Gene Expression Analysis of HIV-1 Linked p24-specific CD4+ T-Cell Responses for Identifying Genetic Markers

Sanjeev Raman and Carlotta Domeniconi
Information and Software Engineering Department
George Mason University
sraman@gmu.edu carlotta@ise.gmu.edu

Abstract

The Human Immunodeficiency Virus (HIV) presents a complex knot for scientists to unravel. After initial contact and attachment to a cell of the immune system (e.g. lymphocytes, monocytes), there is a cascade of intracellular events. The endproduct of these events is the production of massive numbers of new viral particles, death of the infected cells, and ultimate devastation of the immune system. HIV is an epidemic and a crisis in many continents [1]. Since there are many variations of the virus and differences in people’s genetic make-up, rapid diagnosis and monitoring of tailored treatments are essential for future medicine. To combat this problem, microarray technology can perform a single scan on thousands of genes. However, without a proper research design and data mining techniques, the results from such a technology can be very skewed. Thus, using a normalized, clean dataset (time-series) from the CD4+ T-cell line CEM-CCRF, we designed and implemented hierarchical clustering and pattern-based clustering algorithms to identify specific cellular genes influenced by the HIV-1 viral infection. This research can contribute to the HIV Pharmacogenomics field by confirming HIV genetic markers, which would lead to rapid diagnosis and customized treatments.

Keywords: pattern-based clustering, hierarchical clustering, HIV, gene expression analysis, genetic markers.

1. Introduction

Since viruses (i.e. human immunodeficiency virus type 1 - HIV-1) can impact a diverse set of host cell’s biochemical processes, many of these interactions can be characterized by changes in cellular mRNA levels that could depend on both the stage of infection and the biological stage state of the infected cell [2]. For example, viral infection induces the interferon antiviral response, modulates the cell’s transcriptional, translational, and trafficking machinery. Thus, the recent emergence of high-density DNA arrays (microarrays and oligonucleotide chips) has revolutionized gene expression studies by providing a means to measure mRNA levels for thousands of genes simultaneously [3].

In this paper we conducted a gene expression analysis, which is a novel approach to identifying and profiling genes related to the pathology and responsiveness of a potential treatment. In the case of HIV-1, where the infection is worldwide and the subtypes are many, measuring the efficacy of a potential treatment in distinct populations from a molecular level is essential. Since people can have different responses to treatments based on their genetic make-up, the Food and Drug Administration is going to mandate pharmacogenomic studies to be submitted with drug submission research [4].
Thus, we focused on two main objectives:

1. Researching and discussing the various techniques and approaches for gene expression analysis.

2. Identifying and confirming global genetic markers for HIV-1 by designing and implementing data mining algorithms.

Our approach utilized two proven computational techniques: hierarchical clustering and pattern-based clustering. All the data analysis will be based on time series data and genes from the CD4+ T-cell line CEM-CCRF in order to identify specific cellular genes influenced by HIV-1 viral infection. The details of the research design are discussed in subsequent sections. Prior research has been conducted in this field, however, the research was done when the technology to do the data analysis was very new to the market (1998 and 1999) and thus, the analysis was very broad. This is because the focus was on classes of genes. In contrast, in this work our objective is the identification of potential global genetic markers [5].

1.1 Motivation and Contribution

The results from this study can give great insight of how to quickly measure the effectiveness of a treatment according to a person's genetic make-up and what specific genes are important in the regulation of HIV/AIDS. This study will help to confirm previous results from a molecular level and contribute to the overall knowledge domain of pharmacogenomic-HIV research [6], which will eventually lead to customized diagnosis and treatment of the disease.

2. Background and Related Work

HIV is a retrovirus and thus, contains a genome composed of two copies of single stranded RNA housed in a cone-shaped core surrounded by a membrane envelope. A transfer RNA is located near the 5' end of each RNA and serves as an initiation site for reverse transcription. Viral enzymes housed in the core include reverse transcriptase, protease, and integrase. The envelope proteins consist of a transmembrane portion (gp41) and a surface molecule (gp120), which is the attachment site to the receptor on the host cell. Like all retroviruses, HIV-1 genome encodes for gag, pol, and env. However, HIV-1 also contains six accessory gene products that are somewhat essential for HIV replication and reproduction (tat, rev, vif, vpu, vpr, and nef) [7].

