EARLY DETECTION OF OVARIAN CANCER USING GABOR WAVELET PHASE QUANTIZATION AND BINARY CODING

Stuart M. Morton

Submitted to the faculty of the School of Informatics
in partial fulfillment of the requirements
for the degree
Master of Science
in Bioinformatics,
Indiana University

November 2006

Accepted by the Faculty of Indiana University,
in partial fulfillment of the requirements for the degree of Master of Science
in Bioinformatics.

**Master's Thesis
Committee**

____
Jeffrey Huang, Ph.D., Chair

____
Mahesh Merchant, Ph.D.

____
Snehasis Mukhopadhyay, Ph.D.

Dedicated to my wife Amy, my daughter Kate, and my parents. Without their love and support this would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

ACKNOWLEDGEMENTS

I would like to wish a sincere wish of gratitude to all of those individuals who have supported me during my pursuit of a Masters degree in Bioinformatics at Indiana University. A special thanks to Dr. Huang for all of his input and understanding during the last 18 months of this maters program and thesis research.

ABSTRACT

Stuart Morton

EARLY DETECTION OF OVARIAN CANCER USING GABOR WAVELET PHASE QUANTIZATION AND BINARY CODING

Ovarian cancer is the 5th most common cancer in women, but it is the most difficult to detect in its early stages. Early detection and treatment of ovarian cancer has been shown to increase the five year survival rate of a woman from 12% if caught in stage four of the disease up to 92% if caught in stage one of the disease. Using signal processing, pattern classification and a learning algorithm, it is possible to identify patterns in high dimensionality mass spectrometry data that distinguishes between cancer and non-cancer ovarian samples. For our research, proteomic spectra were generated using SELDI-TOF mass spectrum data, which was composed of 162 ovarian cancer and 91 non-ovarian cancer samples. We introduce a Gabor filter on the mass spectrometry data and design a binary coding scheme for phase quantization encoding that is used for the pattern classification. This pattern will expose crucial features in the data that can be used to correctly classify unmasked samples for the presence or absence of ovarian cancer. Our proposed algorithm was able to successfully discriminate ovarian cancer and non-ovarian samples that yielded results with sensitivities, specificities and accuracies in the 90% to 100% range.

CHAPTER ONE: INTRODUCTION

Introduction to Subject

Data published by the American Cancer Society has shown that approximately 1.4 percent of women in the United States will develop ovarian cancer during their lifetime (Fergus 2000b). This percentage is equivalent to about 22,500 new cases of ovarian cancer each year in the United States, which resulted in approximately 15,000 deaths in 2006 (NOCC 2006). As with other types of cancer, ovarian cancer is the result of excessive growth of some cells in the epithelial layer that surrounds the ovaries. The cancer can then spread to other parts of the body via the bloodstream or lymphatic system, via a process called shedding, which involves ovarian cancer cells that break away and spread to other tissues or organs.

Ovarian cancer has four defines stages: I, II, III and IV, each of which has three sub stages except for stage IV. Stage I involves the growth of the cancer that is limited to the ovary or ovaries; Stage II is the growth of the cancer in one or both of the ovaries with an extension into other pelvic organs; Stage III has growth one or both of the ovaries and either the cancer has spread to the lining of the abdomen or to the lymph nodes; Stage IV is the most advanced stage of ovarian cancer with growth in one or both of the ovaries and distant metastases, which is growth outside of the peritoneal cavity (NOCC 2006).

During the last ten years, scientists have discovered two genes that greatly increase a woman's chances of developing ovarian cancer. These genes are

BRCA1 (Breast Cancer 1), and BRCA2 (Breast Cancer 2). It is believed that these genes were tumor suppressor genes, but scientists now believe that these two genes are involved with the mismatch repair functionality in genes (Fergus 2000a). The mismatch repair function corrects mistakes in DNA, which can cause a gene to stop producing a protein, or even prevent that protein from functioning properly. When either of these two genes is mutated, the normal functionality of BRCA1 or BRCA2 is reduced, and thus cannot prevent a cell from becoming cancerous.

Ovarian cancer is not the most common cancer among women, but it has proven to be the most difficult to detect. When the ovarian cancer is detected in Stage I and treated, the five year survival rate can be as high as 92.1%. If it is detected in Stage IV and treated, the five year survival rate is as low as 11.6% (Unknown 2002). There are three main methods that are used today to screen for ovarian cancer: Pelvic/rectal examination, ultrasound and the CA-125 blood test. These tests are not consistently reliable, or accurate in screening for ovarian cancer, and they are very poor in determining ovarian cancer in the early stages of the disease. The pelvic/rectal exam involves the physician feeling the uterus and ovaries to find any abnormalities in their shape or size (NOCC 2006). An ultrasound uses sound waves to create pictures of the ovaries to determine if the tissue is healthy, or contains fluid-filled cysts, or has tumors. The CA-125 blood test detects the levels of the CA-125 cancer antigen protein levels in the bloodstream. When a woman has ovarian cancer, the amount of CA-125 tends to rise above 35 u/ml. This is not a definitive test, because non-cancer patients can test positive to this test, and cancer patients may not produce enough CA-125 to produce a positive response to the test (Unknown 2002).

"Mass spectrometry (MS) is a powerful tool for determining the masses of biomolecules and biomolecular fragments present in a complex sample mixture." (Lilien 2003) A mass spectrum is created using a mass spectrometer, which is composed of three components: ionizer, mass analyzer and the ion detector (Yates 2000). The ionizer converts the molecules of a sample into ions by using a laser to excite the molecules into a gaseous phase. These excited molecules are separated by passing through a magnetic field until they collide with an ion detector. The mass spectrometer determines the mass to charge ratio for the molecule based upon the electric current that is generated when the molecule strikes the ion detector along with the time of flight from the ionizer to the detector. The result of the mass spectrometer is a set of mass to charge (m/z) values that match up with relative intensities based upon the number of molecules that strike the detector with a particular m/z value. When a mass spectrum is relatively small, a visual inspection can be used, but as the number of protein fragments in the sample increases, it is necessary to utilize algorithms to detect the presence or lack of presence of particular molecules in a sample.

Many techniques have been proposed to deal with detection of cancer from high dimensional of mass spectrum data. These include biomarker detection, decision trees, peak detection, and principle component analysis. In the proposed algorithm, we utilize Gabor filters, which have the

> "… advantage of (1) maximizing joint localization in both spatial and frequency domains; (2) flexibility; [Gabor functions] can be freely tuned to a continuum of spatial positions, frequencies and orientations, using arbitrary bandwidths…." (O. Nestares 1998)

The Gabor filters transform a mass spectrum data matrix of size *n*-by-*m* into a matrix of n-by-(m- (2 * filter size)) where each value in the matrix is a complex number that falls into one of the four phases of a Cartesian coordinate system. In the proposed algorithm, the phases are quantized and coded to identify the quadrant where the coefficient *a* + *bi* from the output matrix resides. Using the coded values, a two-bit encoding scheme creates a binary string that represents the characteristics of that patient (ovarian cancer positive or negative). By combining all of the binary coded strings of the ovarian cancer patients into one pattern using a voting scheme with a threshold criterion, it is possible to predict blinded testing samples for ovarian cancer. In a similar manner, the control samples will be combined to create the overall pattern that is indicative for ovarian cancer negative patients.

Importance of Subject

As described above, the ability to predict ovarian cancer while in Stage I of the disease is critical to ensure the highest five year survival rate. Current screening techniques (Pelvic/rectal examination, ultrasound and the CA-125 blood test) have proven to be very ineffective in detecting ovarian cancer during its early stages. For example, the CA125 test is only able to predict 40 to 50% of early ovarian cancers (Check 2002). The actual incidence of ovarian cancer in the typical population is very low, only 40 to 50 cases per 100,000 women. With this in mind, it is easy to see that any effective detection technique must minimize the number of false positives. In addition, the technique must be able to correctly identify the benign samples with a similar effective rate. Dr. Steven Skates, a biostatistician at Massachusetts General Hospital and assistant professor of medicine at Harvard Medical School, has determined that to achieve the low rate of false positives and false negatives, the specificity of the technique must exceed 99.6% and the sensitivity must exceed 98% (Check 2002). The proposed algorithm will shown that it can meet these two criteria by tuning the frequency and variance of the Gabor filter along with the threshold value used in the voting scheme for coding the ovarian cancer and control pattern binary codes.

