University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Masters Theses

Graduate School

12-2004

Maximal Clique Enumeration and Related Tools for Microarray Data Analysis

Nicole E. Baldwin
University of Tennessee, Knoxville

Follow this and additional works at: https://trace.tennessee.edu/utk_gradthes

Part of the Computer Sciences Commons

Recommended Citation

Baldwin, Nicole E., "Maximal Clique Enumeration and Related Tools for Microarray Data Analysis. " Master's Thesis, University of Tennessee, 2004.
https://trace.tennessee.edu/utk_gradthes/4654

This Thesis is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Masters Theses by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a thesis written by Nicole E. Baldwin entitled "Maximal Clique Enumeration and Related Tools for Microarray Data Analysis." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Michael Langston, Major Professor

We have read this thesis and recommend its acceptance:

David Straight, Jian Huang

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Nicole E. Baldwin entitled "Maximal Clique Enumeration and Related Tools for Microarray Data Analysis." I have examined the final paper copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Michael Langston, Major Professor

We have read this thesis
and recommend its acceptance:

David Straight

Jian Huang

Acceptance for the Council:

Vice Chancellor and Dean of Graduate Studies

Thesis
2004
.B265

MAXIMAL CLIQUE ENUMERATION AND RELATED TOOLS FOR MICROARRAY DATA ANALYSIS

A Thesis
Presented for the
Master's of Science
Degree
The University of Tennessee, Knoxville

Nicole E. Baldwin
December, 2004

DEDICATION

This dissertation is dedicated to my parents, Barbara and Darrell Baldwin. Hopefully, it will serve as evidence that yes, there *is* an end to the research! They have gone above and beyond, supporting me through not one, but two graduate experiences. There is no greater measure of love.

No, really. I mean it.

ACKNOWLEDGEMENTS

I would like to acknowledge all those who helped me complete my Master's thesis. I thank my major advisor, Dr. Langston for allowing me to be the interface between molecular biology and computer science, and his students, particularly Faisal AbuKhzam, Chris Symons, Lan Lin, and Xinxia Peng, with whom I worked most closely over the last year-and-a-half. I would also like to thank Dr. David Straight for both serving on my committee and helping me juggle both a Research Assistantship and a Teaching Assistantship. Finally, I'd like to thank Dr. Jian Huang, my third committee member. I wish that I had time to explore graph visualization and work with you.

ABSTRACT

The purpose of this study was to investigate the utility of exact maximal clique enumeration in DNA microarray analysis, to analyze and improve upon existing exact maximal clique enumeration algorithms, and to develop new clique-based algorithms to assist in the analysis as indicated during the course of the study. As a first test, microarray data sets comprised of pre-classified human lung tissue samples were obtained through the Critical Assessment of Microarray Data Analysis (CAMDA) conference. A combination of exact maximal clique enumeration and approximate dominating set was used to attempt to classify the samples.

In another test, maximal clique enumeration was used for a priori clustering of microarray data from *Mus musculus* (mouse). Cliques from this graph, though smaller than the anticipated groups of co-regulated genes, exhibited a high degree of overlap. Many genes within the overlap are either known or suspected to be involved in one or more gene regulatory networks.

Experimental tests of four exact maximal clique enumeration algorithms on graphs derived from *Mus musculus* data normalized by either RMA or MAS 5.0 software were performed. A branch and bound Bron and Kerbosch algorithm was shown to perform the best on the widest range of inputs. A base Bron and Kerbosch algorithm was faster on very sparse graphs, but slowed considerably as edge density increased. Both the Kose and greedy algorithms were significantly slower than both Bron and Kerbosch algorithms on all inputs.

Means to improve further the branch and bound Bron and Kerbosch algorithm were then considered. Two preprocessing rules and more exacting bounds were added to the algorithm both together and separately. The low degree preprocessing rule was found to improve performance most consistently, though significant improvement was only observed with the sparsest graphs, where improvement is least necessary.

Finally, a first attempt at developing an algorithm that would integrate genes that were likely excluded from a clique as a result of noise into the appropriate group was made. Initial testing of the resulting paraclique algorithm revealed that the algorithm

maintains the desired high level of inter-group edge density while expanding the core clique to a more acceptable size. Research in this area is ongoing.

TABLE OF CONTENTS

CHAPTER 1

A COMBINATORIAL APPROACH TO THE ANALYSIS OF DIFFERENTIAL GENE EXPRESSION DATA: THE USE OF GRAPH ALGORITHMS FOR DISEASE PREDICTION AND SCREENING **1**

INTRODUCTION 1
DATASETS EMPLOYED 2
A CLIQUE-BASED STRATEGY 3
The Clique Problem *3*
Scoring Method *4*
REFINEMENT VIA DOMINATING SET 7
The Dominating Set Problem *7*
Scoring Method *8*
RESULTS 9
Experiment One *10*
Experiment Two *12*
Experiment Three *12*
CONCLUSIONS 17

CHAPTER 2

APPLYING MAXIMAL CLIQUE ENUMERATION TO ELUCIDATING GENE REGULATORY NETWORKS **19**

INTRODUCTION 19
Rationale for the Use of Maximal Clique Enumeration *20*
EXPERIMENTAL DESIGN 20
GRAPH GENERATION 21
RESULTS 21
CONCLUSION 26

CHAPTER 3

EXPERIMENTAL ANALYSIS OF EXISTING MAXIMAL CLIQUE ENUMERATION ALGORITHMS **28**

INTRODUCTION 28
DESCRIPTION OF ALGORITHMS 28
Base Bron and Kerbosch Algorithm *28*
Bron and Kerbosch Algorithm *30*
A Constructive Algorithm *32*
A Greedy Algorithm *34*
METHODS EMPLOYED 35
RESULTS 37
CONCLUSIONS 40

CHAPTER 4

ALGORITHM DEVELOPMENT FOR APPLICATION TO DNA MICROARRAY ANALYSIS ... 41

INTRODUCTION 41
RESULTS 41
Enumeration Adaptations *41*
Noise Compensation *44*
Paraclique *45*
CONCLUSIONS 46

LIST OF REFERENCES 49

VITA 54

LIST OF TABLES

Table 3-1. Graph Edge Densities ____ *36*

Table 3-2. Time Trials for Maximal Clique Enumeration on RMA Microarray Data ____ *38*

Table 3-3. Time Trials for Maximal Clique Enumeration on MAS 5.0 Microarray Data ____ *39*

Table 4-1. Time Trials for Preprocessing and Boundary Rules on MAS 5.0 Microarray Data ____ *43*

Table 4-2. Comparison of Paraclique and 1-Neighborhood. ____ *47*

List of Figures

Figure 1-1. Weights between sample pairs using 105 genes from the Michigan dataset. 6
Figure 1-2. Weights between sample pairs using 78 genes (Harvard data)..... 11
Figure 1-3. Unweighted graph of the Harvard data set resulting from a threshold of 7.9. 13
Figure 1-4. Clique frequency distribution from Harvard data set. 14
Figure 1-5. Unweighted graph of the Michigan data set resulting from a threshold of 8.7. 15
Figure 1-6. Clique distribution from Michigan data set using 109 genes and a threshold of 8.7. 16
Figure 2-1. Transformation from normalized data to unweighted adjacency matrix..... 22
Figure 2-2. Degree histogram of 0.85 threshold MAS 5.0 graph. 23
Figure 2-3. Histogram of clique sizes for 0.85 threshold MAS 5.0 graph. 24
Figure 2-4. Clique intersection graph for 0.85 threshold MAS 5.0 graph. 25
Figure 2-5. Representative clique containing veli3 (lin7c). 27
Figure 3-1. Edge weight histogram of MAS 5.0 and RMA derived graphs. 29
Figure 3-2. Pseudocode for base Bron and Kerbosch algorithm. 31
Figure 3-3. Any k-clique is comprised of two (k-1)-cliques sharing k-2 vertices. 33

Chapter 1

A Combinatorial Approach to the Analysis of Differential Gene Expression Data: The Use of Graph Algorithms for Disease Prediction and Screening

This chapter is a revised form of another paper published under the same name in *Methods of Microarray Data Analysis IV, Papers from CAMDA '03* in 2003 by Michael A. Langston, Lan Lin, Xinxia Peng, Nicole E. Baldwin, Chris T. Symons, Bing Zhang, and Jay R. Snoddy:

M. A. Langston, L. Lin, X. Peng, N. E. Baldwin , C. T. Symons, B. Zhang, and J. R. Snoddy. A Combinatorial Approach to the Analysis of Differential Gene Expression Data: The Use of Graph Algorithms for Disease Prediction and Screening. *Methods of Microarray Data Analysis IV, Papers from CAMDA '03.*

My use of "we" in this chapter refers to my co-authors and myself. My primary contributions to this paper include (1) researching, coding, and running the maximal clique enumeration algorithm, (2) assisting with developing both the maximal clique-based and the dominating set-based procedures, (3) assisting with the development of both weighting and scoring functions, (4) elucidating the rationale for utilizing clique based clustering as opposed to currently popular methods, (5) interpreting results from a biological viewpoint, (6) helping to pull the individual sections into an integrated chapter, (7) performing a significant portion of the writing and figure creation, (8) editing between chapter submission and publication, and (9) presenting the work at the Critical Assessment of Microarray Data Analysis (CAMDA) conference.

Introduction

A fundamental problem in cancer treatment is early and reliable detection. Identification of a set of genes whose expression levels serve as an accurate discriminator among normal and cancerous tissue samples would not only represent significant

progress towards developing more reliable cancer diagnosis protocols, but might also identify novel therapeutic targets. With this motivation in mind, we investigated the hypothesis that only a modest number of genes may suffice for this task. We sought to develop algorithms and software for this purpose, and introduce a graph theoretical method of differential gene expression analysis. The goals of this method were to identify a set of genes useful in discriminating among tissue samples, and to use these genes in disease prediction and screening.

One of the important features of our algorithms is the computation of discrimination scores for each gene represented in an input microarray.. These scores estimated a gene's relative ability to distinguish among sample tissue classes. We then selected the highest-scoring genes, and used them to calculate a pairwise similarity metric between patients' tissue sample expression profiles. Genes that failed to discriminate among a defined percentage of the samples were eliminated using a dominating set algorithm as a high pass filter. With this information, we constructed a complete weighted graph, in which the vertices represented the tissue samples and the edges were weighted by the similarity metric between sample vertices. A user-defined threshold was next used to transform the complete weighted graph into an incomplete unweighted graph where the weights were ignored. The combination of these tools produced some very encouraging predictive results.

Datasets Employed

We used the Harvard [Bhattacharjee et al., 2001.], Michigan [Beer et al., 2002], and Stanford [Garber et al., 2001] datasets in this study. We did not include the Ontario dataset due to a lack of overlap in annotated genes with the other datasets. Since the log-expression image plots for Samples L54, L88, L89 and L90 in the Michigan dataset showed large, round dark spots at the center of the arrays [Hu et al., 2003] indicative of poor data quality, they were removed from the dataset. This left us with 92 samples from the Michigan dataset. Because the Harvard and Michigan datasets were generated by different institutes using different Affymetrix array types (HG_U95A and HUGeneFL,

respectively), the distributions of the two datasets may not be comparable. Thus, we chose to normalize the two datasets separately. The log-scale quantifications of the gene expression levels for each probe set were obtained by robust multi-array average (RMA) [Irizarry et al., 2003.] using Bioconductor.

Since we intended to train and test our algorithms on different datasets, we needed a mapping schema among the different datasets. However, the three datasets came from different array platforms using different gene identifiers; hence, direct mapping is not possible. We chose to use LocusLink IDs (LL_IDs) for gene mapping, because the NCBI LocusLink Database is both relatively reliable and stable. For the Harvard and Michigan datasets, we mapped each probe set ID to its corresponding LL_ID using array annotation files from Affymetrix. For the Stanford dataset, we mapped each UNIGENE ID to its corresponding LL_ID using our local database, GeneKeyDB. To construct a gene expression summary for each LL_ID, we averaged the values within each sample across the original gene identifiers that map to a common LL_ID. The final datasets used in this study include: the Harvard dataset, which had expression profiles for 8509 unique genes among 254 samples; the Michigan dataset, which had expression profiles for 4985 unique genes among 92 samples; and the Stanford dataset, which had expression profiles for 8829 unique genes among 73 samples.

A Clique-Based Strategy

The Clique Problem

Clique is a well-known NP-complete problem, and is typically formulated as in Garey and Johnson [1979]:

Input: A graph $G=(V,E)$ and a positive integer $k \leq |V|$.

Question: Is there a subset $V' \subseteq V$ for which $|V'| \geq k$ and such that every pair of vertices in V' is joined by an edge in E.