Microarray expression analysis has become one of the most widely used functional genomics tools. Efficient application of this technique requires the development of robust and reproducible protocols. This process involves several aspects of optimization such as Polymearse Chain Reaction amplification of target cDNA clones, microarray printing, probe labeling and hybridization, and developed strategies for data normalization and analysis [8].

Efficient expression analysis using microarrays requires the development and successful implementation of a variety of laboratory protocols and strategies for fluorescence intensity normalization. The process of expression analysis can be broadly divided into three stages [9]: (1) Array Fabrication; (2) Probe Preparation and Hybridization; and (3) Data Collection, Normalization and Analysis.

The genome of an organism is the genetic code that regulates the expression of various features and functions of the organism. This regulation is brought about by the co-ordination of various genes in the genome. These genes communicate with each other to trigger or suppress the expression of each other. A typical experiment on the gene expression would therefore have to take into account the simultaneous observation of these genes.

2.1 Hierarchical Clustering

Hierarchical clustering is by far the most popular method to cluster microarray data. There are two types of hierarchical clustering – agglomerative and divisive. Agglomerative clustering takes an entity (i.e. a gene) as a single cluster to start off with and then builds bigger and bigger clusters by grouping similar entities together until the entire dataset is encapsulated into one final cluster. Divisive hierarchical clustering works the opposite way around – the entire dataset is first considered to be one cluster and is then broken down into smaller and smaller subsets until each subset consists of one single entity. The sequence of clustering results is represented by a hierarchical tree, called a dendogram, which can be cut at any level to yield a specific number of clusters [10].

The agglomerative approach is most commonly used in microarray analyses. The reason is that divisive clustering is more computationally expensive when it comes to making decisions in dividing a cluster in two, given all possible choices. However, the divisive approach retains the "super structure" of the data. This means that one can confidently say that the root or "upper levels" of the dendogram are highly representative of the original structure of the data. Although, this does not mean that the agglomerative approach is not just as robust [10]. We focused on the agglomerative approach.

The basic rules for agglomerative hierarchical clustering are as follows [11]:

1. Derive a vector representation for each entity (i.e. gene expression values for each experiment make up the vector elements for a specific gene);

2. Compare every entity with all other entities by calculating a distance. Input that distance into a matrix. Calculation of the distance depends on:

a. the linkage method (distance between clusters) being implemented;
b. the actual distance measure used;

3. Group closest two entities (or clusters) together (which makes a new cluster) and go back to step 2, considering the new cluster as a single entity, recalculate distances between entities and cluster closest entities together. Step 2 should be repeated until all entities are contained within one big cluster.

The distance between clusters is usually computed in one of three different ways: *Single linkage* is the minimum distance between a point in one cluster and a point in the other cluster; *average linkage* is the average of the distances between points in one cluster and points in the other cluster; *complete linkage* is the largest distance between a point in one cluster and a point in the other cluster. Thus, an agglomerative hierarchical clustering approach can be implemented using, for example, the Euclidean distance measure and the average linkage method.

2.2 Pattern Based Clustering

Pattern-based clustering (or p-clustering) groups a set of objects based on their coherent trend in a subset of dimensions. This differs slightly from subspace clustering as subspace uses global distance/similarity measures, which may not be able to detect coherent trends. There are two distinct features of pattern-based clustering: there is no global defined similarity/distance measure, and clusters may not be exclusive. When using pattern-based analysis, subsets of genes whose expression levels change coherently under a subset of conditions are identified. This analysis can be critical in revealing the significant connections in gene regulatory networks.

There are two issues to be concerned with when performing pattern-based clustering. Issue one is that there can be many pattern-based clusters, thus maximal pattern-based clusters must be determined. Second, the methodology to mine maximal pattern-based clusters must be efficient [12]. Traditionally, a pattern score is used to calculate the similarity between two objects. For example, [12] defines the pattern score of two objects r_x, r_y on two attributes a_u, a_v as follows:

$$pScore\left(\begin{bmatrix} r_x.a_u & r_x.a_v \\ r_y.a_u & r_y.a_v \end{bmatrix}\right) = \left\|(r_x.a_u - r_y.a_u) - (r_x.a_v - r_y.a_v)\right\|$$

Also, a threshold is established. For example, for any objects $r_x, r_y \in R$ and any attributes $a_u, a_v \in D$, in [12] it is required:

$$pScore\left(\begin{bmatrix} r_x.a_u & r_x.a_v \\ r_y.a_u & r_y.a_v \end{bmatrix}\right) \leq \delta \quad (\delta \geq 0)$$

In regards to maximal pClusters, if (R, D) is a δ-pCuster (that is, all pairwise objects in R have a $pScore \leq \delta$ with respect to attributes in D), then every cluster (R', D'), where $R' \subseteq R$ and $D' \subseteq D$, is a δ-pCuster (anti-monotonic property). That is, a large pCluster is accompanied with many small clusters. Therefore, the idea is to mine only the maximal pClusters. A δ-pCuster is maximal if there exists no proper super cluster that is a δ-pCuster [12].