It is important to note that in order to discover ovarian cancer in women in Stage I, it will require both doctors and women to be very diligent in testing on a regular basis. This testing should include women who have family history of ovarian cancer, and women who do not have any family history of ovarian cancer. With the amount of testing necessary to cover the women who are at risk for ovarian cancer, it is critical to produce a test that is both cost and efficient in predicting the disease.   In addition, it is important to utilize a test that provides a sample to the lab with the least amount in invasiveness for the women, so that that individual will not be reluctant to return for testing in future years. Our testing uses blood samples, which many would agree is more pleasant than a pelvic exam.

CHAPTER TWO: BACKGROUND

Detection Algorithms for Mass Spectrometry

This section of the literature review will discuss some the existing techniques being used to analyze mass spectrum data. In order to identify whether a sample of ovarian tissue tests positive for cancer, there needs to be an identifying characteristic in the mass spectrometry data that differentiates the sample. One of the major hurdles in mass spectrometry at this time is the ability to handle large numbers of protein fragments in a sample, and over the last few years, many algorithms have been created to solve this problem.

The goal of a mass spectrometry classification algorithm is the discrimination of one condition from another by analyzing the mass spectra (Lilien 2003). Many types of mass spectrometry classification algorithms (MSCAs) have been developed to detect disease in humans, as well as observe the changes in those diseases (Austen 2000), (Petricoin 2002), (Paweletz 2000) and (Ball 2002).   An MCSA can be categorized by the following items: type of mass spectrometry data, type of algorithm used on the MS data, and the method for classification verification.

The type of mass spectrometry data used in an experiment can be broken down into three components: completeness of the spectrum, manual preprocessing and the source of the sample. If the entire mass spectra are used, which means that all of the *m/z* values from zero to the upper detectable boundary, then the data sets are considered to be complete spectra, otherwise it is labeled as a partial spectra. When the spectra are normalized or if sections of the spectrum have been removed, then the data has been preprocessed. In most cases this is true due to the introduction of noise and inconsistencies of samples by the same piece of MS equipment. Finally, the type of data set used in an experiment can be the result of simple of complex fragments. Again, the majority of the research into MSCAs has utilized complex fragments that result in tens of thousands of peaks.

In terms of the types of algorithms used for classification of mass spectrometry data, there are two main types: heuristic and exact classifications. A heuristic algorithm goes through multiple iterations until it converges on a classifier. Examples of this type are genetic algorithms, neural networks and simulated annealing. On the other hand, exact classification algorithms are "… computationally efficient; they are noniterative and deterministic (i.e., always compute the same solution.) (Lilien 2003)." Some examples of an exact classification are linear discriminant analysis, principal component analysis (PCA), and the Gabor phase quantization algorithm introduced in this paper.

Finally, the method of classification for the algorithm must be confirmed in order to determine the effectiveness of the algorithm to detect future unknown samples. A technique called the leave-out experiment involves splitting the data set into training set and a testing set and repeating that process multiple times. In some cases, one of the split data sets will perform better than the others, and it is important to note that difference in your report. When the testing set contains only one or very few samples, it is considered to be partial. The following paragraphs describe specific examples of MSCAs.

(Lilien 2003) used principal component analysis (PCA) to reduce the dimensionality of the feature set followed by a linear discriminant analysis (LDA) to project the spectrum onto a surface. This projected surface is then used as the criterion for classification for the test data, which is also dimensionally reduced using PCA and then projected onto a surface using the LDA. Using a classification confidence, the sample is either classified as healthy or diseased.

Unlike PCA, which is an unsupervised pattern recognition technique that tends to lose resolution of spectra between proteins as the data set increases in size, (Wagner 2002) was able to show that the use of a supervised learning technique version of PCA called discriminant principal component analysis (DPCA) along with linear discriminant analysis (LDA) could effectively discriminate static time-of-flight secondary ion mass spectrometry data. In addition, they were able to limit the number of misclassifications of the spectral data using just the LDA when compared to the use of either PCA or DPCA on the same data set.

(Li 2002) used a ranking algorithm to determine a large number of peaks in the mass spectrometry data of breast cancer patients that could identify biomarkers according to two diagnostic groups. They determined that there were three top peaks that could be used for biomarkers for breast cancer (BC1, BC2 and BC3). Using multivariate regression, the three biomarkers were combined into a single composite index. Using a boot-strap cross validation, they were able to improve the diagnostic power (area under the curve) for the single composite as compared with the individual biomarkers.

Another group, (Yasui 2003), used SELDI-TOF equipment to create high dimensionality mass spectrometry data. Their research was two-fold: (1) separate protein signals from the background noise of the intensity signal to identify biomarkers to discriminate prostrate samples form control samples and (2) calibrate the protein mass/charge measurements across samples. The latter object is an issue that we addressed in our research, which is the reason why we used the smoothing technique to minimize the fluctuation of the mass spectrometry data from sample to sample.

(Petricoin 2002) used ovarian cancer mass spectrometry data that contained about 15,200 mass to charge values. Using a genetic algorithm and cluster analysis, they created a pattern of about five to twenty key proteins in their training model to differentiate the cancer and non-cancer samples. Using this model, they performed the same genetic algorithm and cluster analysis on a test sample in order to verify the model. The results of this model were 100% sensitivity, 95% specificity, and a positive predictive value of 94%, which is much higher than the 35% predictive value of the CA-125 blood test.

Utilizing decision trees and a boosting algorithm, (Qu 2002) and (Wagner 2002) were able to effectively use SELDI mass spectrometry data to differentiate prostrate cancer samples from noncancer samples. Traditionally, learning algorithms are susceptible to overfitting, which means that the algorithm works well for that particular set of data, but when it is used on a new data set it fails to predict the cancer form the noncancer samples. By using boosting, they were able to increase the minimize margin, so that the chances of misclassifying the test samples were decreased.

In (Wang 2004) they used mass spectrometry to analyze a three marker panel of transthyretin, full-length apolopoprotein A1, and an internal fragment of inter-α-trypsin inhibitor chain 4 (ITIH4). The purpose of their study was to show that the use of mass spectrometry immunoassay would be more effective in distinguishing modified forms of proteins then the traditional immunoassays. Using a 96-well filter plate to prepare the three markers, they sent the samples through a mass spectrometer for each of the three proteins. The experiment showed that the mass spectrometry immunoassay was able to simultaneously quantify and distinguish multiple forms of the transthyretin, as well as identify the single peak of the ITIH4 along with the smaller peptides.

A slightly different approach was taken by (Wu 2003) to analyze mass spectrometry data. Instead of using one type of classification algorithm, they decided to analyze four different types of algorithms: bagging, arc-fs, arc-x4, and random forest. The bagging algorithm involves creating classification trees by recursively splitting subsets of the data until a terminal node has been reached, which is labeled as a final classifier. Both arc-fs and arc-x4 utilize the concept of boosting, which is an adaptive re-sampling of the original data, so that the weights are increased for the frequently misclassified samples. Arc-fs uses a weighted voting scheme, while the arc-x4 uses a un-weighted voting scheme. Finally, the random forest blends two machine learning mechanisms: bagging and random feature selection, which increases the predictive accuracy. After running each of the algorithms, they discovered that the random forest technique provided the lowest misclassification as well as a more stable assessment of the errors in the classification.

Gabor Filters

This section describes the use of Gabor filters to detect patterns in two or three dimensional images that can be used to discriminate one image from another image. Mass spectrometry data produces a signal that can be graphed into a two-dimensional image on an X-axis (mass to charge values), and on the Y-axis (intensity). This signal provides an excellent opportunity to make use of Gabor filters, which can extract spatially localized spectral features (Prasad 2005). Gabor filters are created using a one dimensional sinusoid that is modulated with a Gaussian, which is a symmetrical frequency distribution that has a precise mathematical formula that is related to the mean and standard deviation of the sample (Caldwell 2006). One dimensional Gabor filters are calculated using a variance of the Gaussian and a frequency, while two dimensional Gabor filters use the variance of Gaussian, frequency and orientation.

