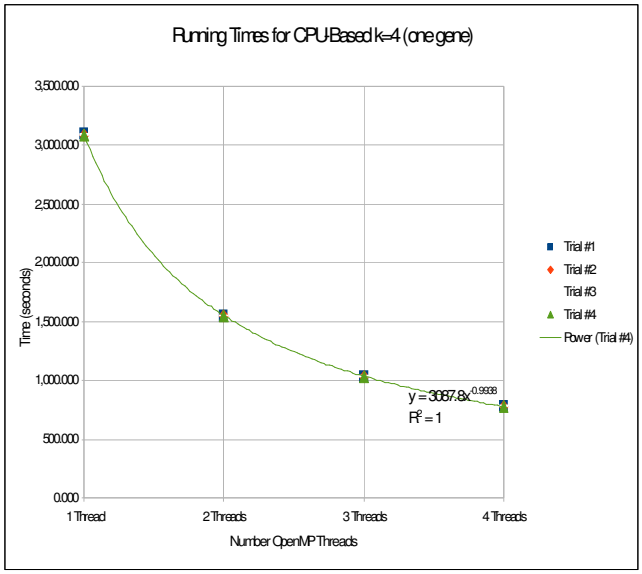

Fig. 3. Four essentially indistinguishable execution time data and power curves for a $k=4$ analysis with one gene using *1,2,3, & 4* OpenMP threads

with the Sun Grid Engine (SGE) scheduling system with a "head" node and "worker" or "execute" nodes. StarCluster is available for download and is python-based. The EC2 HPC cluster accessed via StarCluster proved convenient, critical, and invaluable for the expedient and large-scale deployment of our code.

All nodes were created from Amazon Machine Images (AMIs). AMIs are "pre-configured operating system and virtual application software which are used to create virtual machines within the Amazon Elastic Compute Cloud (EC2). They serve as the basic unit of deployment for services delivered using EC2." [29] All nodes allocated with the STAR package have the AMI ID ami-999d49f0 which refers to an AMI with an Ubuntu-based distribution of the Linux operating system. Such an AMI configuration helped ease the transition of the code to the cloud and in minimizing configuration changes necessary for successful deployment. For testing and running our code we used the *m1.small* and *c1.xlarge* instance types [30]. Different instance types have different hardware and memory specifications. The *m1.small* and *c1.xlarge* have 1 core and somewhat less than 2GB of RAM and 8 cores and about 7GB of RAM respectively.

We ran the computations in the Amazon EC2 cloud in a three-phase fashion. The first phase included verifying that the code would run on the cloud. The second phase consisted of ensuring that the code would take advantage of multi-core SMP virtualized resources and of carrying out time trials to estimate compute resources needed for a full run. The third and final phase consisted of the actual allocation of a large number of virtualized servers and running of the all calculations.

To carry out the first phase of the cloud-computing implementation of our algorithm, we downloaded the STAR cluster package and set it up to allocate a small HPC cluster with the *m1.small* instance types with one worker node and one head node. With that allocated virtual HPC, using the indicated AMI, we achieved rapid transition and porting to the Amazon EC2 cloud. After a small number of modifications, the code was recompiled and tested. Several tests verified the code's proper execution.

To carry out the second phase of the cloud-computing implementation of our algorithm, we next allocated another small HPC exactly as before, but with the *c1.xlarge* instance type. We modified the code slightly to ensure full use of the 8 cores available with the OpenMP API for threads. Several tests verified the use of the cores and correct execution of the code. Moreover, using the smaller, but more powerful HPC cluster, we carried out a small timing study. The timing study consisted of the analysis of a single gene, but in a multithreaded way. The timing study indicated that each gene, run with *8*-way parallelism, could be analyzed with all TFs mining for heterotetrameric factors in about *1* hour and *45* minutes.

We desired to rapidly execute our code in about *24* hours to obtain timely results. Based on the timing study data from phase *2*, we estimated that we needed to allocate *264* *c1.xlarge* nodes to run the analysis across all *3166* genes with all of the TFs. With *8* cores per node, this is a total of *2112* cores. We conferred with Amazon on the scope of our compute resource demands and to verify use and availability.