3. Research Design and Methodology

As mentioned in the previous section, gene expression analysis can be divided into sequential stages: array fabrication, probe preparation and hybridization, data collection, normalization, and analysis. In this section, we explain and describe in detail the specific design and techniques needed to perform the gene expression analysis of HIV-1 linked p24-specific CD4+ T-cell responses for identifying genetic markers.

The human immunodeficiency virus type 1 (HIV-1) infection alters the expression of host cell genes at both the mRNA and protein levels. To obtain a more comprehensive view of the global effects of HIV infection of CD4-positive T-cells at the mRNA level, we analyze a cDNA microarray dataset generated from the University of California, San Diego [5]. We perform p-clustering and hierarchical clustering analysis on mRNA expressions of approximately 6800 genes. These mRNA expressions were monitored at eight time points [0.5h, 2h, 4h, 8h, 16h, 24h, 48h, 72h] from a CD4+ T-cell line (CEM-GFP) during HIV-1 infection. The CEM-GFP cells were inoculated with HIV-1 at a multiplicity of infection of 0.5, an inoculum sufficient to ensure that every cell is contracted by virus particles. Aliquots of cells were obtained as described above. A mock infection served as a control at each time point, essentially replacing the volume of viral input by an equivalent volume of culture medium from uninfected cells. Each sample was tested on two chips and the average was taken. Normalization for this dataset was done using global normalization and scaling. The objective is to identify a specific set of universal genes that can be used as genetic markers for measuring the effectiveness of a potential treatment based on time series patterns and levels consistently changing more than 1.5-fold. A fold is defined mathematically as $\log_2(Cy5/Cy3)$, where typically, $Cy5$ represents treated/infected samples and $Cy3$ represents untreated/uninfected samples. Thus, for example, if the log ratio is 2.0 for a given condition, then this means the gene is over-expressed by 2 fold, and is usually represented with a red light indicator in the visual output for that spot from the microarray chip. Vise versa, if the log ratio is -2.0, then this means the gene is under-expressed by 2 fold, and is usually represented with a green light indicator in the visual output for that spot from the microarray chip. Therefore, the expression values will be clustered by trends over a period of time and by fold regulation [13].

3.1 Data Normalization and Tools

We implemented a normalization technique based on fluorescence intensities. This is a popular method based on total intensity normalization, where each fluorescent intensity value is divided by the sum of all the fluorescent intensities [14].

The normalization, cleaning, and analysis of the data take place in Oracle 10*i*. Oracle10*i* Data Mining simplifies the process of normalizing and extracting intelligence from large amounts of data. It eliminates off-loading vast quantities of data to external special-purpose analytic servers for data mining and scoring. With Oracle 10*i* Data Mining, all the data mining functionality is embedded in Oracle10*i* Database, so the data, data preparation, model building, and model scoring activities remain in the database. Because Oracle 10*i* Data Mining performs all phases of data mining within the database, each data mining phase results in significant improvements in productivity, automation, and integration. Significant productivity enhancements are achieved by eliminating the extraction of data from the database to special purpose data mining tools and the importing of the data mining results back into the database. These improvements are notable in data preparation, which often can constitute as much as 80% of the data mining process. With Oracle 10*i* Data Mining, all the data preparation can be performed using standard SQL manipulation and data mining utilities within Oracle9*i* Data Mining [15].