Gabor filters have been used for image processing, such as image coding and compression, and analysis of texture. (O. Nestares 1998) propose an optimized solution for spatial implementation of the Gabor scheme. This optimized solution allows for an improvement in the quality of image reconstruction, so that the reconstructed images are visually indistinguishable form the original image.

(Daugman 2004) proposes the use of a two dimensional Gabor filter for iris pattern recognition. Unlike the face, which has a variety of expressions, the iris is more consistent and varies greatly between individuals, and the iris is not sensitive to the angle of the illumination to produce a high quality image. Each iris pattern is demodulated to extract the phase information using the two-dimensional Gabor filter. The phases are coded to identify the location of a given area of the iris by using a binary system ((1,1), (1,0), (0,1) and (0,0)). This process was repeated until 2048 bits were obtained. Irises of two different individuals were compared by sending the 2048 bits into the Hamming distance formula, which calculates the number of differences that exist between two binary numbers. The ability to decide if two irises were from the same individual was achieved by having the smallest Hamming distance value. In fact, 50% of the image comparisons between the same irises had a Hamming distance of zero, and an average hamming distance of 0.019.

(Lepistö 2003) used Gabor filters to discriminate textures and color in images. Their algorithm used multi-resolution Gabor filters to be applied to a hue, saturation and intensity channels (HSI) color space. The Gabor filters were applied to each color channel separately to create a feature vector, which was then combined with the other color channels into one vector that was used for classification based upon the k-nearest neighbor.

Research Hypotheses

Detecting ovarian cancer in its earliest stages is critical to the five year survival rate of women who are positively diagnosed with the disease. Current detection methods have fallen short in effectively revealing ovarian cancer while it is still in Stage I or Stage II.

Hypothesis:

H1: Using a one-dimensional Gabor filter along with a binary coding scheme, it will be possible to detect ovarian cancer with a sensitivity and specificity rate greater than 90%.

CHAPTER THREE: METHODOLOGY

Software

The mass spectrum data used in this research was stored in two sets of Excel™ files: one for the ovarian cancer samples and one for the control samples were retrieved from the medical school at Northwestern University in Chicago, IL. Each file contained two columns that contained the mass to charge values and the corresponding intensity value. Using a script written for Matlab™, the intensity values for all of the ovarian cancer samples for mass to charge values between 500 to 11,000 m/z were extracted into training and testing data files. The same procedure was used to create the testing and training data files for the control samples. Matlab™ was also used to run a script file that process and analyze the data files. The Matlab™ related script files can be found in ***Appendix A*** and ***Appendix B***.

Background

SELDI-TOF Mass Spectrometry

The ovarian cancer data was collected from patients' serum samples and their mass spectrum was generated by using Surface-Enhanced Laser Desorption and Ionization - Time of Flight (SELDI-TOF) instrument. The SELDI-TOF uses a protein chip array whose surface acquires proteins with the help of special protein docking sites that are either biologically or chemically created. After the chip has been created and washed, the desired sample is crystallized with energy absorbing molecules that serve to absorb the energy from the laser and ionize the protein (Yasui 2003). Each protein fragment will then fly through the flight tube to the detector at the other end based upon the molecular weight of the protein and its associated charge. When the protein fragment reaches the detector, the intensity at that instant in time is recorded into a data pair along with the mass and the charge for that protein. The ovarian and control sample data produced over 48,000 unique data pairs over the mass/charge range of 0 – 30,000. For our analysis, we used the mass/charge ratios of 500 to 11,000, which gave us over 17,500 data points. These data points were utilized in our algorithm for classifying ovarian versus non-ovarian cancer samples. Figure 1 illustrates the SELDI-TOF process.

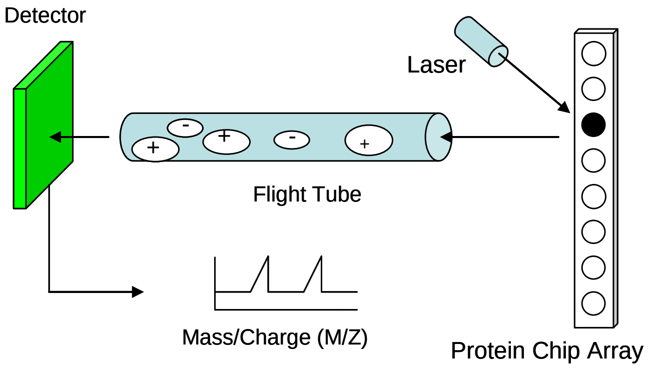


Figure 1: SELDI-TOF Process

Gabor Wavelets and Phase Quantization

Gabor filters have been used for image processing, such as image coding and compression, and analysis of texture (Daugman 2004). (O. Nestares 1998) proposed an optimized solution for spatial implementation using a Gabor scheme. This optimized solution allows for an improvement in the quality of image reconstruction, so that the reconstructed images are visually indistinguishable form the original image. Gabor filters have been shown to respond to optimal localization properties in the spatial and frequency domains, which is shown in Figure 2 from (Prasad 2005) .

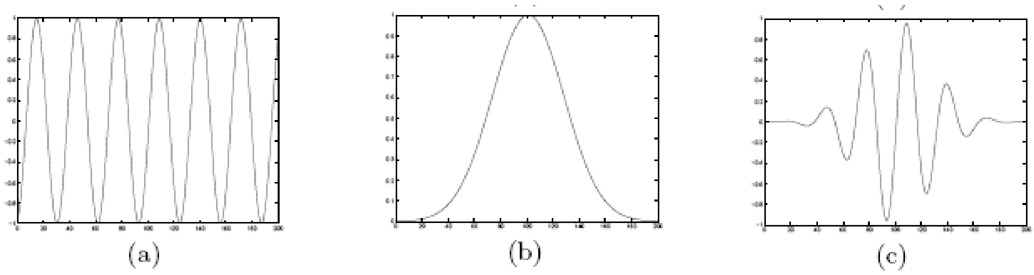


Figure 2: Gabor filter composition for 1D signals: (a) sinusoid, (b) a Gaussian kernel, (c) the corresponding Gabor filter.

In this paper, a Gabor filter bank is created using a one dimensional sinusoid that is modulated with a Gaussian, which is a symmetrical frequency distribution that has a precise mathematical formula that is related to the mean and standard deviation of the sample (Caldwell 2006). For 1D mass spectrum analysis, we applied the one dimensional Gabor filter bank, which requires four parameters including (i) an *n*-by-*m* matrix for *n* observations and *m* features, (ii) variance along the *x* and *y*-axes respectively, (iii) center frequencies along the *x* and *y*-axes respectively, and (iv) the size of filter. Equation 1 shows the 1-D Gabor wavelet is:

$$G(x,\omega,\sigma_x) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{(-\frac{x^2}{2\sigma_x{}^2})} e^{j\omega\pi} \tag{1}$$

where σ_x is variance, ω is frequency. In order to determine the most effective combination, we used multiple combinations of the variance and the frequency (5 and 5, 5 and 10, 10 and 5, 10 and 10 respectively) to construct filter bank. The output vector after Gabor convolution is a matrix of size *n* by (*m*–(2*filter size)). Each Gabor coefficient after the convolution is a complex number and it falls into one of the four phases in Cartesian coordinate system. These phases are quantized and coded to identify the quadrant where the coefficient *a* + *bi* resides, which is shown in Figure 3.