To carry out the third and final phase of the cloud-computing implementation of our algorithm, we allocated the desired *264* nodes and ran each gene as a sun grid engine job. This way, *3166* jobs were set up. To accomplish this task, a few scripts were composed and run to set up and execute the jobs using the *qsub* command. The *qsub* command permits job submission to the job scheduler for eventual execution. Using the *264*-node cluster, at any given time, about *264* genes were analyzed simultaneously on the *2112* allocated virtual cores.

3 Results

Our *C/C++* k_{max}*=4* 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)* a successful run on the cloud in about *24* hours.

Table 2 presents the top *10* genes, putative TF-tetratmers, and absolute improvement scores, ranked by absolute improvement score of our analysis results.

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

	AI	GENE	TF1	TF2	TF3	TF4
1	0.71	FGB	IRF1	MGA	PAPOLA	SNAPC3
2	0.67	FGB	E2F5	ILF3	MGA	SP110
3	0.67	FGB	EPC1	ITGB3BP	PAPOLA	SP110
4	0.67	FGB	SDCCAG33	TWISTNB	ZNF155	ZNF83
5	0.66	FGB	ILF3	MGA	NFYA	SP110
6	0.65	AFP	ILF3	MGA	SP110	ZNF83
7	0.65	EHHADH	ELL2	EWSR1	PCAF	PPARBP
8	0.65	FGB	IRF1	ITGB3BP	PAPOLA	SNAPC3
9	0.65	FGB	PRKAR1A	TWISTNB	ZNF155	ZNF83
10	0.64	FGB	E2F5	IRF1	ITGB3BP	PAPOLA

We note that in the top *10* results from the cloud-based analysis that the FGB gene is seemingly overrepresented as well as the SP110, ZNF83, and MGA transcription factors. FGB forms the beta portion of fibrinogen; it helps form blood clots. The max-gene-associated protein (MGA) is a TBOX DNA-binding protein. Besides table 2, it has been suggested elsewhere [34, 35], to possibly interact with TFs such as the E2F proteins. The SP110 transcription factor plays a role forming a part of a leukocyte-specific nuclear-body [14, 20]. 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 *40,000* putative target genes, correlations, and heteropolymeric TF complexes, dataset and source code are available from the corresponding author of this paper as well.

4 Discussion

We here briefly discuss the efficacy of the algorithm, the role that missing values may have played in it, the role of the Amazon EC2 cloud in implementing our algorithm, its utility, and briefly compare it with our previous GPU/CUDA based implementation [26]. 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 three out of five known target genes for the AP-1 complex in the top ten listed target genes (out of *3166* transcripts total). This result suggests that the method has some value, but that to be most useful, it should be improved. We believe that the parametric calibration of alpha certainly improved the analysis outcome as well.

The data used had over *44,000* missing values (*40,080* in the gene dataset, *4806* in the TF dataset). With missing values being propagated to the hypothetical composite TF expression profile, they may present a challenge to the algorithm unless they are filled in or imputed. This presents an opportunity for improvement of the technique.

The bioinformatics concept of data mining for true positives causes us to recall the fact that the "gold standard" technique to indicate how two or more proteins heteropolymerize are

standard "wet lab" protocols. Crystallography and co-immunoprecipitation (co-IP) may be used to find such complexes [25]. Crystallography [24] examines actual crystallized structures, in 3D; co-IP isolates protein-protein-DNA complexes out of a solution using immunochemistry techniques. These procedures unfortunately, are neither quick nor cheap. In addition, as the number of proteins whose polymerization is examined goes up, additional work is needed to determine whether they in fact 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.

Our previous CUDA-based implementation [26] ran in about 4 days whereas in contrast, our cloud-based implementation here ran in about *1* day. The programs finished execution so quickly because of the sheer number of nodes, *264*, brought to bear on the computations. As the code scaled linearly during the timing studies, we could have opted for ½ as many nodes, but tolerated two days of execution time. Such possibilities reflect the flexibilities of the Amazon EC2 service. That is a flexibility that a CUDA-based implementation of a program simply may not have unless its implementation is highly used and quite mature.