3.2 Preprocessing

We performed hierarchical clustering, p-clustering, and plotting analysis on mRNA expressions of approximately 6800 genes using the cDNA microarray dataset generated from the University of California, San Diego [5]. It is important to note the difference between p-clustering and subspace clustering. These mRNA expressions were monitored at eight time points [0.5h, 2h, 4h, 8h, 16h, 24h, 48h, 72h] from a CD4+ T-cell line (CEM-GFP) during HIV-1 infection. The CEM-GFP cells were inoculated with HIV-1 at a multiplicity of infection of 0.5, an inoculum sufficient to ensure that every cell is contracted by virus particles. Aliquots of cells were obtained as described above. A mock infection served as a control at each time point, essentially replacing the volume of viral input by an equivalent volume of culture medium from uninfected cells. Each sample was tested on two chips and the average was taken. Normalization for this dataset was done using global normalization and scaling. Other cleaning techniques were applied to the dataset, as described below:

1. % Present >= X. This removes all genes that have missing values in greater than (100 - *X*) percent of the columns. In our case, X was 90.
2. SD (Gene Vector) >= X. This removed all genes that have standard deviations of observed values less than *X*. In our case, X was 2.0.
3. At least X Observations abs(Val) >= Y. This removes all genes that do not have at least X observations with absolute values greater than Y. We require at least 8 observations with absolute value greater than 2.0.
4. MaxVal-MinVal >= X. This removes all genes whose maximum minus minimum values are less than *X*. In our case, X was 2.0.

For cleaning technique 1, we set X = 90 because if a gene had a missing value for just one column, this would be very significant since there are only eight time points. So, by setting 90 as a threshold, we select only the genes with values for all columns, which leads to more accurate data analysis.

For cleaning technique 2, X=2.0 because in order to do fairly accurate data analysis, the gene expression values should not be too small. Otherwise, results could be skewed. Thus, 2.0 would serve as a fair standard deviation tolerance to delete genes that could potentially affect the final results.

For cleaning technique 3, again, to avoid skewing of the results because of the gene expression values being too small, we made sure every gene included in the analysis had values greater than 2 for each and every time point.

For cleaning technique 4, it was more efficient to delete genes that would be of no significance for the analysis. Setting X=2 as the difference between the maximum and minimum values was an easy way to dismiss genes (less than or equal to X) that were of no significance.

After normalization and cleaning of the data, 167 genes out of 6823 genes (2.5%) were deleted from the dataset. Then, the data was organized into two smaller datasets for analysis. The first dataset was the mock infection and the second dataset was the actual infection.

3.3 Analysis and Results

Discovering co-expressed genes and coherent expression patterns in gene expression data is an important data analysis task in bioinformatics research and biomedical applications. It is often an important task to identify the co-expressed genes and the coherent expression patterns from the gene expression data. A group of *co-expressed genes* are the ones with similar expression profiles, while a *coherent expression pattern* characterizes the common trend of expression levels for a group of co-expressed genes. In practice, co-expressed genes may belong to the same or similar functional categories and indicate co-regulated families. Coherent expression patterns may characterize important cellular processes and suggest the regulating mechanism in the cells [16].

To find co-expressed genes and discover coherent expression patterns, many gene clustering methods have been proposed [12]. In our case, each cluster was considered as a group of co-expressed genes. The coherent expression pattern was identified via a comparative analysis of the percentage increase/decrease of each gene. Finally, the mean (or centroid) of the expression profiles of the genes in the resulting sub-clusters gives the corresponding coherent expression pattern. While clustering algorithms have been shown useful to identify co-expressed gene groups and discover coherent expression patterns, due to the specific characteristics of gene expression data and the special requirements from the biology domain, several great challenges for clustering gene expression data remain [17].

An interesting phenomenon in gene expression data sets is that *groups of co-expressed genes may be highly connected by a large amount of "intermediate" genes*. Technically, two genes g_x and g_y that have very different expression profiles in a data set may be bridged by a series of intermediate genes such that each two consecutive genes on the bridge have similar profiles. An empirical study has shown that such "bridges" are common in gene expression data sets. The high connectivity in the gene expression data raises a challenge: *It is often hard to find the (clear) borders among the clusters*. Many existing clustering methods use one of the following two strategies. On the one hand, the data set is decomposed into numerous small clusters. While some clusters consist of groups of biologically meaningful co-expressed genes, many clusters may consist of only intermediate genes. Since there is no biologically meaningful criteria (e.g., size, compactness) to rank the resulted clusters, it may take a lot of effort to examine which clusters are meaningful groups of co-expressed genes. On the other hand, an algorithm may form several large clusters. Each cluster contains both the co-expressed genes and a large amount of intermediate genes. However, those intermediate genes may mislead the centroids of the clusters into going astray. The centroids then no longer represent the true coherent patterns in the groups of co-expressed genes [17].