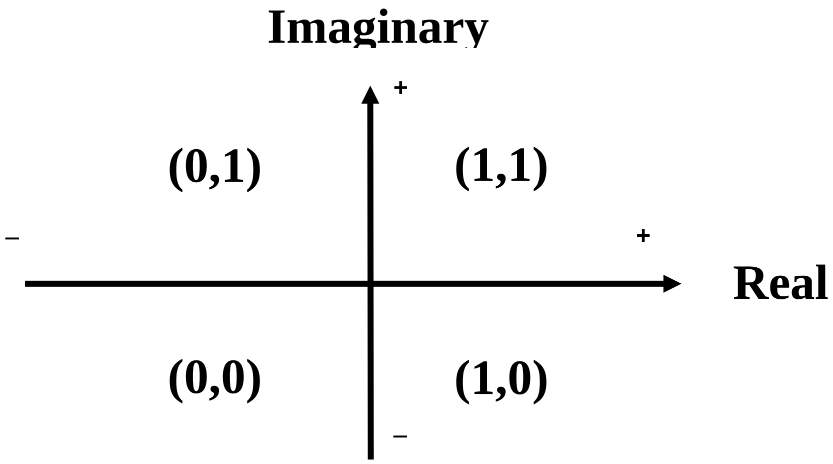


Figure 3: Four Phase Quantization Diagram

The two-bit encoding scheme used in the algorithm is based on the signs of real part and imagery part of Gabor coefficient, such as (*sgn*(*a), sgn*(*b*)) where

$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

i.e. the real and imaginary part of coefficients can be represented by one of four quadrants: (1,1), (1,0), (0,1) and (0,0) as shown in Figure 3. For example, a spectrum vector (3.2-0.2 *i*, -5.0 + 0.7 *i*, -4.6 -8.5*i, ...*) can be converted to a binary string of (10, 01, 00, …).

Hamming Distance

Hamming distance is defined to be the "distance between binary datawords c_1 and c_2, denoted by $d(c_1,c_2)$ to be the minimum number of bits that must be "flipped" to go from one word to the other (Johnson 2003)." When a dataword is transferred over a medium errors occur and the dataword that is received on the other end is different than the original dataword. By using the minimal Hamming distance, it is possible to create a dataword that closely resembles the original dataword (Bhatti 1995). For example, if the original dataword was 11000, but the received dataword was 11001, then the minimal Hamming distance would be one. In this case, the last one would be flipped to become a zero, and then the original dataword is recreated. We use Hamming distance to determine how closely a testing sample reflects the binary code of the two types of training codes (cancer and control). If the Hamming distance between the test sample and the training cancer binary code is smaller when compared to the Hamming distance of the test sample and the training control binary code, the sample is labeled as cancer positive. The test sample would be labeled as cancer negative if the opposite is true.

Procedures

The data used in this project was provided by the National Ovarian Cancer Early Detection Program (NOCEDP) clinic at Northwestern University Hospital in Chicago, IL. A total of 253 samples consisting of 162 with ovarian cancer and 91 normal samples were provided with a feature set of well over 13,500 peaks after the first 2000 peaks were truncated. This truncation was due to the low M/Z values, and the fact that the 'energy-absorbing-molecules' used in the creation of mass spectrometry data distort the intensity of the proteins below the 2000 peak level (Yasui 2003). For this project, we used a subset of the ovarian (120 patients) and normal (65 patients) data samples during the first run in order to get a sense of the effectiveness of our algorithm. Figure 4 shows the gel views of both cancel and normal serum samples.

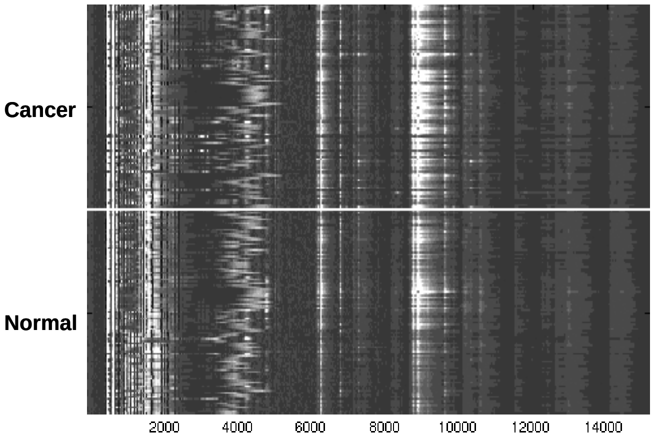
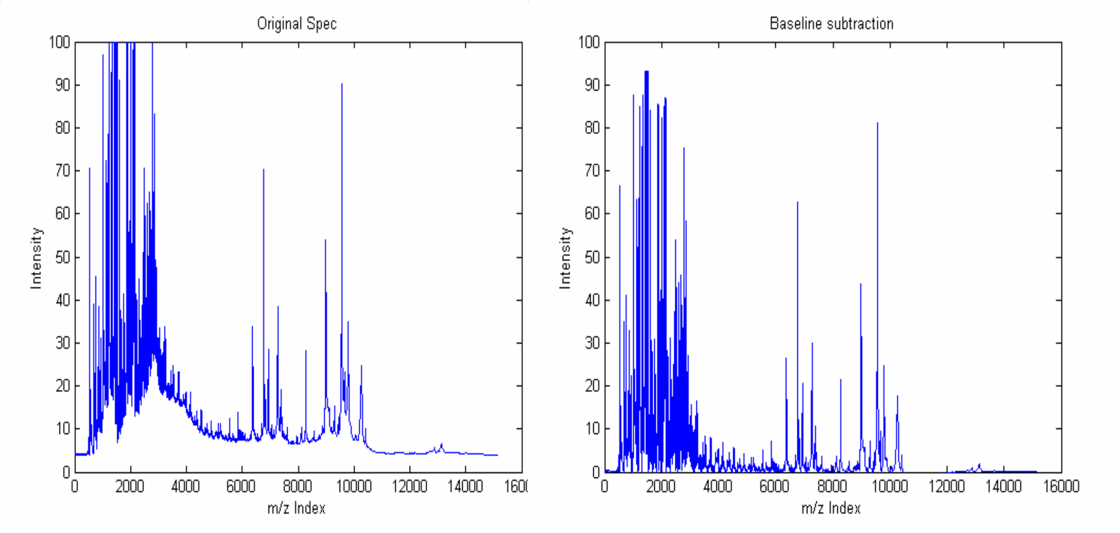


Figure 4: Gel view of cancer and normal serum samples

The data set from the mass spectrometry data was split into two parts: a training set and a testing set. The training set for the ovarian cancer group used 2/3 of the original ovarian cancer samples, and the control group used 2/3 of the original non-ovarian cancer samples. For the testing set for the ovarian cancer group, we used the remaining 1/3 of the original ovarian cancer samples, and the testing control group used the remaining 1/3 of the original non-ovarian cancer samples. It should be noted that the ovarian cancer data and the control group (non-ovarian cancer) data were not mixed in this study, because we wanted to create a pattern that would indicate just ovarian cancer or non-ovarian cancer. In order to prevent overfitting, the original data set was split in a similar manner two more times using different combinations of the ovarian cancer samples and the control samples.

After the data set was split into two sets, the data was preprocessed using a baselining technique and a smoothing function to remove some of the noise from the data. The data was baselined using baseline subtraction, which is calculated using a window size of 128 points and three functions. The functions in order are min, mean and median, which resulted in the finalized baselined data. In addition to the baselining, we used a smoothing function that is the overlapping averaging on the range to each data point with a window size again of 128 data points. The preprocessing of the data was executed on both the training and testing sets. **Figure 5** illustrates the spectra with (b) and without (a) baseline subtraction process.



(a)

(b)

Figure 5: (a) Before and (b) After baseline subtraction

Following the preprocessing of the data, we applied a 1-D Gabor filter with different combinations of the frequency and variance to produce the phase data. At this point in our algorithm, we need to create a code from the output phase data for the training ovarian samples and another code for the training control samples, which will be used to determine if a sample is ovarian cancer positive or negative. As we have described above, the output of the 1-D Gabor filter will be a real number along with an imaginary number, each of which will have either a positive or negative sign. Based upon the sign of the each component, we assign a value of one or zero as described above in Equation 2. This process continues for each feature in the training cancer sample, which results in a matrix of binary strings. In order to generate a coded binary pattern that is indicative of a positive cancer sample, we use a voting scheme to evaluate the consistency of the bits with in the same column of the sample. The voting scheme is based upon a threshold. When the number of ones in a column exceeds the threshold, that column is labeled as a one, and the same is true when the number of zeroes in a column exceeds the threshold. If neither the number of ones nor zeroes exceeds the threshold, then the column is marked as indeterminate (x). In the program, the number of zeroes was checked first to see if it exceeded the threshold value, and then the number of ones was tested. To ensure that this did not bias towards zero when the threshold value was small (<30%), the program was modified to examine the ones first and then the zeroes. With this modification, the results were the same. After the cancer samples in the training set have been evaluated, the non-cancer (control) training samples are processed in the same manner. **Table 1** provides an example using a threshold of 80%.