Clique is rapidly becoming recognized for its relevance in bioinformatics. It can be roughly viewed as a clustering algorithm based on graph theory. In our own work, for example, we used clique in the following ways. In [Abu-Khzam et al., 2003], we devised and applied fast parallel algorithms for clique to extremely large microarray datasets in an effort to identify putatively co-regulated genes in murine neural regulatory networks. In another application [Baldwin et al., 2004], we employed high performance implementations of clique in the study of cis-regulatory elements to discover putative motifs.

Scoring Method

Our goal in training was to develop graph-theoretic tools to distinguish among sample groups (such as normal and adenocarcinoma). Ideally, we hoped to be able to construct an unweighted graph in which edges connect mainly members of the same group. At that point, clique analysis would be an attractive approach for testing our methods against additional data.

In order to pinpoint a modest number of genes out of thousands from the original dataset, our first step in training was to determine which genes appeared to discriminate best among sample types. To accomplish this, a discrimination score was calculated for each gene. Only the best genes (those with the highest scores) were retained for subsequent steps. Since the distributions of the expression values of these genes would be expected to be bimodal with respect to two distinct sample classes, the differences between class medians gave us a general measure of the difference of expression between two classes. Subtracting the sum of the standard deviations of a gene within each group allowed us to eliminate, or at least diminish, the importance of any gene whose expression levels varied excessively.

The data was obtained as an n x m matrix, A, of expression values. Rows represented test samples, and columns denoted genes. When training on the Michigan dataset in order to learn to distinguish between normal (group 1) and adenocarcinoma

(group 2) samples and using a lower limit of zero, our method delivered a collection of 105 genes for further evaluation.

An assignment of inter-sample weights can help demonstrate the degree to which these genes and their respective scores delineated normal samples from adenocarcinoma. Here, the weight between samples i and j represented the degree of similarity in their respective expression profiles and can be viewed as equivalent to the distance function for clustering. We computed this weight as a sum over all genes selected in the previous step, because it is these genes that seemed to have the greatest potential to serve as good discriminators. Accordingly, we set weight(i,j) to:

$$\sum \text{score}(\text{gene}_k) \bullet (1 - |\text{expression_value}_{ik} - \text{expression_value}_{jk}|)$$

As is shown in Figure 1-1, higher-weighted sample pairs tended to be homogenous. That is, either both tissue samples were normal or both were adenocarcinoma. Conversely, lower-weighted pairs tended to be heterogenous, where one sample was normal and the other was adenocarcinoma. While this seems to confirm our gene scoring and selection procedure, other scoring approaches appeared to be viable as well. Therefore, we investigated several other alternatives before settling on this approach.

Two of these alternative approaches were worthy of note in the computation of gene discrimination scores. One was the elimination of outliers before computing the scores, which was motivated by the fact that outliers might affect both the median and the standard deviation. The other involved changing our original scoring function to a variant of the t-test function, a standard statistical measurement of population similarity. This test was realized using division rather than subtraction within our scoring function. Neither of these appeared to improve upon our original results. We also experimented with Pearson's Correlation Coefficients and Spearman's Rank Correlation Coefficients, two popular methods of weighting. Neither of these methods was helpful. In fact, neither even revealed the bimodal distribution we observed using our weight function.

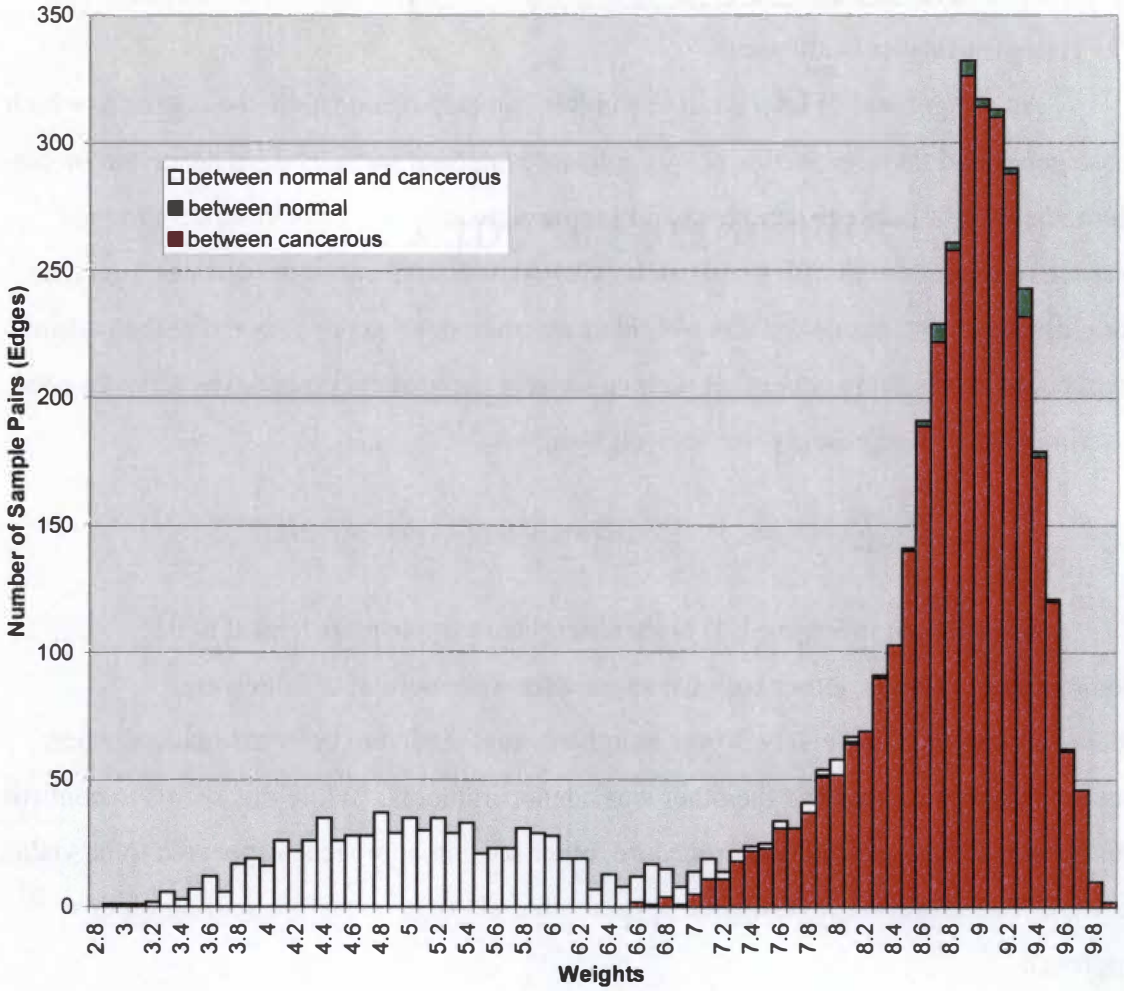

Figure 1-1. Weights between sample pairs using 105 genes from the Michigan dataset.

In addition to confirming the validity of our approach, Figure 1-1 also suggests an initial threshold weight below which we deleted edges in a later step (to be described shortly). Call this threshold T. For example, based on the figure, we chose as a somewhat informed but still rather arbitrary starting value T=7.6. We used our restricted set of genes to build an edge-weighted graph. In this graph, samples were represented by vertices and the weight of an edge between a pair of samples was set using the simple summation formula already described. Any edge whose weight was less than T was removed. The resulting unweighted graph was then searched for all maximal cliques. Our aim was to train our codes so that we can find appropriately sized cliques to cover all groups.

Because we know which samples are normal and which are adenocarcinoma in the Michigan dataset, we were able to iterate our method until we have a reasonable set of covering cliques. The optimal threshold seemed to be centered at around T=8.1. We were not completely satisfied, however, with the lingering presence of overlapping cliques. Additional experimentation with gene cutoff scores seemed to indicate that the presence of genes with low scores is problematic. However, neither raising the cutoff score nor additional modification of the threshold was of much use. What seemed to be missing in our estimates of gene discrimination was a way to determine which genes impact the greatest number of samples and to eliminate the rest. For this, we turned to another graph metric, dominating set.

Refinement Via Dominating Set

The Dominating Set Problem

Dominating Set, another well-known NP-complete problem, can be stated as follows:

Input: A graph $G=(V,E)$ and a positive integer $k \leq |V|$.

Question: Is there a subset $V' \subseteq V$ for which $|V'| \leq k$ and every vertex $v \in V - V'$ is joined to a vertex in V' by an edge in E.

Using the theory of fixed-parameter tractablility (FPT) [Downey and Fellows, 1999], dominating set may be even more difficult than clique. This is because clique is W[1]-complete and can be solved using graph complementation and vertex cover. Practical, efficient kernelization techniques are known for vertex cover [Abu-Khzam et al., 2004]. The same, however, may not hold for dominating set. In fact the dominating set version we address here is nonplanar red/blue dominating set, which is W[2]-complete. Although its complement problem is FPT, there are currently no practical kernelization techniques known for it. Thus, we only approximated solutions to dominating set.

Scoring Method

We first assumed a normal distribution of the expression values of each gene, and estimated for it the mean and standard deviation. We did this separately for each of the sample groups. Then, based on the estimated normal distribution, we calculated the p-values for the original individual expression values. It is perhaps easiest to formulate our approach by constructing a bipartite graph. In this graph, one set of vertices represented the genes, and the opposing set represented the samples. We placed an edge between a gene and a sample if and only if the p-value of the expression value corresponding to that gene-sample combination was greater than 0.05. Following statistical convention, we considered a p-value below this cutoff to indicate an outlier.

In this setting, we wanted to identify the genes that dominate (or nearly dominate) all the samples. Therefore, we winnowed out from consideration any gene vertex not adjacent to at least 90% of the sample vertices. For example, in the Michigan dataset, a gene was eliminated if it was connected to fewer than 74 of the adenocarcinoma samples or fewer than nine of the normal samples. The choice of 90% was arbitrary, but selected only after extensive testing.

Next, in an effort to remove any remaining genes with a low possibility of discriminating between the two groups, we calculated the p-values for tests of equal means using both the Wilcoxon and t-test methods. We used both since the t-test

assumes a normal distribution, while the Wilcoxon test does not. Only genes for which both p-values are less than 0.05 were retained.

For those genes that remain, we generated scores based on the previously calculated p-values from the Wilcoxon tests. We then filtered out genes using an adjusted p-value cutoff by means of the Bonferroni method. Specifically, we chose a significance level of $\alpha = 0.01$ and only kept genes with a p-value less than α/N, where N is the total number of genes we began with at this step. Since a smaller p-value indicates a greater probability that the groups' expression values are different for a given gene, we used -log10(p-value) for the gene score.

Finally, and most importantly, we computed the intersection of the genes identified by the clique-based approach described in the last section with the genes chosen by the dominating set method as described in this section. We were left with a set of genes that passed both the clique and the dominating set tests. We found that this refinement of our gene lists gave us improved results in the testing phase of our experiments.

Results

Having completed the training phase, we proceeded to testing on a new dataset under the assumption that we will not know sample classification in advance. We evaluated our approach with the following three experiments. First, we trained on the Michigan dataset as explained in section 3 in order to learn to distinguish between normal and adenocarcinoma samples. We proceeded to test our ability to classify samples on the Harvard dataset. Second, we reversed this process, applying our training algorithms to the Harvard dataset to distinguish between cancerous and normal samples and testing our method on the Michigan dataset. Third, we trained on the Harvard dataset to learn to separate adenocarcinoma from squamous samples, testing on the Stanford dataset.

Experiment One

Clique-based training on the Michigan dataset identified 105 genes that distinguished between adenocarcinoma and normal samples. Our dominating- set-based refinement reduced this to 84 genes, 78 of which are available in the Harvard data. Figure 1-2 shows the distribution of the edge-weight scores generated using these genes on the normal and adenocarcinoma samples from the Harvard dataset. If our method is to be predictive, we expected to see something of a bimodal distribution, although peak height is dependent on the relative populations of the two groups. This is because weights between members of the same group are expected to be high, while weights between members of different groups are expected to be low. Such a distribution is in fact what we observed in Figure 1-2.

We exploited this property when carrying out threshold selection. We chose an initial threshold slightly to the right of the median edge-weight value. We then enumerated all maximal cliques in the unweighted graph, and checked to see whether every sample is in at least one clique. If not, we chose lower and lower threshold values until we had full coverage (that is, until every sample was in at least one clique). If, on the other hand, our initial threshold gave us full coverage, we incrementally selected higher and higher thresholds until we generated an unweighted graph for which there was at least one sample that was missing from every maximal clique. At this point, we went back one step and used the highest threshold with full coverage. Naturally, this is only one possible method for selecting the threshold; other methods may work equally well. After a suitable threshold was determined, we analyzed the data by testing the supposition that all cliques of significant size were uniform in the sense that they contained samples from adenocarcinoma samples only or from normal samples only.