In a gene expression data set, there are usually multiple groups of co-expressed genes as well as the corresponding coherent patterns. Moreover, there is typically a hierarchy of co-expressed genes and coherent patterns in a gene expression data set. At the high levels of the hierarchy, large groups of genes approximately follow some "rough" coherent expression patterns. At the low levels of the hierarchy, the large groups of genes break into smaller subgroups. Those smaller groups of co-expressed genes follow some "fine" coherent expression patterns, which inherit some characteristics from the "rough" patterns, and add some distinct characteristics [17].

In our analysis, after cleaning the data, we proceeded to use an agglomerative hierarchical clustering approach based on average linkage [16] to hierarchically cluster the genes. Then we examined the clustered results and identified a cross-sectional point to start the coherent analysis. The cross-sectional point was three levels in from the root level. This level was chosen because it was the last level that had sibling nodes that covered all the genes analyzed from the microarray. This approach proved to be more effective and accurate than just simply taking the mean of each hierarchical cluster because not every gene which displays a similar pattern is necessary similar in function.

At that point, we developed and implemented an algorithm similar to the p-clustering concept. When examining all the sibling nodes (starting at 3 levels in), we computed the percentage increase/decrease between adjacent time points for each gene in each of the sibling nodes, and computationally compared such percentage variations for all the genes in that cluster. Using a 10% dis-similarity tolerance between the percentages, we were able to computationally reclassify the genes into sub-clusters based on pattern similarity. More formally, we can represent a gene as a eight dimensional vector. Let $g_x = (g_{x1}, \cdots, g_{x8}), g_y = (g_{y1}, \cdots, g_{y8})$ be such two gene vectors. We define the *pSiminarity* between the *ith* and *(i+1)th* components of two genes g_x and g_y as follows:

$$pSimilarity(g_x, g_y, i) \cong$$
$$\left|(((g_{xi+1} - g_{xi}) / g_{xi}) \times 100) - (((g_{yi+1} - g_{yi}) / g_{yi}) \times 100)\right|$$

The above equation computes the (absolute value of the) difference between the percentage decrease/increase between the corresponding sequential time points of two genes g_x and g_y. Genes that are under or equal to a 10 percent dissimilarity for all 7 (8 time points) percentages are clustered in the same sub-group. That is:

$$g_x, g_y \in same \;\; cluster$$
$$if \quad pSimilarity(g_x, g_y, i) \le 10 \quad \forall i = 1, \cdots, 7$$

In the example below, g_x is constant through out the loop and g_y represents the gene that is being compared to g_x from the same hierarchical cluster at level 3. Thus, the loop continues until all genes from that cluster is computationally compared to gene g_x.

```
1oopcount = 1
While (loopcount <= X)     //X = the number of genes in the given hierarchical cluster at level 3
{
        if
($\left|(((g_{xi+1} - g_{xi}) / g_{xi}) \times 100) - (((g_{yi+1} - g_{yi}) / g_{yi}) \times 100)\right| \le 10$)
$\forall i = 1, \cdots, 7$

        then    cluster=true;

        else    cluster=false;
loopcount = loopcount + 1;
}
```

After g_x was compared, and all similar genes were clustered with g_x, the next non-clustered gene replaced g_x and was compared to all other non-clustered genes. The loopcount was also modified to the number of non-cluster genes left. This cycle continued until all genes belonged to disjoint clusters. For clusters that visually displayed ‘rough’ patterns (i.e., when the majority of genes in the cluster were close to the 10% dissimilarity threshold), we re-ran the algorithm to generate more ‘fine’ sub-clusters using a higher degree for the tolerance (i.e. 5%). Once all the ‘rough’ patterns were refined, we took the average for each time point for all the genes in each cluster to represent the pattern trend for that cluster. Thus, when each cluster was plotted, it was very easy to decipher which clusters had potential genetic markers for HIV-1/AIDS because they exhibited sharp pattern trends.

After identifying a set of genes as potential genetic markers from the lower level clusters, we traced them back to the original dendogram to see if they were similar based on expression profiles, which would indicate similar functionality of these genes as well. We also used the public genome database to help confirm the results, which are discussed below.