Table 1: Pattern calculation example

Sample #	Feature #1	Feature #2	Feature #3	Feature #4	Feature #5	Feature #6
1	1	0	1	1	0	1
2	1	1	0	0	0	1
3	0	0	1	1	0	0
4	1	0	1	1	0	0
5	1	0	0	1	1	1
Result	**1**	**0**	**x**	**1**	**0**	**x**

Once the patterns have been created for both the ovarian cancer samples and the non-ovarian cancer samples in the training set, the next step is to process the testing set in a similar manner to the training set. The samples will again use the 1-D Gabor filter, and then the output will be coded with either be a zero, one or a symbol **x** to create a binary string for that sample.

At this point, the testing set will processing will diverge from the training set. Each coded binary string in the testing set was individually compared against the coded patterns created for the ovarian cancer and non-ovarian cancer training samples. A coded binary string can be considered to represent the cluster of cancer samples or the cluster of control samples. In order to determine if an unknown test sample belongs to one cluster or the other, a measurement must be performed to determine if the test sample lies closer to the cancer or control cluster. To make this determination, we used the Hamming distance. The Hamming distance, which is defined as the number of differences of symbols between two string inputs, was used to compare the two binary strings. If the

Hamming distance of the testing string and the ovarian training string was smaller than the Hamming distance of the testing string and the non-ovarian training string, then that testing string was labeled a positive for ovarian cancer. The testing sample was a labeled negative if the opposite was true. When all of the testing set (ovarian and control) was evaluated, the sensitivity, specificity and accuracy measurements were calculated to determine the effectiveness of our coded pattern algorithm. **Figure 6** summarizes the overall process for serum protein profiling analysis based on mass spectrum data sets.

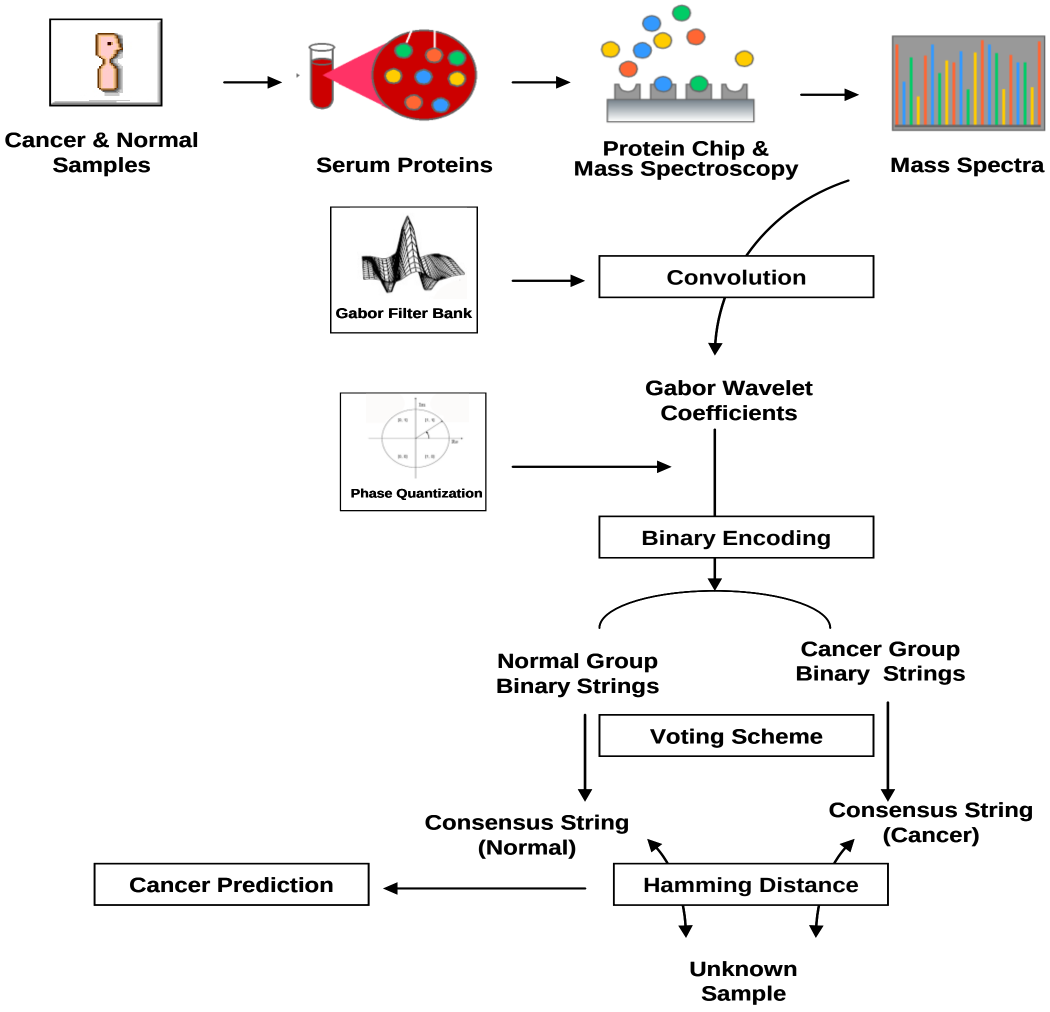
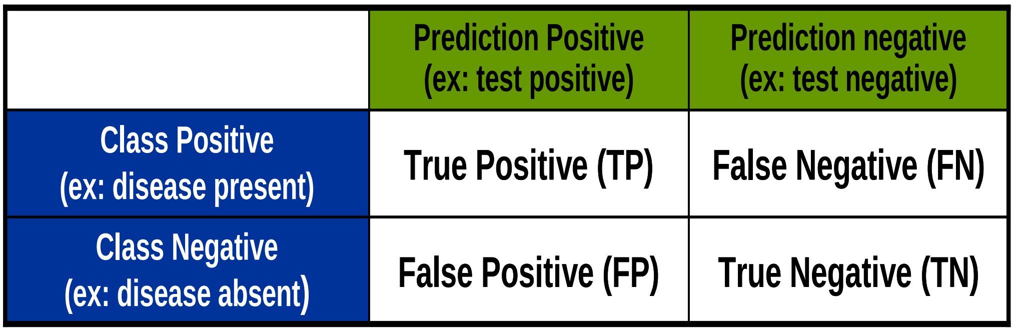


Figure 6: System Architecture for Serum Protein Profiling Analysis

Analysis

The analysis of the results from the serum protein profiling algorithm involved using the technique of a confusion matrix, which is described in the following section.

A confusion matrix contains information about actual and predicted classifications that are generated by a classification system. In the confusion matrix, there are four possible outcomes: true positive (TP), false negative (FN), false positive and true negative (TN). True positive and a true negative are obviously correct predictions. The false positive is the result of predicting an outcome as positive when it is in fact negative, and a false negative is a negative prediction when the actual result is positive. These values are described in Figure 7.



	Prediction Positive (ex: test positive)	Prediction negative (ex: test negative)
Class Positive (ex: disease present)	True Positive (TP)	False Negative (FN)
Class Negative (ex: disease absent)	False Positive (FP)	True Negative (TN)

Figure 7: Confusion Matrix

Sensitivity, which is the number of patients with disease that have a positive test result, is calculated as TP/(TP + FN). Specificity is defined as the number of patients who do not have disease that have a negative test result is calculated as TN/(TN + FP). Finally, accuracy of successfully predicting both the patients with disease and those without disease is calculated as (TP + TN)/(TP +TN + FP +FN).