When this iterative process was carried out on the Harvard dataset without the use of any previous knowledge pertaining to its sample classifications, we were effectively able to separate the subjects into adenocarcinoma cliques and normal cliques. In fact, at our chosen threshold of 7.9, only one sample out of the 207 combined adenocarcinoma and normal samples was misclassified according to the Harvard dataset using this

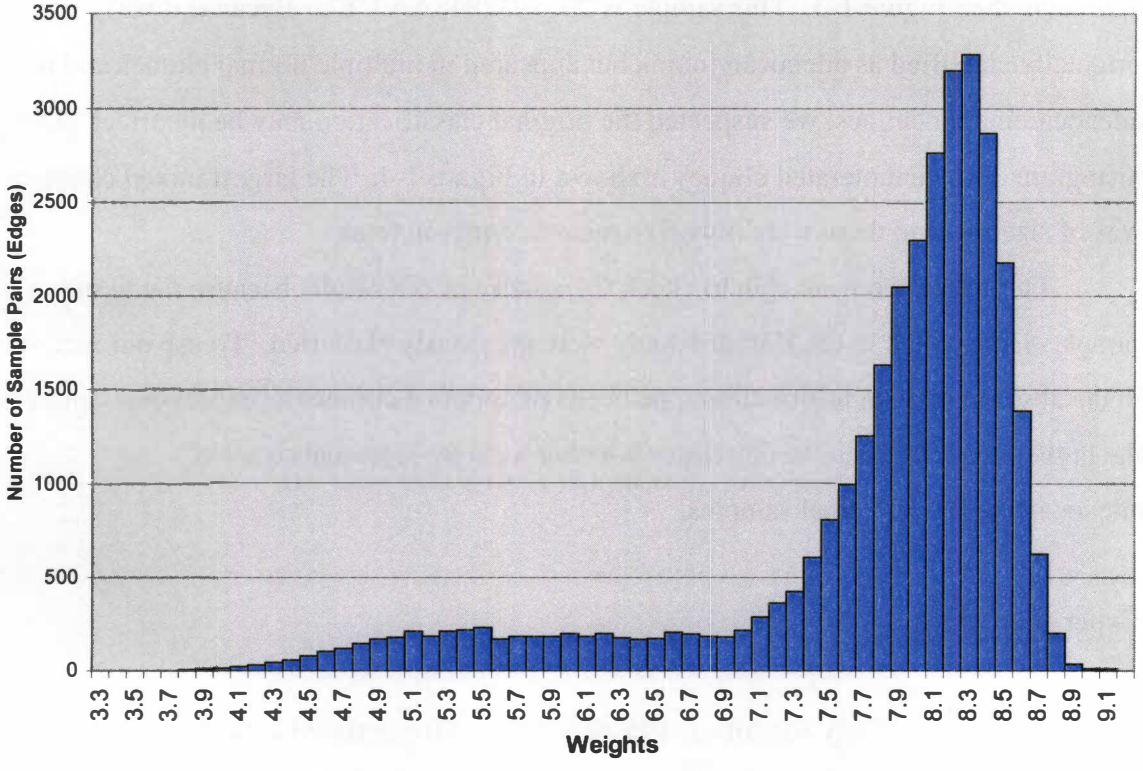

Figure 1-2. Weights between sample pairs using 78 genes (Harvard data).

approach. See Figure 1-3. This sample is 2001032848AA.CEL. Because it was originally classified as adenocarcinoma but appeared in multiple normal cliques and no adenocarcinoma cliques, we suspected the original classification may be incorrect. The histogram of the enumerated cliques is shown in Figure 1-4. The largest mixed clique was of size six, and there were only five mixed cliques in total.

Of course, we were able to check the quality of our results because the tissue samples represented in the Harvard study were previously classified. To use our methods in the absence of such information, one needs merely to examine the expression values of the highest-scoring genes to determine whether a clique represents a set of adenocarcinoma or normal samples.

Experiment Two

In this case, we initially identified 195 genes that differentiated cancerous and normal samples. This was reduced to 180 using our refinement technique, and 109 of these genes were available in the Michigan dataset.

After following the process we have detailed, we selected a threshold of 8.7. We enumerated maximal cliques on the resulting unweighted graph shown in Figure 1-5. Our methods were able to sort the samples into cancerous and normal cliques almost flawlessly. In fact, out of the 235 cliques of size 3 or greater in the resulting graph, only one clique had both cancerous and normal samples, and this was very small (size 3). The resultant frequency distribution of these cliques is depicted in Figure 1-6.

Experiment Three

Training on the Harvard dataset to discriminate between adenocarcinoma and squamous cell carcinoma initially gives us 37 genes. After refinement, 35 are left, 26 of which are found in the Stanford data set. In this case, the results given by our method are not as compelling as in the previous two experiments. By using the largest clique containing each sample, we classify 41 out of 47 samples correctly according to the

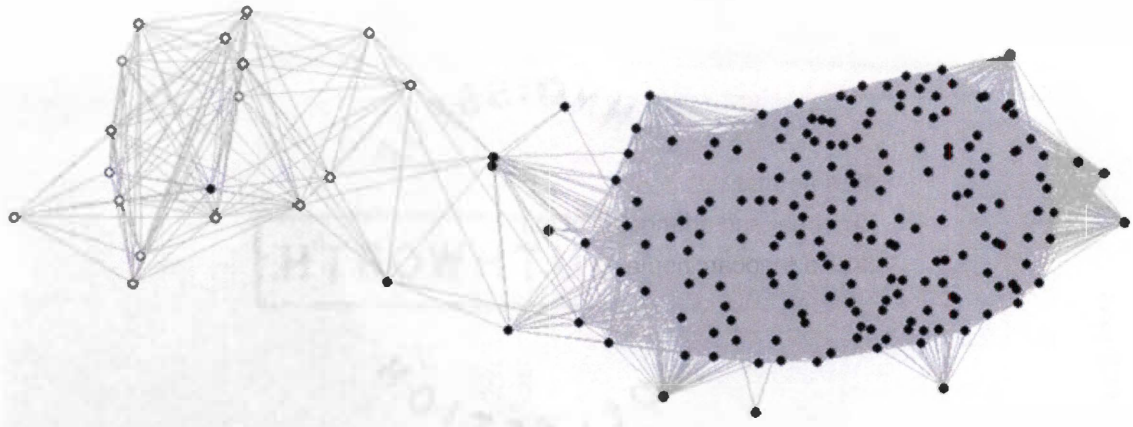

Figure 1-3. Unweighted graph of the Harvard data set resulting from a threshold of 7.9. Black vertices represent adenocarcinoma samples. White vertices represent normal samples.

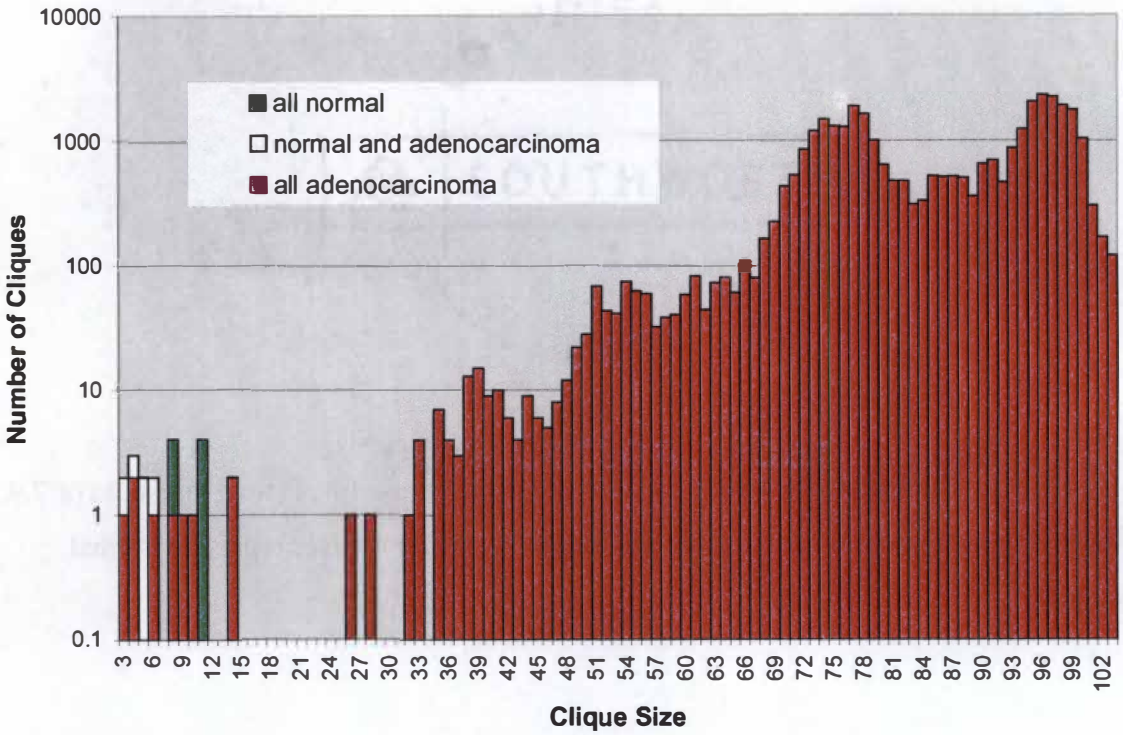

Figure 1-4. Clique frequency distribution from Harvard data set. Adenocarcinoma and normal samples were compared, using 78 genes and a threshold of 7.9.

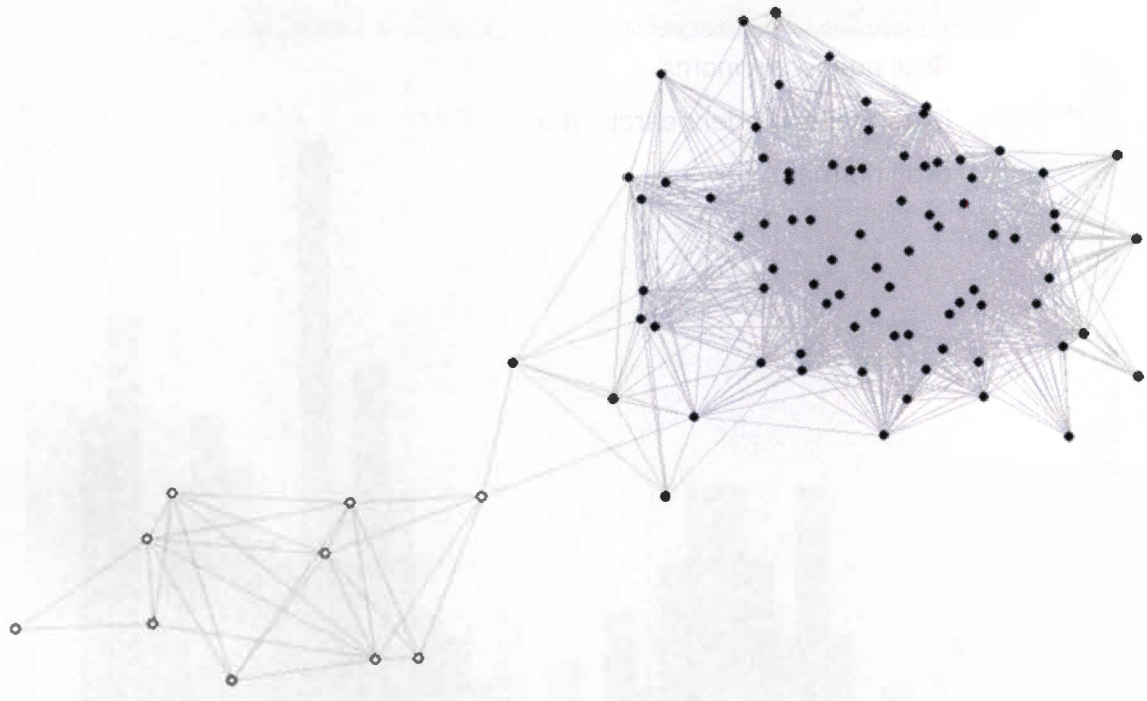

Figure 1-5. Unweighted graph of the Michigan data set resulting from a threshold of 8.7. Black vertices represent adenocarcinoma samples. White vertices represent normal samples.