From the analysis, we were able to single out individual genes that would serve as potential genetic markers by breaking down the clusters into smaller sub-clusters using the algorithm described. The reason is that we were strictly looking for genetic markers as in genes that show a significant, constant change in their expression profile when exposed to the virus. Whether this behavior was triggered by other genes is irrelevant because we are not looking for a deep understanding of the gene other than knowing at a basic level why the gene could have been affected. The use of the public genome database is a sure way of confirming the results. The accession number for the first gene is *J04423*. Because this gene was of high interest during the microarray experiment, six different probe sets were used with each resulting in a significant fold regulation by 72 hours. The probe that yielded the highest fold increase had an upfold regulation of 1.85 ($\log_2$ (25448.1/7187.9)) at 72 hours. The next gene - accession number *XO3453* - was analyzed with two different probe sets. The probe that yielded the highest fold regulation had an upfold regulation of 1.55 ($\log_2$ (65440.2/22487.1)) at 72 hours. The other four genes (accession numbers stated below) of interest were only analyzed using one probe set and yielded the following results:

- *U14573*: upfold regulation of 1.5 (log$_2$ (95340.6/34555.2)) at 72 hours
- *AB000905*: upfold regulation of 1.5 (log$_2$ (210.2.9/76)) at 72 hours
- *D43951*: upfold regulation of 2.45 (log$_2$ (111.6/20.7)) at 72 hours
- *M21388*: upfold regulation of 1.5 (log$_2$ (28749.2/10162.9)) at 72 hours

In Figures 1-6, the pink line represents infected CEM-GFP cells, while the blue line represents non-infected CEM-GFP cells. The graphs show the expression value for each time point and the over all pattern for all the time points for the given gene.