For any algorithm, acquiring 100% accuracy is very difficult to achieve, but the number of false positives and false negatives must be minimized. As Figure 8 from (Tape) indicates, the overlap of normal and diseased patients produces a number of false positive and false negative results for any particular criteria shown as the black line in the figure. As the black line is moved to the right, the number of false positives is reduced, but the number of false negatives is increased. If the black line is moved to the left, then the number of false positives increase, and the number of false negatives decreases. For ovarian cancer screening, it is important to weigh the effect of telling someone that they are positive for cancer and must seek treatment when it is not needed, or informing someone that they are negative for the disease when in fact the test is not correct.

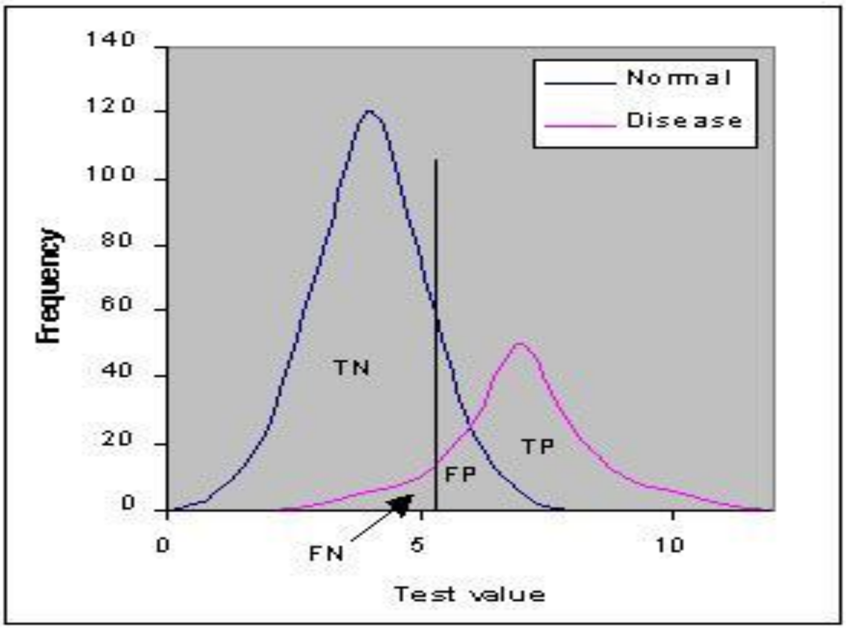


Figure 8: Example of Diagnostic Results

CHAPTER FOUR: RESULTS

The results are reported in three main sections. The first section presents the results from the first data set combination, the second section presents the results from the second data set combination, and the last section presents the third data set combination.

Data Set One

As stated above, multiple combinations of variance and frequency for the 1-D Gabor filter were used to determine which combination produced the best results for the first pass of the data using 120 ovarian cancer samples and 65 control samples as shown in Table 2.

Table 2: Data Set One Results

Configuration	Sensitivity	Specificity	Accuracy
Variance = 5 Frequency= 5	97%	92%	95%
Variance = 5 Frequency= 10	95%	100%	97%
Variance = 10 Frequency= 5	97%	96%	97%
Variance = 10 Frequency= 10	100%	100%	100%

As we can see from this table, the best results were a result of using a variance of 10 and a center frequency of 10. Missing from the previous table is the fact that we initially used a threshold value of 85% to determine the code pattern from the ovarian cancer and the non-ovarian cancer samples in the training set. With this in mind, we wanted to see if different threshold values for determining the coded pattern would affect the results from the best entry in Table 2. The values of the threshold ranged from 50% to 90%, and the sensitivity, specificity and the accuracy values were calculated and are shown in Table 3.

Table 3: Threshold Comparison for Data Set One

Threshold	Accuracy	Sensitivity	Specificity
50%	95%	66%	82%
60%	96%	72%	86%
65%	87%	96%	74%
70%	96%	85%	92%
75%	94%	100%	83%
80%	100%	86%	95%
85%	100%	100%	100%
90%	100%	100%	100%

Looking at the table, you can see that the threshold value of 85% maximizes all of the criteria that we have been using to determine the effectiveness of our coded pattern algorithm along with the variance of 10 and the frequency of 10 for the 1-D Gabor filter. Beyond the 85% threshold, the sensitivity, specificity and accuracy have the same values as those at a threshold of 85%. The standard deviation (SD) values for the sensitivity, specificity and accuracy for data set combination one are shown in Table 4.

Table 4: Standard Deviation for Data Set One

	Sensitivity	Specificity	Accuracy
Standard Deviation (SD)	2.06	3.86	2.06

Data Set Two

The results from data set combination two are very similar to those of data set combination one. Examining Table 5, it is apparent that the variance/frequency combination of 10 and 10 respectively has produced the best results, but in contrast to the data set combination one, the variance of 10 and frequency of 5 has also produced excellent results.

Table 5: Data Set Two Results

Configuration	Sensitivity	Specificity	Accuracy
Variance = 5 Frequency= 5	96%	85%	92%
Variance = 5 Frequency= 10	96%	91%	94%
Variance = 10 Frequency= 5	98%	91%	95%
Variance = 10 Frequency= 10	98%	91%	95%

Similar to the SD values from data set combination one, the SD values for this set are similar for the sensitivity and the accuracy and about 2 times the value for the specificity as shown in the following table.

Table 6: Standard Deviation for Data Set Two

	Sensitivity	Specificity	Accuracy
Standard Deviation (SD)	1.15	3.00	1.41

Data Set Three

The results from data set combination three produced slightly different patterns than the two previous sets with the variance/frequency combination of 10 and 5 respectively creating the best results.

Table 7: Data Set Three Results

Configuration	Sensitivity	Specificity	Accuracy
Variance = 5 **Frequency= 5**	91%	93%	92%
Variance = 5 **Frequency= 10**	88%	96%	91%
Variance = 10 **Frequency= 5**	98%	93%	91%
Variance = 10 **Frequency= 10**	88%	96%	91%

Standard deviation values for the sensitivity, specificity and accuracy for data set combination three are shown in the Table 8.

Table 8: Standard Deviation for Data Set Three

	Sensitivity	Specificity	Accuracy
Standard Deviation (SD)	4.72	1.73	0.5

Overall Findings

Using all of the combinations of variance and frequency on all three of the data sets, the following average sensitivity, specificity and accuracy values were calculated along with their corresponding standard deviation values are detailed in Table 9.

Table 9: Overall Sensitivity, Specificity and Accuracy Values

	Sensitivity	**Specificity**	**Accuracy**
All values of variance and frequency	95.17%	92%	94.17%
Standard Deviation (SD)	4.00	4.23	2.89

Using only the variance value of 10 and the frequency value of 10 as inputs for the 1-D Gabor filter for training sets one, two and three, the following overall results were observed.

Table 10: Overall Results for Variance=10, Frequency=10

	Sensitivity	**Specificity**	**Accuracy**
Variance = 10 **Frequency= 10**	95.33%	95.67%	95.33%
Standard Deviation (SD)	6.43	4.51	4.51

CHAPTER FIVE: DISCUSSION

This section discusses the results reported in chapter four, which includes the explanation of the outcomes from the confusion matrix.

Explanation of Outcomes

Our coded pattern algorithm produced highly successful results for determining whether a given sample was positive for ovarian cancer. In the first data set combination the results for the sensitivity, specificity and accuracy ranged from 66% to 100% with the best results being achieved when the Gabor filter used the variance value of 10 and the frequency value of 10. Since this algorithm depends on a voting threshold to determine the binary value of each feature column, it is important to see if the using the arbitrary threshold of 85% really produced the best results for a variance of 10 and a frequency of 10. As is indicated in Table 3, a threshold value of 60% and below produced poor results, while a threshold value of 85% and above produced an accuracy of 100%.