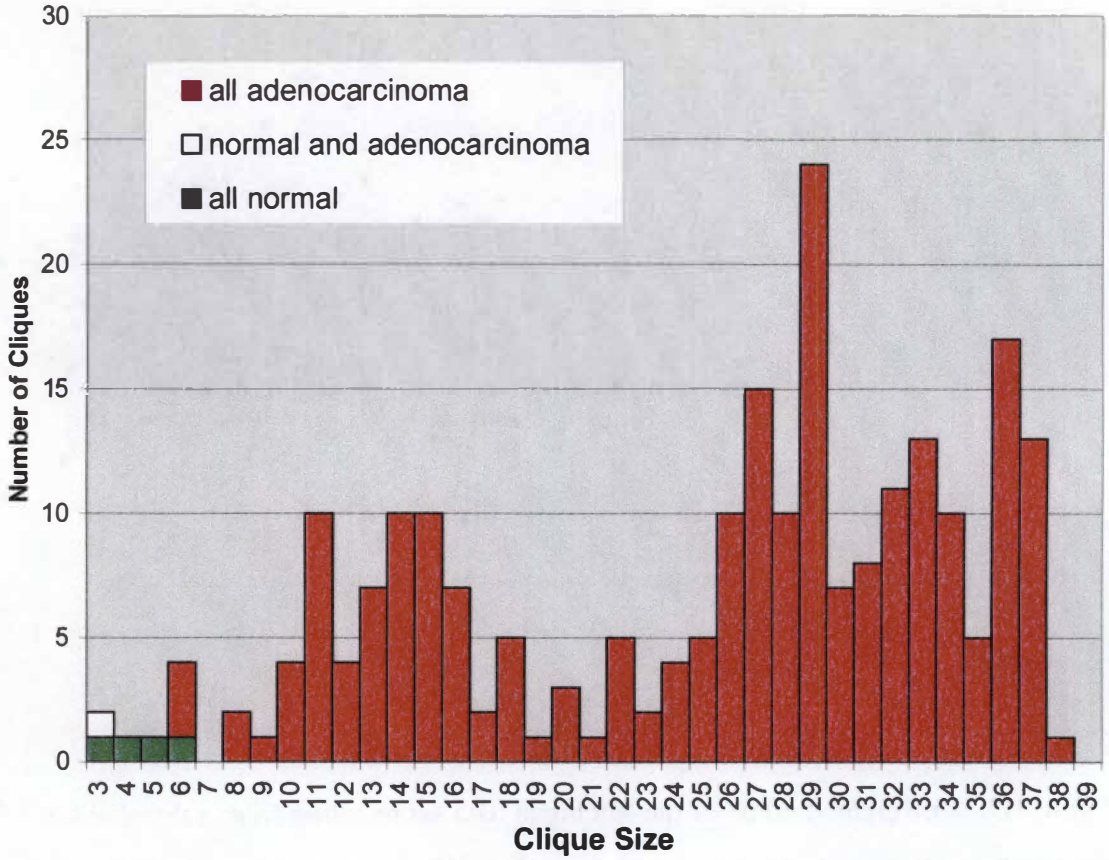

Figure 1-6. Clique distribution from Michigan data set using 109 genes and a threshold of 8.7.

Stanford classifications. Nevertheless, there were still too many mixed cliques. This was not unexpected. Our methods isolated a set of 35 genes as good discriminators. However, with only 26 of these available in the test dataset, their use provided at best a crude classification tool.

Conclusions

There is no apparent consensus as to the best approach for mining microarray data. Popular methods in current use include Bayesian analysis [Friedman et al., 2000, Sok et al., 2003], hierarchical clustering, and scale-free networks [del Rio et al., 2001], to name just a few. We believe that the novel methodology we have described here can be used to complement these techniques, and also is of independent interest. Deliverables accompanying this effort include the algorithmic framework of our overall strategy, the software tools we have developed and implemented, and of course the resultant gene sets themselves.

A key feature of our approach is the use of two distinct gene-scoring systems, each coupled with a different combinatorial algorithm. One was based on finding optimal cliques within general graphs, the other on isolating near-optimal dominating sets within bipartite graphs. Used in tandem, these algorithms appeared to provide an effective means for identifying and ranking predictive genes whose expression levels serve as an accurate discriminator between adenocarcinoma and normal tissues. We emphasize that the use of clique and dominating set together seems to produce better results than would be possible with either approach alone.

The high fidelity with which the resulting cliques partitioned cancerous and normal samples, as illustrated in Figures 1-4 and 1-6, prompts us to posit that our methodology has the potential to become the basis for a highly reliable tool for cancer prediction. No a priori knowledge of the number of classes contained in the dataset is required. Moreover, it is known that tumor tissue samples are frequently a mixture of multiple types of cells, and that the exact ratio of this mixture is not necessarily consistent, even among samples from the same tumor. Therefore, it is expected that

tissue samples might have significant similarity to more than one class, such as adenocarcinoma and normal. This is, in fact, what was observed. Using our method, the classification of the sample is not limited to one class. Nor is the classification based on the highest similarity score. Instead, it is based on a significant degree of similarity to the greatest number of samples that also are significantly similar to each other. In other words, classification is based on the largest (maximal) clique to which the sample belongs. This should result in a higher degree of confidence in our classification.

As a further proof of principle, several of the genes we identified as discriminators in the Michigan data are known or suspected to play a role in oncogenesis. Among these are: CYP4B1, a cytochrome P450 enzyme that has been implicated in both bladder and lung cancer in humans [Czerwinski et al., 1994, Imaoka et al., 2000]; FHL1, shown to have cytotoxic effects on melanoma cell lines and possibly to play a role in cellular differentiation[de Vries et al., 1975]; the p85 alpha subunit of phosphoinositide-3-kinase, which plays a role in human breast cancer [Das et al., 2003., Mahabeleshwar et al., 2003]; and tetranectin, which has already been shown to have prognosticative value for survival rates at certain stages of ovarian cancer [Hogdall et al., 2002].

Chapter 2

Applying Maximal Clique Enumeration to Elucidating Gene Regulatory Networks

Introduction

Since maximal clique enumeration proved its usefulness in analyzing microarray data for disease prediction and screening, we turned to another, more common type of analysis – that required by basic research[1]. One of the main goals of fundamental biology is to elucidate gene regulatory networks, or the collection of cellular components (genes, proteins, etc.) and their interactions that carry out a specific function. For example, one of the simplest such networks would be the less than twenty member set of genes and their products that are responsible for regulation of lactose metabolism in the bacterium *Escherichia coli* [Reznikoff, 1992]. Most networks, particularly in advanced organisms, are more extensive and can involve hundreds of genes. Until recently, available methods of investigating such networks allowed researchers to observe only a few genes at a time. With such limitations, it took decades to understand even small networks.

In order to comprehend the interactions within and among larger networks, a way to observe the actions of a large number of genes in response to any experimental stimulus was needed. This is now possible with DNA microarrays, which are capable of testing an entire genome (all genes in a cell) simultaneously. Unfortunately, it is not a simple task to interpret such a mass of information, particularly considering the noise inherent in all biological experiments, and in particular microarray experiments. A first goal in analyzing microarray data in relation to gene regulatory networks is to be able to group genes that exhibit similar responses to series of specific stimuli. This implies that the genes may be co-regulated and therefore acting within the same network.

In this case, clustering must be accomplished a priori, as typically there is insufficient knowledge about the system or systems being studied to permit a training phase. This lack of information also makes determining the correctness of the clustering

[1] All figures in this chapter are the work of the author, except where indicated.

impossible without extensive and time-consuming laboratory experiments to verify the results. Instead, clusters are used for their probative value in order to generate new hypotheses to be tested, or to evaluate those that already exist.

Rationale for the Use of Maximal Clique Enumeration

For this purpose, maximal clique enumeration has three attractive features that are lacking in other popularly used techniques such as those mentioned in Chapter 1. Firstly, cliques are, by nature, the most stringent measure of similarity possible. This affords the advantage that any genes that are a member of a clique are highly likely to be co-regulated. This level of stringency does not effectively cope with noise, but that issue can be addressed by a variety of methods, some of which will be discussed in Chapter 4. Secondly, maximal clique enumeration permits transcript membership in multiple cliques. This is a significant advantage, because it is common for a gene to participate in multiple networks. Forcing such a gene into one cluster not only loses critical information, but also has the potential to significantly skew subsequent classifications. Finally, it is not necessary to know or be able to infer the expected number of clusters, a value that is rarely available for microarray data. Supplying an incorrect value to an algorithm that required such would clearly invalidate any result.

Experimental Design

All microarray data described in this chapter was provided courtesy of collaborators Dr. Robert W. Williams and Dr. Elissa J. Chesler from the Department of Anatomy and Neurobiology of the University of Tennessee in Memphis. The Affymetrix U74Av2 array was used to test 12,422 probesets in samples from the brain of *Mus musculus* (mouse). Each sample consisted of tissue from three genetically identical mice. One sample was collected from each of three related recombinant inbred strains of mice, bred such that each strain was a genetic mosaic of the parental strains (C57BL/6J and DBA/2J). In other words, a gene in one of the recombinant inbred strains has an equal

chance of having been inherited from the C57BL/6J or DBA/2J parental strain. The difference in genetic background of each of the three recombinant inbred strains served as changing experimental conditions. In all other aspects, the samples were treated the same. The experiment was repeated three times and the data pooled.

Graph Generation

A simplified example of converting normalized data to an unweighted graph is shown in Figure 2-1 (Data and figures in this chapter, with the exception of Figures 2-1 and 2-2, are being published in Baldwin et al., In press.). Raw data from DNA microarray experiments was normalized using the MAS 5.0 (Microarray Suite) software package. Pairwise Spearman's rank coefficients were calculated, resulting in a 12,422 x 12,422 weighted adjacency matrix, where 12,422 was the number of genes measured in the microarray experiment. A threshold of 0.85 was chosen by our colleagues in neurobiology because the maximum clique size at that threshold (17) was of appropriate size. The weighted matrix was filtered using this threshold to produce an unweighted matrix where an edge (i, j) is present if and only if the absolute value of the Spearman rank coefficient for (i, j) is greater than or equal to the threshold value. A degree histogram of the resulting unweighted graph is shown in Figure 2-2.

Results

Maximal clique enumeration of the unweighted graph discussed in the previous section resulted in a total of 5,227 maximal cliques. The maximum clique size was 17, with a user-determined minimum clique size of 3. The distribution of clique sizes generated is shown in Figure 2-3. There was a tremendous amount of overlap among these cliques, as shown in the clique intersection graph in Figure 2-4. As indicated by the lack of an isolated vertex in the aforementioned graph, every clique of size 15 or greater `(179 in total) overlapped with at least one other clique by more than 76%. Additionally, a very high density region containing the three maximum cliques (shown in red) can be

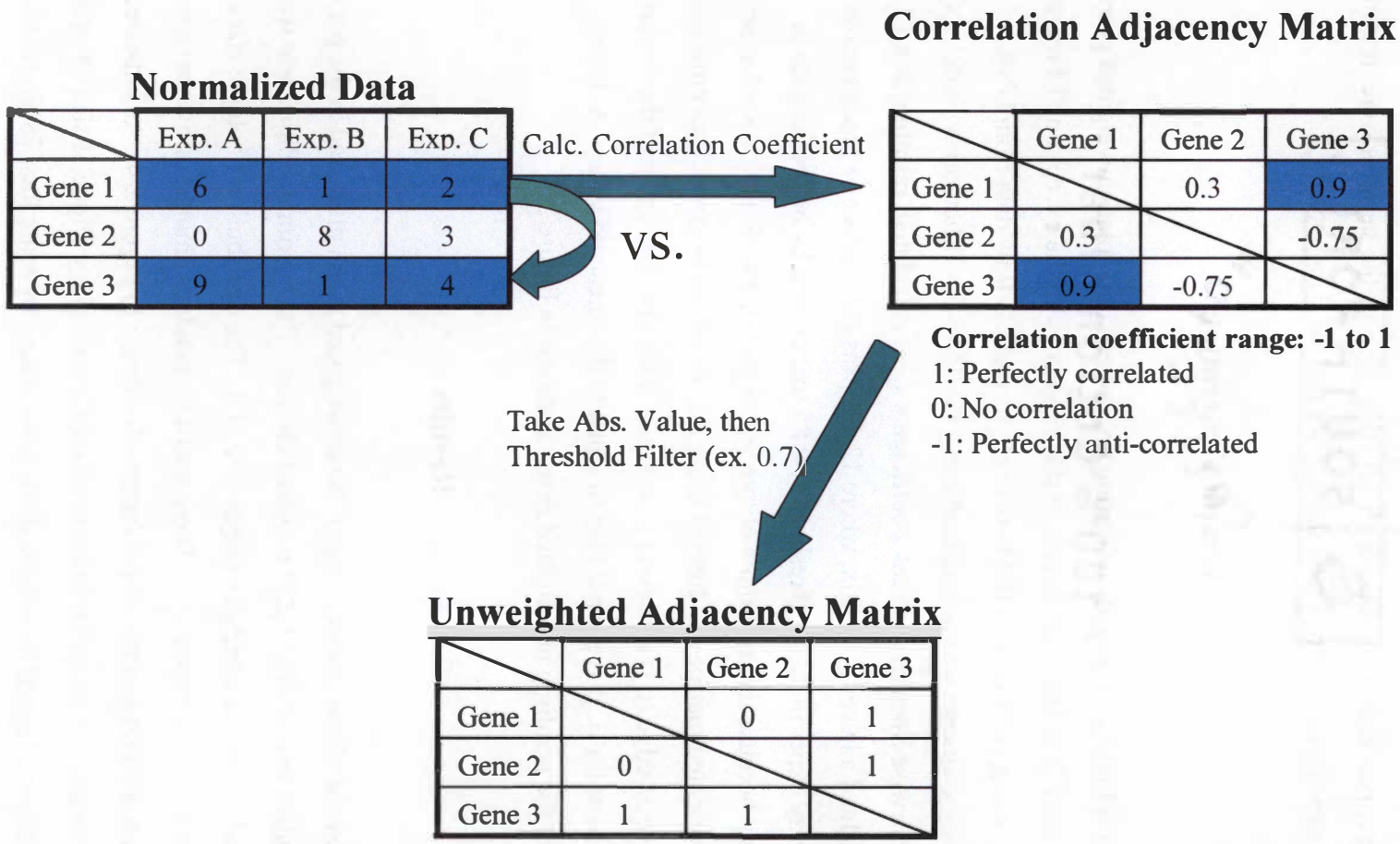

Figure 2-1. Transformation from normalized data to unweighted adjacency matrix.