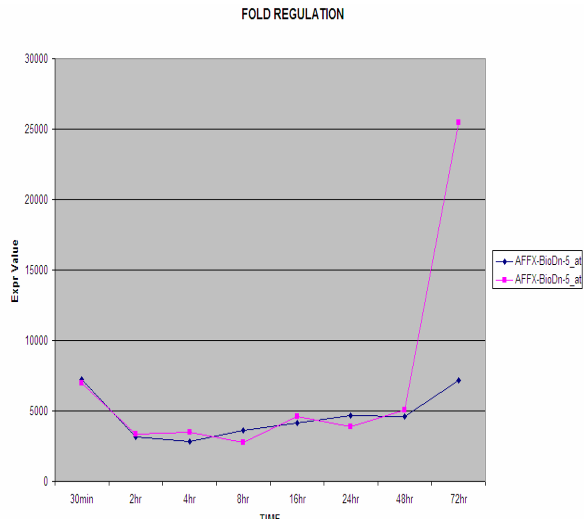

Figure 1: *J04423*

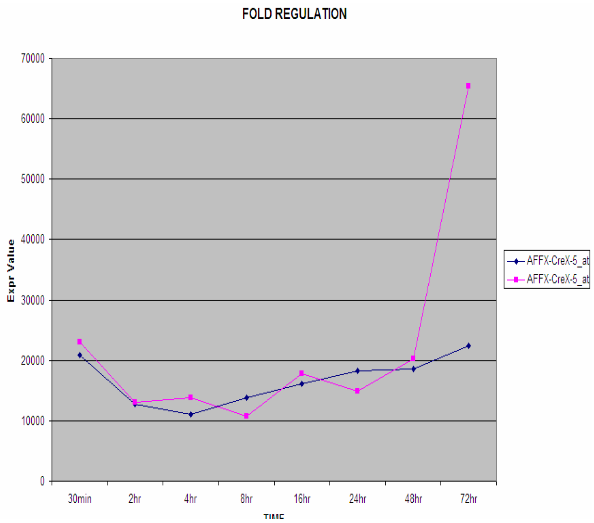

Figure2: *XO3453*

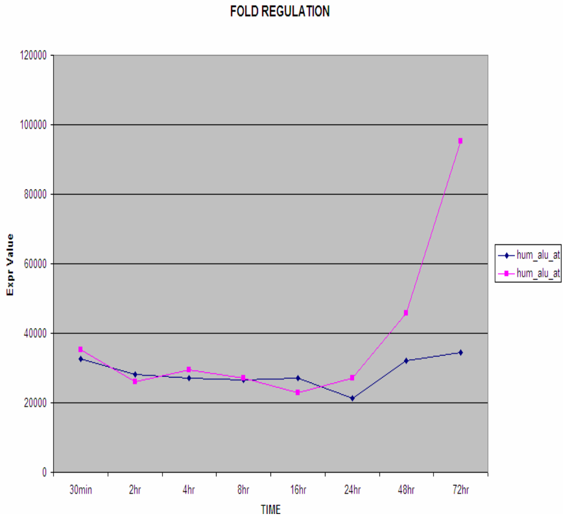

Figure 3: *U14573*

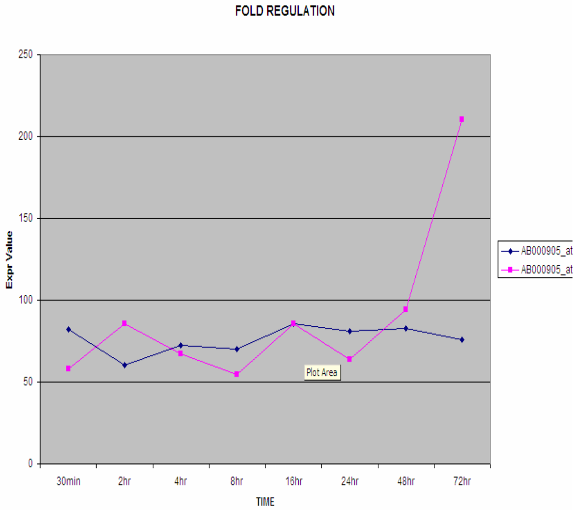

Figure 4: *AB000905*

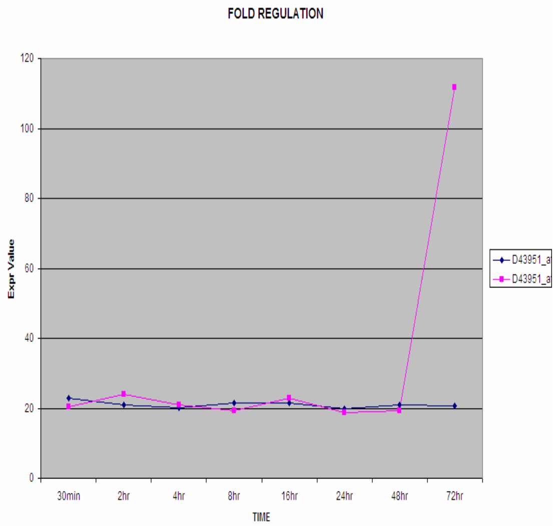

Figure 5: *D43951*

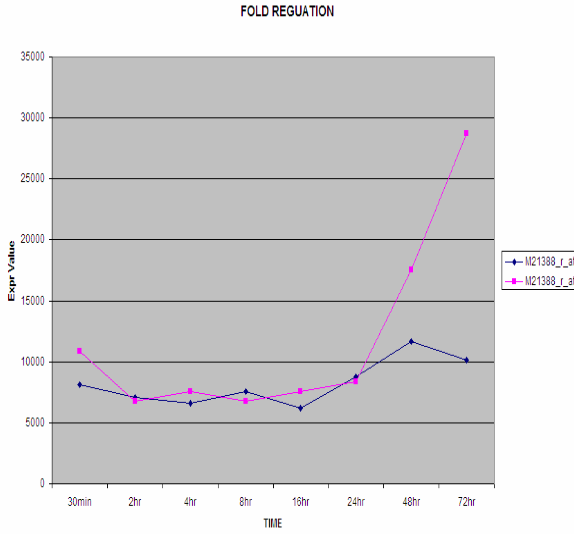

Figure 6: *M21388*

Thus, using 1.5 increase or decrease fold regulation as the cut-off between 48 hours and 72 hours, we obtain 6 different genes that we can use as potential genetic markers. We choose to pay close attention to Day 2 and Day 3 because previous published research has indicated that drastic changes in gene expression profiles for infected HIV genes occur after 48 hours [8]. Thus, the accession numbers for these genes are:

1. J04423 with AFFX-BioDn-5_at as the probe set
2. X03453 with AFFX-CreX-5_at as the probe set
3. U14573 with hum_alu_at as the probe set
4. AB000905 with AB000905_at as the probe set
5. D43951 with D43951_at as the probe set
6. M21388 with M21388_r_at as the probe set

From looking up the 6 different genes in the GenBank and NCBI databases, we were able to confirm the results as shown in Table 1 [17]:

Accession Number	Gene	Gene Type	Gene Product
J04423	bioD	Protein Coding	enzyme called dethiobiotin synthetase
X03453	cre	Protein Coding	Enzyme called cyclization recombinase
U14573	Alu	Protein Coding	actively transcribed by pol III, altered protein sequences
AB000905	HIST1H4I	Protein Coding	histone 1, H4i
D43951	PUM1	Protein Coding	Assist in RNA binding and mRNA metabolism
M21388	GLA	Protein Coding	Enzyme called alpha-galactosidase

Table 1: Potential genetic HIV-1 markers and their confirmed functionality

Although some of these genes belong to different chromosomes, we can infer that they are affected in a similar fashion when exposed to HIV-1 virus after 3 days. Therefore, we can see why it is important to not only look for co-expressed genes, but also for coherent genes in order to obtain a full snap shot of the gene's profile.