The results were consistent across the other two training/testing sets with specificity, sensitivity and accuracy values in the mid to high 90% range. It is worth noting that in data set combination three that the best results for the sensitivity (the ability to detect cancer positive samples) were obtained using a variance of 10 and a frequency of 5. The third data set also produced the lowest accuracy values out of all three combination data sets, which (Lilien 2003) has reported as a consequence of using a leave-out experiment to validate your classification discriminant.

Looking at the all three data set combinations and all of the combinations of variance and frequency, the sensitivity, specificity and the accuracy were 95.17%, 92% and 94,17% respectively. This leads one to consider the use of multiple Gabor filters across one spectra, but that work for the future and it will require a more iterative approach in order to determine which filter is the most successful for each data point in the spectra. Since we have determined that the variance/frequency combination of 10/10 has produced the best results across the three data sets, looking at the values in Table 10 reinforces the notion that the proposed algorithm is capable of predicting ovarian and control samples with a rate of greater than 95%.

The final evaluation of any algorithm is to compare it with any previous techniques to see if any improvement has been made. The CA125 has been the best known serum marker for ovarian cancer and has been studied for many years. By analyzing the levels of CA125 in the sample, the test is only able to detect ovarian at a rate of about 40% to 50%. In the last five years, scientists have been using a "..statistical model [that] converts the longitudinal CA125 profile to a single number called 'risk of ovarian cancer', which is the risk of having the disease at a given time (Unknown 2002)." This modified cA125 test recorded better results, but it was still only able to increase the sensitivity to a value between 70% and 85% while maintaining the specificity. (Yasui 2003) used a boosting algorithm that only produced sensitivity and specificity rates between

73% and 86%. This is an improvement over the CA125 test, but still does not reach the levels needed to prevent false test results. In contrast to the two previous examples, (Lilien 2003) was able to achieve sensitivity and specificity rates of above 97% using principal component analysis followed by linear discriminant analysis. The downside to this algorithm is that it is more computationally intensive and complex than our proposed algorithm. They showed that their training run time was $O(n^3 + n^2 r)$ and the testing runtime was O(mrn), which equates to about one to one and half minutes on a Pentium™ 4 workstation.

Chapter Six: Conclusion

Detecting the ovarian cancer in its early stages gives the patient a much higher five year survival rate, which has given rise to a need to find an efficient and an effective early detection mechanism for ovarian cancer. There are three existing methods that are used today to screen for ovarian cancer: Pelvic/rectal examination, ultrasound and the CA-125 blood test. These tests are not consistently reliable, or accurate in screening for ovarian cancer, and they are very poor in determining ovarian cancer in the early stages of the disease. Although scientists have discovered two genes, BRCA1 (Breast Cancer 1) and BRCA2 (Breast Cancer 2), that greatly increase a woman's chances of developing ovarian cancer, most BioInformaticists are using proteomic data to determine the presence of ovarian cancer.

Based upon the criteria for a mass spectrometry classification algorithm, our proposed algorithm has the following characteristics: partial spectra, manually preprocessed, complex fragment mixtures, exact classification and used a multiple leave-out experiment for classification. Following the examples of (Petricoin 2002) and (Yasui 2003), we only a portion of the entire mass spectra data, and in particular, we used *m*/*z* values of 500 to 11,000. Below the *m*/*z* value of 500, the energy absorbing molecules that attach to the desired proteins on the protein chip distort the data. On the other end of the spectra, the *m*/*z* values above 11,000 had very minimal intensities that did not produce any features for the data set that made a difference to the final results. This data was normalized between zero and one and was also smoothed to ensure that there were no discrepancies. As explained above, our algorithm is considered an exact classification technique, which is capable of producing the same solution and is noniterative. Finally, the data was split into three training/testing sets to ensure that the algorithm did not fall into the overfitting trap.

Limitations

Due to the limitations of the knowledge of the data set, we are not able to discern whether or not all or some of the ovarian cancer samples were from women in Stage I of the disease. If the samples turn out to be from women who are in the latter stages of ovarian cancer, then it will be necessary to acquire a known ovarian cancer data set where the number of women in each stage is known. This will allow us the ability to measure the effectiveness of the algorithm at different stages of the disease, and whether the algorithm is able to predict the disease at such a high level for Stage I cancer patients.

As the number of false positives increases, the effectiveness of the algorithm decreases. This is a limitation of all of the MSCAs, because none of them can predict with 100% sensitivity and specificity, so a number of the patients will be told that they have ovarian cancer when in fact they do not have it. Also, a number of patients will be told that they do not have ovarian cancer when they do have the disease. This becomes more evident when you start talking about a population of women in the tens of millions. If your sensitivity rate is as high as 95%, for every 10 million women tested, 500,000 women will result in a false positive.

Future Research

Using one Gabor filter for the entire proteomic spectra may not have produced the highest level of results for effectively predicting ovarian and control patients. The data sets could be analyzed using multiple Gabor filters across the entire spectrum. How will this be accomplished? Each of the variance/frequency values used in our research can be used as input to create all of the Gabor filters. After the filters have been created, they will be applies one at a time on each data point. This would be repeated for the entire spectrum, so that the confusion matrix values could be calculated.

In addition to the use of multiple Gabor filters on one spectra, the algorithm discussed in this paper could be applied to different types of cancer (lung, prostrate, etc) to see if it able to produce similar results.

Summary

Our coded pattern algorithm produced highly successful results for determining whether a given sample was positive for ovarian cancer. A risk of analyzing high dimensionality data is creating a model that is overfitting for a particular data set. To prevent the overfitting, we split our data set into three training and testing sets to cross-validated our algorithm. The algorithm was able to predict cancer and control with a sensitivity and specificity of above 95%, which is in contrast to the CA125 test that is only able to predict ovarian cancer at a rate of 40% to 50% This not only limited the affect of overfitting, but also allowed us to see if our algorithm could be used with different types of cancer data. As discussed in the results section, our algorithm provided very similar results in both passes of the cancer/non-cancer data set. It is our hope that we can use our algorithm with different types of cancer data to provide a mechanism that medical professionals can use to diagnose cancer in its earliest stages.

REFERENCES

Austen, B., Frears, E., Davies, H. (2000). "The Use of SELDI Proteinchip Arrays to Monitor Production of Alzheimer's Beta-Amyloid in Transfected Cells." Peptide Science **6**: 459-469.

Ball, G., Mian, S., Holding, F. et al (2002). "An Integrated Approach Utilizing Artificial Neural Networks and SELDI Mass Spectrometry for the Classification of Human Tumors and Rapid Identification of Potential Biomarkers." Bioinformatics **18**: 395-404.

Bhatti, S. (1995). "Channel Coding; Hamming Distance." Retrieved October 13, 2006, from http://www.cs.ucl.ac.uk/staff/S.Bhatti/D51-notes/node30.html.

Caldwell, R. (2006). "Arizona University Glossary." Retrieved June 7, 2006, from http://ag.arizona.edu/futures/home/glossary.html.

Check, W. A. (2002). "Catching Ovarian Cancer with Time to Spare." CAP TODAY **July**.

Daugman, J. (2004). "How Iris Recognition Works."

Fergus, K., Simonson, J. (2000b). "What is Ovarian Cancer?" Genetic Health September 5, 2000 Retrieved May 2, 2006, from http://www.genetichealth.com/BROV_What_is_Ovarian_Cancer.shtml.

Fergus, K., Simonson, J. (2000a). "Genes Can Cause Breast and Ovarian Cancer. ." Retrieved May 2, 2006, from http://www.genetichealth.com/BROV_Gen_of_BROV_Cancer.shtml.

Johnson, D. (2003). "Error Correcting Codes: Hamming Distance." 2006, from http://cnx.org/content/m10283/latest/.

Lepistö, L., Kunttu,L. Autio, J., Visa, A. (2003). Classification Method for Colored Natural Textures Using Gabor Filtering. 12th International Conference on Image Analysis and Processing, IEEE Computer Society.

Li, J., Zhang, Z., Rosenzweig, J., Wang, Y., Chan, D. (2002). "Proteomics and Bioinformatics Approaches for Identification of Serum Biomarkers to Detect Breast Cancer." Clinical Chemistry **48**(8): 1296-1304.