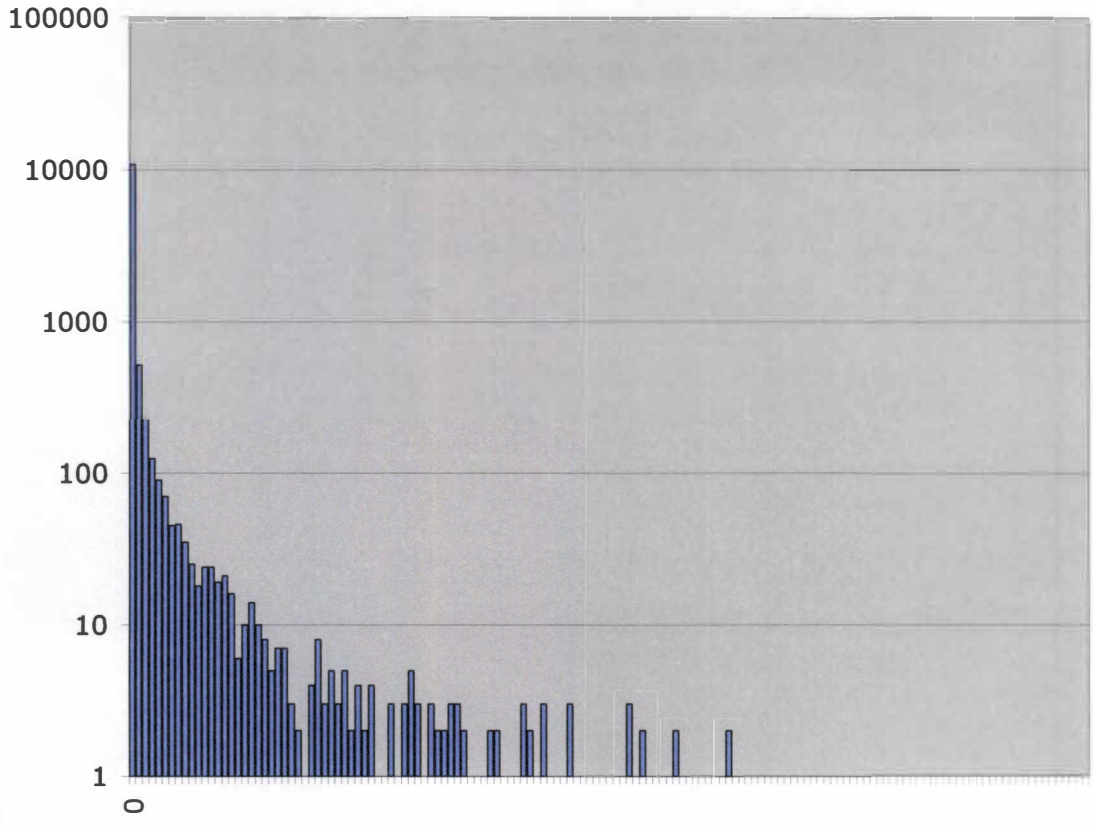

Figure 2-2. Degree histogram of 0.85 threshold MAS 5.0 graph.

Figure 2-3. Histogram of clique sizes for 0.85 threshold MAS 5.0 graph.

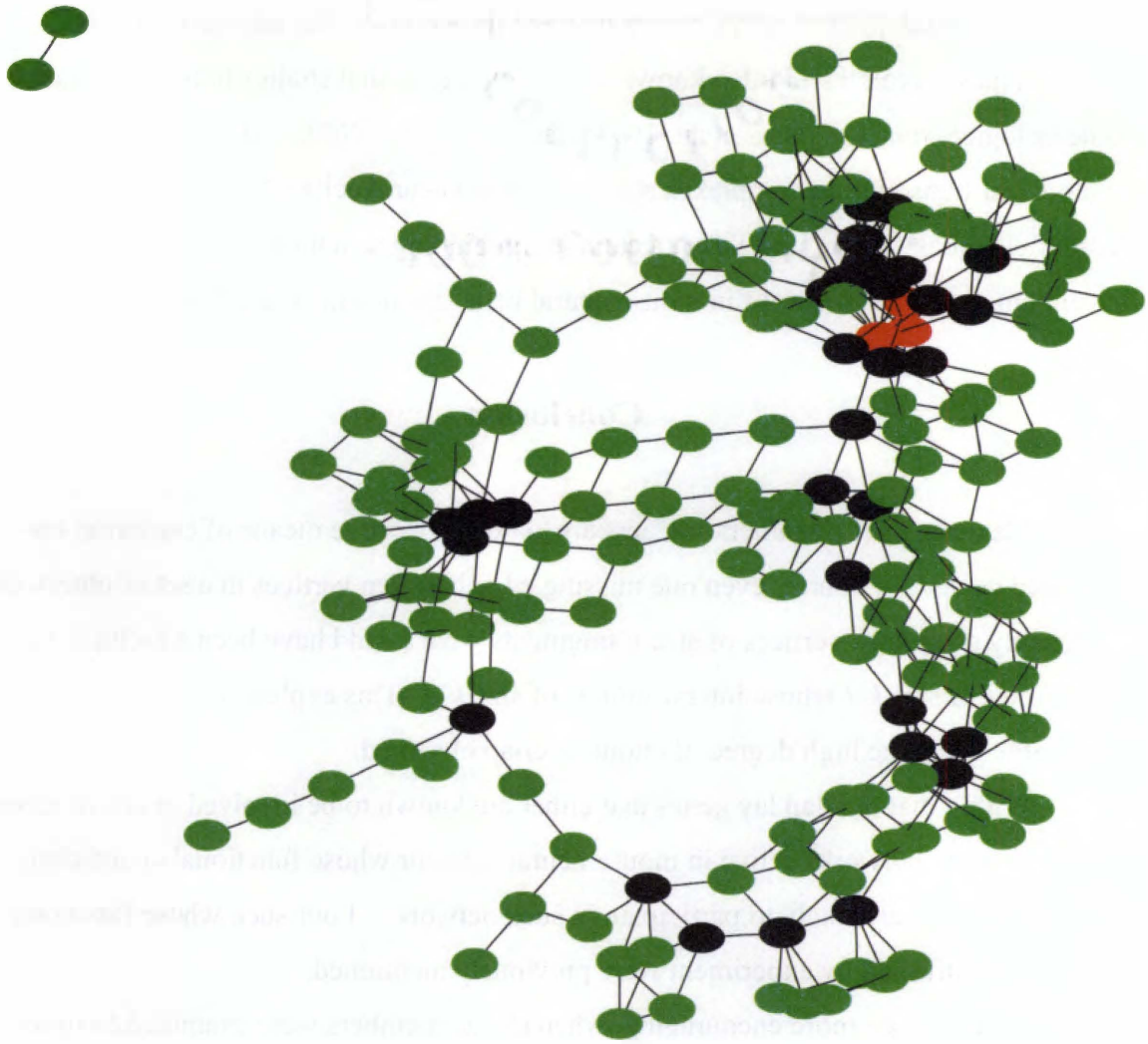

Figure 2-4. Clique intersection graph for 0.85 threshold MAS 5.0 graph. Vertices represent cliques of size 15 (green), 16 (black), and 17 (red). Each edge represents an intersection of at least size 13 between the endpoints (cliques).

observed. Examination of genes occurring most frequently in the intersection of the larger cliques reveals *Veli3* (also known as *Lin7c*), a gene that studies indicate is crucial to neurological function [Butz et al., 1998, Becamel et al., 2002]; *Sp3* and *Atf2*, members of a nuclear transcription complexes active in mouse neural cells [Cheng et al., 2004, Laifenfield, et al., 2004]; and *Strn3*, a calmodulin binding protein thought to be involved in calcium signaling pathways in mouse neural cells [Blondeau et. al., 2003].

Conclusion

Maximal clique enumeration appears to be an effective means of clustering co-regulated genes. Of course, even one missing edge between vertices in a set of otherwise completely connected vertices of size n fragments what would have been a k-clique into two cliques of size $k-1$ whose intersection is of size $k-2$. This explanation is even more reasonable given the high degree of clique overlap observed.

Within that overlap lay genes that either are known to be involved in one or more gene regulatory networks active in mouse neural cells, or whose functional annotations indicate that they are likely to participate in such networks. Four such whose functions have been confirmed by experiment were previously mentioned.

Perhaps even more encouraging, when clique members were examined as to their functional ontology[2], larger cliques were found, largely, to contain members belonging to the same or a closely related ontology group. An example of this can be seen in Figure 2-5 (original figure by Bing Zhang), where a clique of size eight contains five members classified as having a DNA-binding function and the remaining three members' ontologies are unknown.

[2] A formalized, general description of gene product function. Genes may belong to multiple functional ontologies.

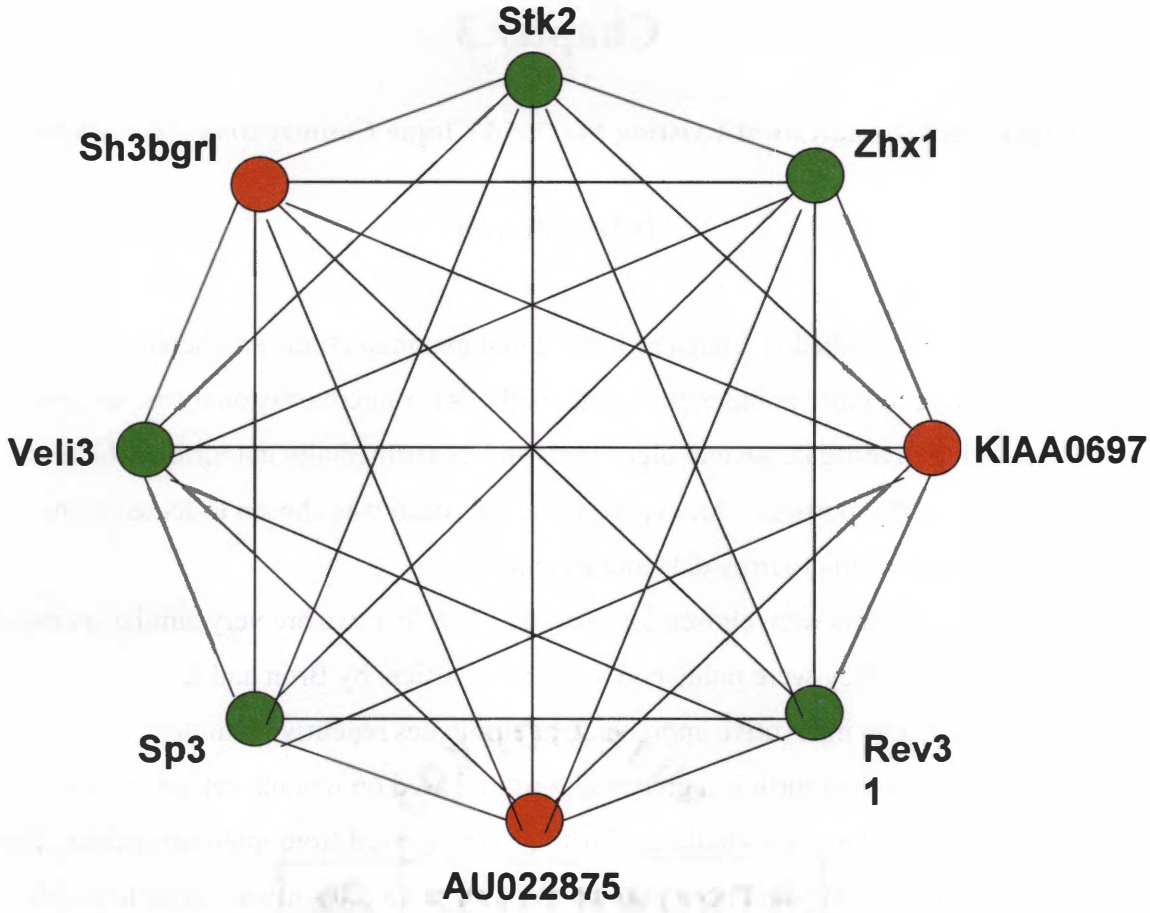

Figure 2-5. Representative clique containing veli3 (lin7c).
Green vertices represent genes whose functions include DNA binding. Red vertices represent genes whose functions are unknown or are not annotated.