4. Conclusions

All of the gene products listed in the given table are highly affected by the HIV-1 virus. However, to really confirm whether these genes can be used as genetic markers in real life, *in-vivo* samples should be tested as well to help confirm these results. This is because *in-vivo* samples come directly from the individual and not post-infected outside the body. *In-vivo* samples from the different stages of HIV/AIDS should also be used.

Overall, the results presented in this paper are promising, and provide a good starting point for further research in this area. This research can contribute to the HIV Pharmacogenomics field by confirming HIV genetic markers, which would lead to rapid diagnosis and customized treatments. In fact, doctors can easily use these markers, along with other markers for other diseases, to rapidly diagnose a patient's profile in one genetic scan. At the same time, these markers can be used to monitor the progression or treatment of the disease.

Acknowledgements

C. Domeniconi is supported in part by the NSF CAREER Award IIS-0447814.

References

1. *AIDS Epidemic Update*, report, UN AIDS, December 2000.
2. Holodniy, M., Kuritzkes, D.R., Byer, D, Murray, P. "HIV viral load markers in clinical practice." Nature Medicine. Volume 2, pp.625-629, 1996.
3. Bumgarner, E., Geiss, G.K., V'houte, D., Haglin, J. "Large scale Monitoring of Host Cell Gene Expression during HIV-1 infection Using cDNA Microarrays." Acedemic Press. December 1999.
4. Conrad, J. Impact of Pharmacogenomics on FDA's Drug Review Process, SACGHS Meeting, Washington, DC, October 22, 2003.
5. Corbeil, J., Genini, D., Sheeter,D. "Temporal Gene Regulation During HIV-1 Infection of Human CD4+ T Cells." Genome Research. 2 April, 2001.
5. Weiner, M.P., Hudson, T.J. "Introduction to SNPs: Discovery of Markers for Disease." Biotechniques. Volume 32, pp. s5-s32, 2002.
6. Gary K. Geiss, G.K., Hammand, D. "Pathogenesis (HIV): Virus can alter the way genes function within days of exposure." Virology. Volume 46, pp. 23-27, 2000.
7. University of Tokyo Japan Laboratory of DNA Information Analysis of Human Genome Center, Institute of Medical Science. Distance/Similarity measures, 2002.
8. Fugen, L., Stormo, G. "Selection of optimal DNA oligos for gene expression arrays." Bioinformatics. Volume 17(11), pp. 1067-1079, 2001.
9. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., "Cluster analysis and display of genome-wide expression patterns". Proceedings of the National Academy of Science USA, 95 14863-14868, December 1998
10. Luo, F., Khan, L. "Hierarchical Clustering of Gene Expression Data", Department of Computer Science, University of Texas, Dallas. March 2003.
11. Yeung, K.Y., Jung, L. "Model-Based Clustering and Data Transformations for Gene Expression Data". The Third Georgia Tech-Emory International Conference on Bioinformatics. 2001.
12. Jiang, D., Zhang, X., Pei, J. "Interactive exploration of coherent patterns in time-series gene expression data." In proceedings of the ninth ACM SIGKDD International Conference of Knowledge Discovery and Data Mining (KDD '03), Washington, DC, USA, August 24-27, 2003.
13. Kano, M., Kashima, H., Slyder, E. "A method for Normalization of Gene Expression Data." Genome Informatics. Volume 14, pp. 336-337, 2003.
14. Oracle Data Mining Technical White Paper. Oracle Corporation. December 2002.
15. Tavazoie S., Hughes D., Campbell M., Cho R.J. Church G. Systematic determination of genetic network architecture. *Nature Genet*, pages 281–285, 1999.
16. Jiang, D., Pei, J., Zhang, A. Towards Interactive Exploration of Gene Expression Patterns. State University of New York at Buffalo, 2002.
17. Rahmann, S. Rapid Large-scale oligonucleotide selection for microarrays. WABI, 2002.