Reasons to use a local, CUDA-based implementation over a cloud-based method include 1) frequent execution and 2) security. A local CUDA-based program requires no payment for Amazon services and is more secure as no data is in the cloud. Reasons for the converse include 1) the previously mentioned flexibility of the cloud and 2) usage frequencies. For certain applications, the flexibility may prove critical – such as certain periodic batch processes. In any case, these are only a few things to consider. Local compute resources require time, money, power, and other resources, perhaps even staff to maintain them. Each use case deserves its own cost-benefit analysis to lead a compute project to the proper choice. It should be noted that some configurations, even cloud-based GPUs, offer a useful heterogeneous cloud-based system for HPC user needs [33].

To our knowledge, the Amazon EC2 cloud has never been used to implement this particular technique for microarray data-mining for TF complexes; we successfully utilized the Star Cluster package to port our code to the cloud. Bioinformatics has many ways to take advantage of the Amazon EC2 cloud [31]. One interesting method, through CloVR [32], couples "cloud" virtualization technology with local virtualization technology (with "VirtualBox" and "VMWare") to aid in certain large scale metagenomic and BLAST analyses.

5 Conclusion

To summarize, we have presented and reviewed an algorithm used to data mine a microarray dataset by calculating correlations between gene and transcription factor expression profiles over tissues. Its objective is to highlight multiple transcription factors that may heteropolymerize and have a target gene whose transcription is then modulated. The method constructs a hypothetical heteropolymeric transcription factor profile whose tissue expression values are imputed by taking minima over tissues. A score-value procedure based on a comparison among the correlation coefficients is used to rank and order. The higher ranked combinations are believed to be more likely to form heteropolymeric complexes and target the gene. We carried out a calibration protocol with some test data showing the efficacy of our program; it gave interesting results in revealing some *3* out of *4* true positives with a *P-value* of $8.4 \cdot 10^{(-5)}$. To examine the heteropolymerization of *4* TFs at a time, the computational demands are high, so we explored using the Amazon EC2 cloud to speed up the analysis. We successfully ran the code in about *24* hours on a 264-node virtual HPC, and presented some the results from that analysis. Finally, we discussed our algorithm and the utility of the Amazon cloud and compared it with our previous GPU/CUDA-based analysis.

6 Acknowledgements

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

Funding: All funding for use of the Amazon EC2 Cloud 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] Watson, et. al., Mol. Biology of the Gene, 6th Edition, 2008

[17] 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

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

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

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

[21] 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

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

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

[24] 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

[25] 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

[26] CUDA-Accelerated Data-Mining for Putative Heteromeric Transcription Factors and Target Genes Using Microarray Gene Expression Profiles, Proc. Of BioComp 2012, Las Vegas, NV, USA

[27] The Amazon EC2 Web page http://aws.amazon.com/ec2 accessed 3/15/2013

[28] The Star Cluster home page, http://star.mit.edu/cluster/ accessed 3/15/2013

[29] Amazon AMI Webpage https://aws.amazon.com/amis accessed 3/15/2013

[30] Amazon EC2 Instance Types web page http://aws.amazon.com/ec2/instance-types/ accessed 3/15/2013

[31] Fusaro VA, Patil P, Gafni E, Wall DP, Tonellato PJ (2011) Biomedical Cloud Computing With Amazon Web Services. PLoS Comput Biol 7(8): e1002147. doi:10.1371/journal.pcbi.1002147

[32] Angiuoli, S.V., Fricke W.F., et al., CloVR: A virtual machine for automated and portable sequence analysis from the desktop using cloud computing, BMC Bioinformatics 2011, 12:356

[33] Leinweber, M., et al., GPU-based Cloud computing for comparing the structure of protein binding sites, Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference, vol., no., pp.1-6, 18-20 June 2012

[34] A.M.L. Liekens, et al., BioGraph: Unsupervised Biomedical Knowledge Discovery via Automated Hypothesis Generation, Genome Biology 12:R57, 2011.

[35] http://biograph.be/concept/graph/C1422345/C1167128