Chapter 3

Experimental Analysis of Existing Maximal Clique Enumeration Algorithms

Introduction

Having established in Chapters 1 and 2 that maximal clique enumeration is a viable alternative to more popular clustering methods for microarray analysis, an obvious next step was to investigate several algorithms and experimentally determine which are most suitable for this purpose. An experimental approach was chosen to focus on the particular features of microarray data and its analysis.

Four algorithms were chosen for analysis. The first two are very similar recursive algorithms. Indeed they were published in the same article by Bron and Kerbosch in 1973. The third is an innovative approach that eliminates repetitive search of the same problem space, and the fourth is a greedy algorithm based on random set generation. Each of these algorithms was challenged with graphs derived from microarray data. For one set of graphs, the data had been normalized with MAS 5.0 software prior to graph generation, while the second set had been normalized with RMA. The inclusion of both types of normalization was necessary since both are equally prevalent in microarray analysis, yet each produces a very different end result, as can be seen in the differences in the resulting edge weights as shown in Figure 3-1. Unweighted graphs were produced from each of the MAS 5.0 and RMA normalized datasets using a range of threshold values and the algorithms were challenged with the results.

Description of Algorithms

Base Bron and Kerbosch Algorithm

Published in 1973 along with the more commonly referenced derivative algorithm discussed in the next section, the basic Bron and Kerbosch algorithm is a recursive

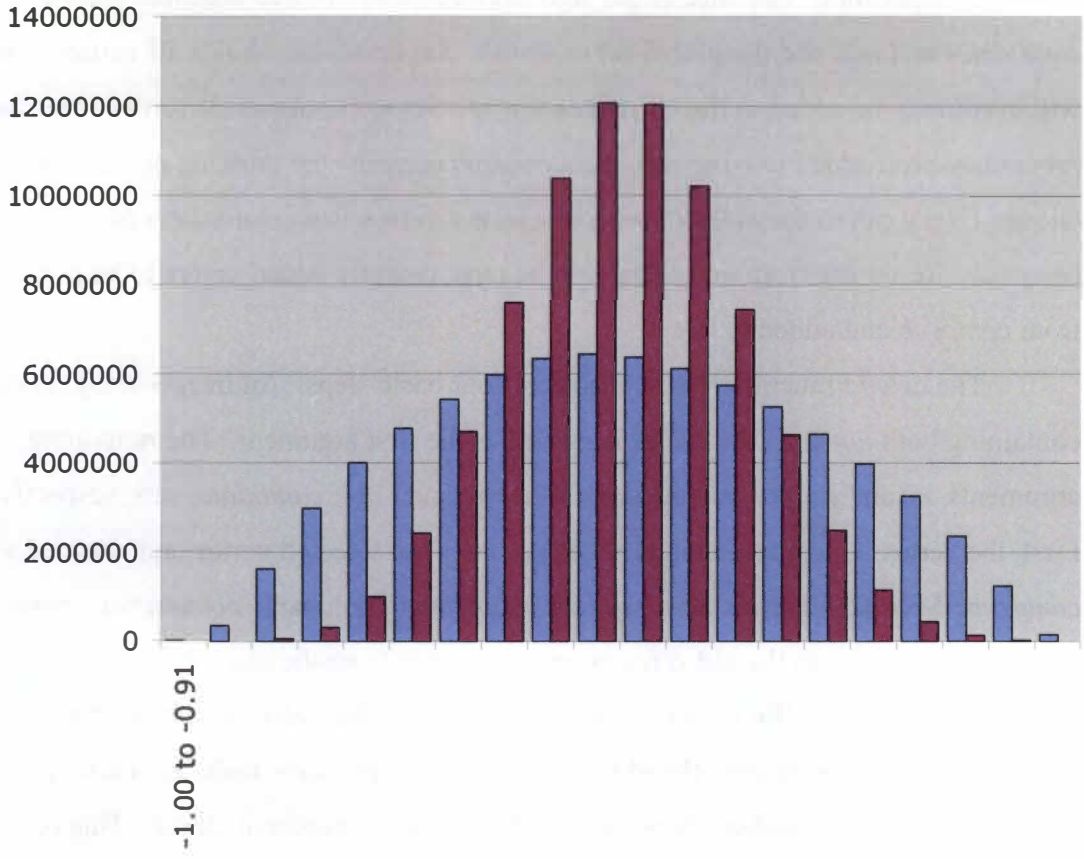

Figure 3-1. Edge weight histogram of MAS 5.0 and RMA derived graphs.

branching algorithm. The core of the algorithm consists of three sets: local sets *candidates* and *not*, and the global set *compsub*. Set *candidates* holds all vertices that will eventually be added to the current *compsub*. Set *not* contains all vertices that have previously been added to *compsub*. Set *compsub* contains the growing or shrinking clique. Every call to the *extend* function selects a vertex from *candidates* to add to *compsub*. Returning from *extend* causes the most recently added vertex to be removed from *compsub* and added to *not*.

The *extend* function itself consists of four basic steps. An integer array, *vertices*, containing both *not* and *candidates* is passed as the first argument. The remaining arguments, *ne* and *ce*, provide the size of the *not* and *not* + *candidates* sets, respectively. First, the vertex at position *ne* in *vertices* becomes the selected vertex and is added to *compsub*. Second, an array, *new_vertices* is created to hold *new_not* and *new_candidates* sets. Iterating through the old *vertices* array, a vertex from the old *not* set is added to *new_not* if and only if the vertex is connected to the earlier selected vertex. The set *new_candidates* is built from the old *candidates* set in the same fashion. Third, if *new_not* and *new_candidates* is empty, *compsub* holds a maximal clique. This is reported, and the function returns. Otherwise, *extend* is called on *new_vertices* to operate on the new sets just formed. Fourth, upon returning, the selected vertex is removed from *compsub* and added to the old set *not*. As long as set *candidates* is not empty, the function begins again with the first step. Pseudocode for this function is provided in Figure 3-2.

Bron and Kerbosch Algorithm

The second of the Bron and Kerbosch algorithms published in 1973 follows the branching blueprint laid out by the base algorithm, but also takes some measures to limit the number of branches traversed. It's worst case time complexity has only very recently been proven to be $O(3^{n/3})$, where n is the number of vertices and the clique list is not printed. Printing the list adds a factor of n, resulting in $O(n \cdot 3^{n/3})$ [Tomita et al., 2004]. The main difference from the base algorithm lies in the choice of

Let *size* = 0 when extend is initially called

```
extend(vertices, ne, ce) {
        while (ne < ce) {
                selected = vertices[ne];
Step 1          compsub[size] = selected;
                size++;

                new_ne = 0;
                for (i = 0; i < ne; i++)
                        if vertices[i] connected to selected {
                                new_vertices[new_ne] = vertices[i];
                                new_ne++;
                        }
Step 2          new_ce = new_ne;
                for (i = 0; ne + 1; i < ce; i++)
                        if vertices[i] connected to selected {
                                new_vertices[new_ce] = vertices[i];
                                new_ce++;
                        }

                if new_ce == 0 {
                        compsub contains maximal clique
                        return;
Step 3          }
                else
                        extend(new_vertices, new_ne, new_ce);

                size--;
Step 4          ne++;
        }
}
```

Figure 3-2. Pseudocode for base Bron and Kerbosch algorithm.

the selected vertex in step one. Instead of choosing vertices in the order they are presented, this algorithm finds the vertex with the most number of connections to the other vertices in *candidates* and swaps it with the vertex at position *ne*. The rationale for this is the following. If at any point set *not* contains a vertex that is connected to all vertices in set *candidates*, it is not possible to generate a new maximal clique with the current sets, and the function should return. Clearly, it would be best in terms of running time if this boundary condition is reached as soon as possible in order to eliminate the most number of branches that would otherwise be traversed.

This modification is, of course, only useful if the time spent finding maximally connected vertices and performing the subsequent swap is less than the time that would have been spent exploring the eliminated branches of the search space. The expectation was that this algorithm would be a better choice for graphs with areas of large numbers of highly overlapping cliques. Under these conditions, the algorithm should encounter the bounding condition more frequently to provide the greatest advantage. On the other hand, the base algorithm should be faster when the input has little clique overlap or the number of cliques is sufficiently small.

A Constructive Algorithm

This algorithm, published by Kose et. al. in 2001, takes a very different approach than the recursive branching procedure of Bron and Kerbosch. It takes advantage of the fact that every clique of size k, where $k \geq 2$, is comprised of two cliques of size k-1 that share k-2 vertices. Using this basic principle (illustrated in Figure 3-3), the algorithm takes as input an edge list with the edges (2-cliques) listed in non-repeating, canonical order and builds from it all possible 3-cliques. Any 2-clique that cannot become a component of a 3-clique is declared maximal and the list of 2-cliques is deleted. The algorithm then attempts to construct 4-cliques from the just built 3-cliques using the same procedure. This continues, enumerating maximal cliques in increasing order of size until it is no longer possible to build a larger clique.

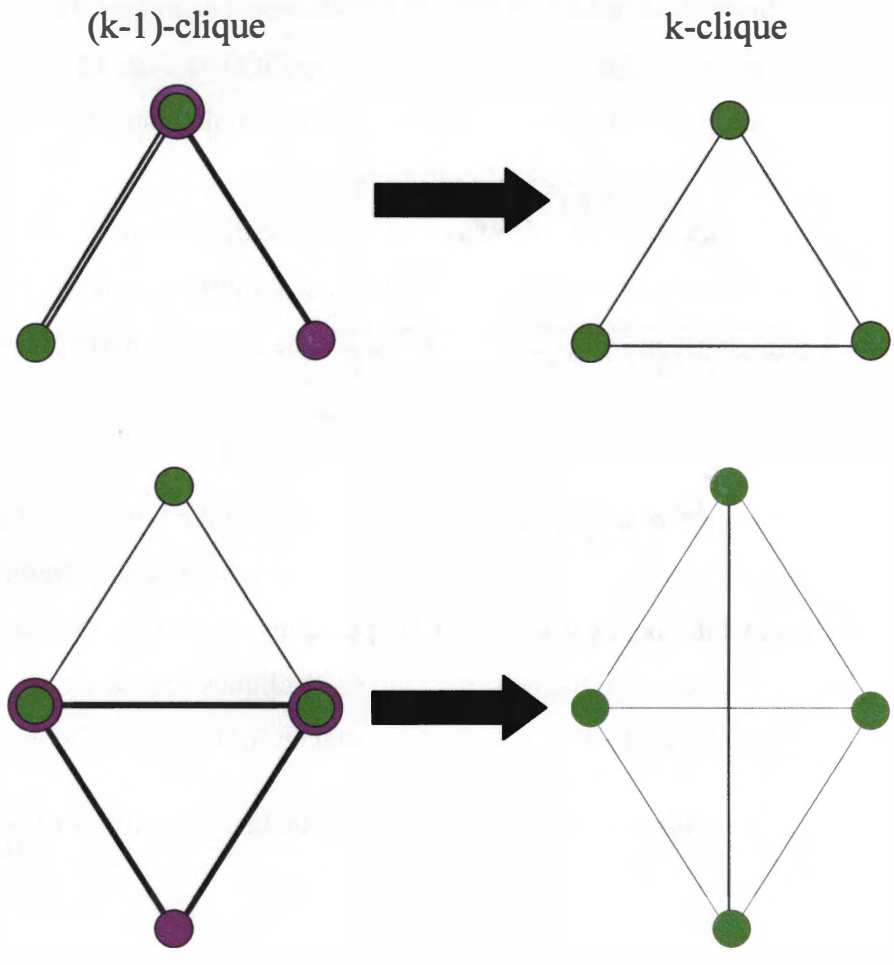

Figure 3-3. Any *k*-clique is comprised of two (*k-1*)-cliques sharing *k-2* vertices. (A) Two 2-cliques sharing one vertex (green and purple). The addition of an edge connecting the green vertex with the purple vertex results in a 3-clique. (B) Two 3-cliques sharing two vertices (green and purple). As in (A), adding an edge between the green and the purple vertices creates a 4-clique.

This algorithm was attractive, in that it prevents repeat searching of the same space. Once a clique is built, the connections that formed it do not need to be re-discovered. The algorithm treats those vertices as a unit from then on. This is in direct contrast to the other algorithms discussed in this chapter.

Unfortunately, the algorithm also has less than appealing features. First, it was evident that building cliques in this manner requires the computer to maintain somewhere a list of cliques being built and a list of cliques that are the current building blocks. With a graph of size *n*, building cliques of size *k* requires $O\left(\frac{n!}{(n-k)!}\right)$ memory space. For graphs of any significant size and density, it is not feasible for the typical workstation to keep these lists in main memory. However, if the lists are kept on disk, a tremendous amount of overhead would be incurred from I/O operations. Secondly, the algorithm has a hidden cost. Every time a *k*-clique is formed, all (*k-1*)-cliques contained within the new clique must be marked as used, or they might be mistaken for maximal cliques. This cost is not negligible, as it requires a search of the (*k-1*)-clique list. This list is $O\left(\frac{n!}{(n-k-1)!}\right)$ in length.

A Greedy Algorithm

The greedy algorithm employed is the most basic of clique enumeration algorithms. The counter, *k*, is set to a user-determined maximal clique size. All vertices with degree less than (*k-1*) are removed from the graph. Then, while *k* is greater than a user-determined minimal clique size, it generates all k-sets and tests each to determine if it is a clique, or that all set members are completely connected to one another. If so, the neighborhood of one of the clique members is examined to determine whether one of the neighbors is completely connected to the set. If not, then the clique is maximal. Once all k-sets have been tested, *k* is decremented and the loop continues. Although this algorithm was likely to perform poorly in comparison with the others employed, given its

$O\left(\sum_{k=3}^{n} \frac{n!}{(n-k)!}\right)$ time complexity. It was chosen for its ability to enumerate maximal cliques in descending order of size, a feature that would be extremely useful in many applications.

Methods Employed

The raw microarray data detailed in Chapter 2 was used in these experiments. The data was normalized with either the MAS 5.0 software package or with the RMA function as implemented in the BioConductor. As before, pairwise Spearman's rank coefficients were calculated for each of the MAS 5.0 and RMA-treated datasets, resulting in two 12,422 x 12,422 weighted adjacency matrices, where 12,422 was the number of genes measured in the microarray experiment. Multiple thresholds, including those chosen for actual analysis of each graph were used to filter the weighted graphs. Thresholds for the MAS 5.0 graph were 0.70, 0.75, 0.80, and 0.85. Thresholds for the RMA graph were 0.95, 0.921954446, 0.90, 0.87. Percent edge densities of the resultant graphs is reported in Table 3-1. (Percent edge density is defined as the number of edges in the paraclique divided by the maximum number of edges possible in the paraclique multiplied by one hundred.) Each unweighted graph was then provided as input for each of the algorithms discussed and the compute times recorded. All experiments were performed on an Apple Powerbook using a 1GHz PowerPC G4 processor with 256K L2 cache and 1MB L3 cache outfitted with 1024MB of RAM. Bus speed was 133MHz.

Table 3-1. Graph Edge Densities

(A) RMA Graphs

	Threshold			
	0.95	0.921954446	0.90	0.87
Edge Density	0.0082%	0.0743%	0.2093%	0.5526%

(B) MAS 5.0 Graphs

	Threshold			
	0.85	0.80	0.75	0.70
Edge Density	0.0080%	0.0371%	0.1178%	0.2972%

Results

Each of the four algorithms was implemented by the author and run on graphs derived from either RMA or MAS 5.0 treated microarray data. Two versions of the Kose algorithm were implemented to determine the overhead induced by I/O operations when accessing clique lists. Results are presented in Tables 3-2 and 3-3.

Under these conditions, the two worst performers were the Kose and greedy algorithms. The greedy algorithm was halted after a day on all graphs. The fastest implementation of the Kose algorithm, that which kept its clique lists in core memory, finished on the two sparsest graphs, the 0.95 threshold RMA graph (0.0082 edge density) and the 0.85 threshold MAS 5.0 graph (0.0080 edge density) in a little over and a little under five hours, respectively. It was not capable of finishing on any other graphs in less than a day. The implementation of Kose storing clique lists on disk was still running after a week's time on both the 0.95 threshold RMA graph and the 0.85 threshold MAS 5.0 graph.

The base Bron and Kerbosch algorithm, as anticipated, performed the best on the sparsest graphs. It was nearly twice as fast as the branch and bound Bron and Kerbosch algorithm, finishing at six seconds as opposed to eleven. However, when challenged with denser graphs, the branch and bound Bron and Kerbosch algorithm was clearly superior to all others tested. It finished the 0.80 threshold MAS 5.0 graph (0.0371 edge density) in thirteen seconds as opposed to the base algorithm's 193 seconds, and was the only algorithm capable of finishing the MAS 5.0 graphs with thresholds of 0.75 (0.1178 edge density) or 0.70 (0.2972 edge density). Similar results were seen with the RMA graphs, where only the branch and bound algorithm finished the 0.921954446 and 0.90 threshold graphs (edge densities of 0.0743 and 0.2093, respectively) in less than a day. Observe from Table 3-2 that 0.921954446 was the threshold used. This number was chosen for a recent analysis of the RMA treated data by our colleagues in neurobiology. The branch and bound algorithm was unable to finish enumerating all maximal cliques of the 0.87 threshold RMA graph (0.5526 edge density) in less than a day.

Table 3-2. Time Trials for Maximal Clique Enumeration on RMA Microarray Data

Algorithm	Threshold			
	0.95	0.921954446*	0.90	0.87
Base BK	6 sec	Halted after 1 day	N.A.	N.A.
BK	11 sec	419 sec	53220 sec	Halted after 1 day
Kose (RAM)	18632 sec	Halted after 1 day	N.A.	N.A.
Kose (Disk)	Halted after 1 week	N.A.	N.A.	N.A
Greedy	Halted after 1 day	N.A.	N.A.	N.A

*Threshold used for actual analysis

Table 3-3. Time Trials for Maximal Clique Enumeration on MAS 5.0 Microarray Data

Algorithm	Threshold			
	0.85*	0.80	0.75	0.70
Base BK	6 sec	193 sec	Halted after 1 day	N.A.
BK	11 sec	13 sec	257 sec	53470 sec
Kose (RAM)	17261 sec	Halted after 1 day	N.A.	N.A.
Kose (Disk)	Halted after 1 week	N.A.	N.A.	N.A.
Greedy	Halted after 1 day	N.A.	N.A.	N.A

*Threshold used for actual analysis

Conclusions

Of the existing maximal clique enumeration algorithms tested, the most suited to DNA microarray analysis seems to be the branch and bound Bron and Kerbosch algorithm. Although the base Bron and Kerbosch algorithm performed better on very sparse graphs, the branch and bound algorithm was significantly faster on the denser graphs and the loss of a few seconds on sparse graphs is not sufficient to rationalize choosing the base algorithm over the branch and bound algorithm.

The Kose algorithm, while interesting is not useful for this application. In addition to being more than 1,000 times slower than either Bron and Kerbosch algorithm at its best, it generates cliques in increasing order. Since, for this application, the desired cliques tend to be large, this confers no advantage. Worse, the fastest implementation of the Kose algorithm has memory requirements that are not likely to be met by most workstations when running graphs of any significant density. Running this algorithm on the sparsest graphs was only possible with all other processes save system software were terminated, as it monopolized the available memory. This would only worsen as the graph density increased.

Another promising algorithm is the greedy algorithm based on k-set enumeration. Although it was not able to enumerate all maximal cliques within a day on any provided input, the algorithm has ample opportunity for improvement with the introduction of boundary conditions, such as are used in the Bron and Kerbosch algorithm. It is possible that this algorithm could be useful in enumerating cliques when tight size boundaries are imposed. We realize that this experimental study has a number of limitations. Among these are the limited amount of data and chosen thresholds, and therefore, a limited number of graphical inputs. We anticipate more extensive studies as new maximal clique algorithms are brought online.

Chapter 4

Algorithm Development for Application to DNA Microarray Analysis

Introduction

Thus far, our application of graph-based algorithms to DNA microarray analysis had used only pre-existing maximal clique enumeration algorithms in order to establish their utility for this purpose. Satisfied that the basic approach was a complementary approach to more popular clustering techniques and that the problem, though *NP*-hard, was solvable in a reasonable amount of time for most expected inputs, the focus turned to improving the speed of the existing enumeration algorithms. Only the branch and bound Bron and Kerbosch algorithm described in Chapter 3 was considered for improvement, based on its overall performance. Additionally, a means to address the issue of noise was sought. A new, clique-based algorithm was developed and tested for that purpose.

Results

Enumeration Adaptations

The notion of fixed-parameter tractability [Downey and Fellows, 1999] has been useful when devising approaches for solving *NP*-complete problems. Formally defined, a problem is fixed-parameter tractable if it can be solved in time $O(f(k)\cdot n^{\alpha})$, where f is any function, and α is a constant independent from both k and n. Although clique is not fixed-parameter tractable unless the W-hierarchy collapses, the notion of imposing limits based on expected inputs should allow the adapted algorithms to process graphs that were previously unsolvable in a reasonable amount of time.

One such technique that is a cornerstone of fixed-parameter tractability is data preprocessing. When a minimum clique size is specified, two preprocessing rules become available. The first of these is the low degree rule. The rule states that all

vertices of degree less than *k-1* may be removed from the graph if the minimum clique size is known to be *k*. A corollary to this is the minimum common neighbor rule [AbuKhzam, 2004], that states that if two connected vertices, *i* and *j*, share less than *k-2* common neighbors, then the edge *(i, j)* may be removed from the graph, again, given that the minimum clique size is *k*. These methods are doubly useful in that they can be applied regardless of the enumeration algorithm, so long as a minimum clique size is defined.

A second technique to limit the search space is to introduce more bounding rules to the original algorithms. On examination, it was found that two rules could be added to the existing algorithm. Testing for a known minimum clique size allows the branch and bound algorithm to return immediately if there are insufficient candidate vertices to extend the current clique. Once a *k*-clique is found, removing any member vertex whose degree is *k-1* prevents later redundant searching.

Each of the above mentioned adaptations was implemented by the author and tested, separately and in combination. The experiments were performed as described in Chapter 3 on the same input graphs and hardware. Results are shown in Table 4-1.

The addition of boundary conditions did not improve performance. The boundary conditions were not met frequently enough to counterbalance the increased number of instructions necessary to implement them. When a direct comparison is made, algorithms with additional boundary conditions were either equal to or slower on the same input than those without. Adding either the low degree or minimum neighbor preprocessing rules only improved running time on graphs with an edge density lower than 0.1178%. In any other case, running time increased up to 3.25 fold. Even when running time was improved over the algorithm with no preprocessing rule, it was still slower than the original Bron and Kerbosch algorithm with the same preprocessing rule

The original Bron and Kerbosch algorithm in combination with the low degree preprocessing rule improved performance in all cases where a comparison could be made (when at least one program finished in less than a day) with the exception of the 0.90

Table 4-1. Time Trials for Preprocessing and Boundary Rules on MAS 5.0 Microarray Data.

	MAS 5.0				RMA			
	0.85	0.80	0.75	0.70	0.95	0.92195	0.90	0.87
BK	11 sec	13 sec	257 sec	53470 sec	11 sec	419 sec	53220 sec	Halted after 1 day
BK+Low Degree	< 1 sec	4 sec	253 sec	50285 sec	< 1 sec	113 sec	Halted after 1 day	Halted after 1 day
BK+Min. Neighbor	1 sec	7 sec	388 sec	56787 sec	1 sec	117 sec	59056 sec	Halted after 1 day
BK+bounds	11 sec	16 sec	265 sec	53510 sec	11 sec	425 sec	53318 sec	Halted after 1 day
BK+ bounds+ Low Degree	< 1 sec	6 sec	542 sec	Halted after 1 day	<1 sec	252 sec	Halted after 1 day	N.A.
BK+bounds+ Min. Neighbor	1 sec	12 sec	861 sec	Halted after 1 day	1 sec	263 sec	Halted after 1 day	N.A.

threshold graph. The decrease in running time was more marked on sparse graphs, with a greater than eleven-fold speedup on the 0.85 threshold MAS 5.0 graph and the 0.95 threshold RMA graph. There was still some improvement on the denser graphs, with the worst being a 1.5% speedup on the 0.75 threshold MAS 5.0 graph. In contrast, addition of the minimum neighbor rule only improved running time on graphs with an edge density lower than 0.1178%. In any other case, running time increased up to 11%.

Noise Compensation

One disadvantage to using maximal clique enumeration for DNA microarray analysis is the inability of clique to compensate for noise. This is a serious issue, because there are multiple sources of noise in a microarray experiment. Biological variations among cells and/or tissues are one such, but these are typically subsumed by experimental noise. The greatest sources of noise have been determined to be introduced during the hybridization and subsequent readout steps [Tu et al., 2002]. Unfortunately, this means that significant reductions in noise levels are dependent on improvements in hybridization and image analysis technologies, rather than the more easily controlled experimental design.

For our purposes of all this noise can be to artificially raise or lower a gene's signal strength in both raw and normalized data. While this alteration may have multiple effects, two are our primary concerns. The first occurs when the gene is properly a member of a co-regulation group and the change causes a decrease in pairwise correlation coefficients between the affected gene and one or more group members. The second occurs when the pairwise correlation coefficients between genes that are not co-regulated are increased.

For the most part, it is the first situation that is of greatest concern. Consider what must occur for a gene to be falsely included in a co-regulation group. The correlation coefficients between the gene and a significant number of members of the group must increase above the applied threshold to form a clique of sufficient size. On the other hand, decreasing the correlation coefficient between the gene and even one member of its

proper co-regulation group below the applied threshold fragments the clique. Furthermore, assuming that the affected gene does belong to a co-regulation group, it is more likely that any change in its expression pattern would weaken its relationship to its group members than that the change would strengthen the pattern's similarity to sufficient members of an unrelated co-regulation group.

Therefore, our goal was to develop an algorithm that would detect genes that were likely excluded from a clique as a result of noise and re-integrate them into the appropriate group [Langston, 2004]. The algorithm needed to meet the three requirements. First, the algorithm was to be clique-based. This would provide a solid base from which to expand the co-regulated group. Second, the end result needed to have a high edge density. This would indicate that most likely all members are co-regulated and limit the number of false inclusions. Third, the result should be somewhat robust. That is, changes such as re-ordering the input graphs should not change the result. After some consideration, a simple algorithm was conceived that met all of these requirements.

Paraclique

To handle the noisy data, we devised the scheme described below. Because there are many clique variants already known, we left the naming of this method to our colleagues in neurobiology. Dr. Rob Williams, a colleague from neurobiology, coined the term paraclique, and it has stuck. The paraclique algorithm takes as input a weighted graph, G_w; an unweighted subgraph of G_w filtered with threshold H, G_H; a tolerance, T; a paraclique factor, $0 \leq k < |C_{max}|$; and a maximum clique, C_{max}, from G_H. The paraclique, P, is set to C_{max}. For every vertex, $v \in \{G_H\text{-}P\}$, if v is connected to at least k vertices in P and for all $v_P \in P$, |weight of $(v, v_P)| \geq H\text{-}T$, then $P = P \cup v$. This loop is repeated until no more vertices can be added to P. An unweighted graph, $G_H\text{-}P$, and paraclique, P, are output. If more than one paraclique is desired, the new unweighted subgraph, G_H, is set to $G_H\text{-}P$ from the previous iteration.

This algorithm was applied to the 0.85 threshold RMA graph. The tolerance was set at 0.05. The paraclique factor was maintained at $k\text{-}1$, where k was the size of the input

clique. The 1-neighborhood of the input clique was also computed for comparison. Results are in Table 4-2. In all instances, the paraclique algorithm maintained a high edge density, at least a four-fold increase over the 1-neighborhood. Paraclique edge density increased with increasing size of the input clique.

Conclusions

An experimental study of three methods of potentially improving the Bron and Kerbosch algorithm's performance on graphs derived from microarray data revealed that preprocessing, in particular the low degree rule, was the most effective technique of those tested. However, significant speedup was only observed in the sparsest graphs, where it is least needed. Indeed, on the densest graphs, application of either preprocessing rule resulted in a net decrease in performance. Intuitively, this seems impossible. However, eliminating some vertices can limit the efficacy of the bounding rules in the original Bron and Kerbosch algorithm. This effect was also observed, to a more pronounced degree, when additional bounding rules were applied to the original algorithm lending further credence to this rationale.

As discussed, the primary disadvantage to maximal clique enumeration as a microarray analysis tool is its inability to compensate for the noise inherent in such data. The goal was to develop an algorithm that retains much of the stringent requirements of clique, yet incorporates the "near misses" that cause clique fragmentation in the enumeration algorithm. As a first attempt, the paraclique algorithm was developed. Initial experiments show that a high level of edge density is maintained in the resulting paracliques when the given parameters are used.

Although this algorithm produces only vertex disjoint paracliques, eliminating one advantage to using clique, it can be used to decompose graphs not tractable to maximal clique enumeration, such as the 0.85 threshold RMA graph. The reason for this

Table 4-2. Comparison of Paraclique and 1-Neighborhood.

Core Clique	Paraclique		1-Neighborhood	
	Size	Edge Density	Size	Edge Density
280	466	95.58%	2657	16.09%
113	193	93.80%	1636	17.22%
72	132	90.05%	2067	22.68%
58	127	86.74%	2320	18.50%

is that the paraclique algorithm uses maximum clique, a much more efficient algorithm than maximal clique enumeration, to generate its core clique input. Research into alternate versions of this algorithm are ongoing. A relatively minor change to the algorithm, constraining the maximum clique to contain at least one vertex disjoint from all already elucidated paracliques, would allow overlap. Our colleagues in neurobiology are encouraged by the results of paraclique because it parallels their study of quantitative trait loci. Other methods of determining which vertices become members should also be investigated.

List of References

AbuKhzam FN. Private communication. 2004.

Abu-Khzam FN, Collins RL, Fellows MR, Langston MA, Suters WH, Symons CT. Kernelization algorithms for the vertex cover problem. *Proceedings, Workshop on Algorithm Engineering and Experiments (ALENEX)*, New Orleans, LA, January, 2004.

Abu-Khzam FN, Langston MA, Shanbhag P. Scalable Parallel Algorithms for Difficult Combinatorial Problems: A Case Study in Optimization. *Proceedings, International Conference on Parallel and Distributed Computing and Systems*, Los Angeles, CA, 563-568, November, 2003.

Baldwin NE, Chesler EJ, Kirov S, Langston MA, Snoddy JR, Williams RW, Zhang B. Computational, integrative and comparative methods for the elucidation of gene regulatory networks. *Journal of Biomedicine and Biotechnology*. In press.

Baldwin NE, Collins RL, Langston MA, Leuze MR, Symons CT, Voy BR. High performance computational tools for motif discovery. *Proceedings, IEEE Workshop on High Performance Computational Biology*, Santa Fe, NM, April, 2004.

Becamel C, Alonso G, Galeotti N, Demey E, Jouin P, Ullmer C, Dumuis A, Bockaert J, Marin P. Synaptic multiprotein complexes associated with 5-HT(2C) receptors: a proteomic approach. *EMBO J.* 21(10): 2332-42. 2002.

Beer DG, Kardia SL, Huang CC, Giordano TJ, Levin AM, Misek DE, Lin L, Chen G, Gharib TG, Thomas DG, Lizyness ML, Kuick R, Hayasaka S, Taylor JM, Iannettoni MD, Orringer MB, Hanash S. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine* 9 (816): 816-824, 2002.

Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S, Vasa P, Ladd C, Beheshti J, Bueno R, Gillette M, Loda M, Weber G, Mark EJ, Lander ES, Wong W, Johnson BE, Golub TR, Sugarbaker DJ, Meyerson M. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc Natl Acad Sci USA*. 98 (24): 13790-13795, 2001.

Blondeau C, Gaillard S, Ternaux JP, Monneron A, Baude A. Expression and distribution of phocein and members of the striatin family in neurones of rat peripheral ganglia. *Histochem Cell Biol.* 119(2):131-8. 2003.

Bron C and Kerbosch J. Algorithm 457: finding all cliques of an undirected graph. *Proceedings of the ACM.* 16(9): 575-577, 1973.

Butz S, Okamoto M, Sudhof TC. A tripartite protein complex with the potential to couple synaptic vesicle exocytosis to cell adhesion in brain. *Cell.* 94(6):773-82. 1998.

Cheng L, Jin Z, Liu L, Yan Y, Li T, Zhu X, Jing N. Characterization and promoter analysis of the mouse nestin gene. *FEBS Lett.* 2004. 565(1-3): 195-202.

Czerwinski M, McLemore TL, Gelboin HV, Gonzalez FJ. Quantification of CYP2B7, CYP4B1, and CYPOR messenger RNAs in normal human lung and lung tumors. *Cancer Res.* 54(4): 1085-91, 1994.

Das R, Mahabeleshwar GH, Kundu GC. Osteopontin stimulates cell motility and nuclear factor kappaB-mediated secretion of urokinase type plasminogen activator through phosphatidylinositol 3-kinase/Akt signaling pathways in breast cancer cells. *J Biol Chem.* 278(31): 28593-606, 2003.

R. G. Downey and M. R. Fellows. Parameterized Complexity. Springer-Verlag. 1999.

Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. *J Comput Biol.* 7(3-4):601-20, 2000.

Garber ME, Troyanskaya OG, Schluens K, Petersen S, Thaesler Z, Pacyna-Gengelbach M, van de Rijn M, Rosen GD, Perou CM, Whyte RI, Altman RB, Brown PO, Botstein D, Petersen I. Diversity of gene expression in adenocarcinoma of the lung. *Proc Natl Acad Sci U S A.* 98(24):13784-13789, 2001.

Garey MR, Johnson DS. Computers and Intractability. W. H. Freeman, New York, 1979.

Hogdall CK, Norgaard-Pedersen B, Mogensen O.The prognostic value of pre-operative serum tetranectin, CA-125 and a combined index in women with primary ovarian cancer. *Anticancer Res.* 22(3):1765-8, 2002.

Hu JH, Yin GS, Morris JS, Zhang L, Wright FA. Entropy and survival-based weights to combine Affymetrix array types in the analysis of differential expression and

survival. *Critical Assessment of Microarray Data Analysis "CAMDA'03": Oral and Poster Presenters Abstracts*, 78-82, 2003.

Imaoka S, Yoneda Y, Sugimoto T, Hiroi T, Yamamoto K, Nakatani T, Funae Y. CYP4B1 is a possible risk factor for bladder cancer in humans. *Biochem Biophys Res Commun.* 277(3): 776-80, 2000.

Irizarry RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed TP. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* 4(2): 249-264, 2003.

Kose F, Weckworth W, Linke T, Fiehn O. Visualizing plant metabolic correlation networks using clique metabolite matrices. *Bioinformatics.* 17(12): 1198-1208, 2001.

Laifenfeld D, Karry R, Grauer E, Klein E, Ben-Shachar D. ATF2, a member of the CREB/ATF family of transcription factors, in chronic stress and consequent to antidepressant treatment: animal models and human post-mortem brains. *Neuropsychopharmacology.* 29(3):589-97. 2004.

Langston MA. Private communication. 2004.

Mahabeleshwar GH, Kundu GC. Syk, a protein-tyrosine kinase, suppresses the cell motility and nuclear factor kappa B-mediated secretion of urokinase type plasminogen activator by inhibiting the phosphatidylinositol 3'-kinase activity in breast cancer cells. *J Biol Chem.* 278(8):6209-21, 2003.

Reznikoff WS, The lactose operon-controlling elements: a complex paradigm. *Molecular Microbiology.* 6(17): 2419-22.

del Rio G, Bartley TF, del-Rio H, Rao R, Jin KL, Greenberg DA, Eshoo M, Bredesen DE. Mining DNA microarray data using a novel approach based on graph theory. *FEBS Letters* 509(2):230-4, 2001.

Sok JC, Kuriakose MA, Mahajan VB, Pearlman AN, DeLacure MD, Chen FA. Tissue-specific gene expression of head and neck squamous cell carcinoma in vivo by complementary DNA microarray analysis. *Arch Otolaryngol Head Neck Surgery* 129(7):760-70, 2003.

Tomita E, Tanaka A, Takahashi, H. The worst-case time complexity for generating all maximal cliques. *Proceedings, Computing and Combinatorics Conference (COCOON)*. Jeju Island, Korea. August, 2004.

Tu Y, Stolovitzky, G, Klein, U. Quantitative noise analysis for gene expression microarray experiments. *PNAS*. 99(22):14031-14036, 2002.

de Vries JE, Meyering M, van Dongen A, Rumke P. The influence of different isolation procedures and the use of target cells from melanoma cell lines and short-term cultures on the non-specific cytotoxic effects of lymphocytes from healthy donors. *Int J Cancer*. 15(3):391-400, 1975.

Zhang B, Schmoyer D, Kirov S, Snoddy J. GOTree Machine (GOTM): a web-based platform for interpreting sets of interesting genes using gene ontology hierarchies. To appear in *BMC Bioinformatics*, 2004; http://genereg.ornl.gov/gotm.

Vita

Nicole E. Baldwin was born in Garland, Texas on September 27, 1974. She spent much of her early life in nearby Plano, Tx, attending the since re-named Plano Senior High School and graduating in 1992. She then went to Texas A&M University in College Station, earning both Foundation and University Honors and receiving a B. S. degree with a double major in biochemistry and genetics in 1996.

In 2001, Nicole earned a Ph.D. in microbiology and molecular genetics from the University of Texas Health Science Center at Houston and chose to enter the field of computational biology. To that purpose, she came to the University of Tennessee, Knoxville with the intent of earning an M.S. in computer science, the culmination of which resulted in this thesis.

5901 7111 47
04/06/05 MAB