Lilien, R., Farid, H., Donald, B. (2003). "Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum." Journal of Computational Biology **10**(6): 925-946.

NOCC. (2006). "Understanding Ovarian Cancer." Retrieved September 25, 2006, from http://64.132.170.241/newnocc/m4prev.html.

O. Nestares, R. N., J. Portilla, and A. Tabernero. (1998). "Efficient spatial-domain implementation of a multiscale image representation based on Gabor functions. ." Journal of Electronic Imaging **7**(1): 166-173.

Paweletz, C., Gillepsie, J., Ornstein, D., Simone, N., et al (2000). "Rapid Protein Display Profiling of Cancer Progression Directly from Human Tissue Using a Protein Biochip." Development Research **49**: 34-42.

Petricoin, E. I., Ardekani, A., Hitt, B., Levine, P., Fusaro, V., Steinberg, S., Mills, G., Simone, C., Fishman, D., Kohn, E., Liotta, L. (2002). "Use of proteomic patterns in serum to identify ovarian cancer." The Lancet **359**: 572-577.

Prasad, V., Domke, J. (2005). Gabor Filter Visualization. D. o. C. Science, University of Maryland.

Qu, Y., Adam, B-L., Yasui, Y., Ward, M., Cazares, L., Schellhammer, P., Feng,Z., Semmes, O., Wright Jr, G. (2002). "Boosted Decision Tree Analysis of Surface-enhanced Laser Desorption/Ionization Mass Spectral Serum Profiles Discriminates Prostrate Cancer from NonCancer Patients." Clinical Chemistry **48**(10): 1835-1843.

Tape, T. "Introduction to ROC Curves." Retrieved October 1, 2006, from http://gim.unmc.edu/dxtests/ROC1.htm.

Unknown. (2002). "Ovarian Cancer: Significance of Early Detection." Johns Hopkins Pathology Retrieved May 2, 2006, from http://ovariancancer.jhmi.edu/earlydx.cfm.

Wagner, M., Tyler, B., Castner, D. (2002). "Interpretation of static time-of-flight secondary ion mass spectra of adsorbed protein films by multivariate pattern recognition." Analytical Chemistry **74**(1824-1835).

Wang, Z., Yip, C., Ying, Y., Wang, J., Meng, X-A., Lomas, L., Yip, T-T., Fung, E. (2004). "Mass Spectrometric Analysis of Protein Markers for Ovarian Cancer." Clinical Chemistry **50**(10): 1939-1942.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., Zhao, H. (2003). "Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data." Bioinformatics **19**(13): 1636-1643.

Yasui, Y., McLerran, D., Adam, B.L., Winget, M., Thornquist, M., Feng, Z. (2003). "An Automated Peak Identification/Calibration Procedure for High-Dimensional Protein Measures from Mass Spectrometers." Journal of Biomedicine and Biotechnology **2003**(4): 242-248.

Yasui, Y., Pepe, M., Thompson, M., Adam, B-L. et al (2003). "A Data-Analytical Strategy for Protein Biomarker discovery: Profiling of High-Dimensional Proteomic Data for Cancer Detection." Biostatistics **4**(3): 449-463.

Yates, J. R. I. (2000). "Mass Spectrometry: From Genomics to Proteomics." Trends Genet **16**(1): 5-8.

APPENDICES

Appendix A

Excel™ Extraction Script

```
%% This file will extract the intensity values from a set of Excel files

fid=fopen('control_testing.txt','r');   % Needs to be your *.name file
fout=fopen('control_test.data','w');  % Needs to be your *.data file

while 1
   new=fgetl(fid);
         if ~isstr(new),break,end
   temp=csvread(new,1,0);
   [rows,cols] = size(temp);
   for z=1:rows
           for q=1:cols
                   if ((temp(z,1) >500) && (temp(z,1) <11000))
                   fprintf(fout,'%10.6f,',temp(z,2));
                   end
           end
   end
   fprintf(fout,'\n');

   fprintf('%s\n', new);

end
fprintf('Done\n');
fclose all;
```

Appendix B

Process/Analysis Script

```
% This file will take the input cancer and control data and use a Gabor
% filter using only one phase of the filter.  The data will then be
% measured using the Hamming distance to determine the classification of
% the cancer versus the non-cancer testing data.
clear all;


%% Init the parameters for the Gabor function and coded threshold
threshold = 0.85;
s_value   = 10;
w_value   = 5;
fsize     = 3*s_value;


%%%% Create the output file
fid = fopen('confusion_matrix__.txt','w');
fid1 = fopen('count.txt','w');


fprintf('\n');
fprintf('The time and date is %s\n',datestr(now));
fprintf('\n');

%%% Add note for which pass in the confusion_matrix.txt file
fprintf(fid,'Third Data Set\n');
fprintf(fid,'\n');
fprintf(fid,'Threshold value: %4.2f\n',threshold);
fprintf(fid,'S value: %d,  W value: %d, F_Size: %d\n',s_value,w_value,fsize);


%%%%%%%%%%%%%%%%%%%%%
% Read in the data for the first pass  %%%
%%%%%%%%%%%%%%%%%%%%%
fprintf('Reading in the training control data....\n');
control_train = load ('C:\Documents and Settings\Stuart Morton\My
Documents\School\I692\Data\Second Set\control_train.data');
[contrain_row,contrain_col] = size(control_train);

fprintf('Reading in the training cancer data....\n');
cancer_train  = load ('C:\Documents and Settings\Stuart Morton\My
Documents\School\I692\Data\Second Set\ovarian_train.data');
```

```
[cantrain_row,cantrain_col] = size(cancer_train);

fprintf('Reading in the testing control data....\n');
control_test = load ('C:\Documents and Settings\Stuart Morton\My
Documents\School\I692\Data\Second Set\control_test.data');
[contest_row,contest_col] = size(control_test);


fprintf('Reading in the testing cancer data....\n');
cancer_test  = load ('C:\Documents and Settings\Stuart Morton\My
Documents\School\I692\Data\Second Set\ovarian_test.data');
[cantest_row,cantest_col] = size(cancer_test);


%%%%%%%%%%%%%%%
%% Pre-process the data:  %%
%%                        %%
%% 1) Baseline            %%
%% 2) Smooth              %%
%%                        %%
%%%%%%%%%%%%%%%
fprintf('Preprocessing the data....\n');

fprintf('Baselining the training data....\n');
control_train = prepbaseline(control_train,'min',128);
cancer_train = prepbaseline(cancer_train,'min',128);

fprintf('Smoothing the training data....\n');
control_train = prepsmooth(control_train, 128);
cancer_train = prepsmooth(cancer_train, 128);


fprintf('Baselining the testing data....\n');
control_test = prepbaseline(control_test,'min',128);
cancer_test = prepbaseline(cancer_test,'min',128);

fprintf('Smoothing the testing data....\n');
control_test = prepsmooth(control_test, 128);
cancer_test = prepsmooth(cancer_test, 128);


%%%%%%%%%%%%%%%%%%%
%% Create and apply Gabor filter %%
%%%%%%%%%%%%%%%%%%%

fprintf('Using a threshold value of %f\n',threshold);
```

```
%%%%%%%%%%% Training Data %%%%%%%%%%%

%%% Control train %%%
final_controltrain = zeros(contrain_row,(contrain_col-(2*fsize)));
size(final_controltrain);

%%% Cancer train  %%%
final_cancertrain = zeros(cantrain_row,(cantrain_col-(2*fsize)));
size(final_cancertrain);


%%% Temp matrix %%%
temp = zeros(1,contrain_col);

fprintf('Running the Gabor Filter with s=%d, w=%d, fsize=%d on train
data....\n',s_value,w_value,fsize);


for (i=1:contrain_row)
        temp = control_train(i,:);
        [Response,filter] = prepgaborfilter(temp,s_value,w_value,fsize);
        final_controltrain(i,:) = Response;
end

%%%% Determine size of final_controltrain
[controltrain_row,controltrain_col] = size(final_controltrain);


%%% Temp matrix  %%%
temp = zeros(1,cantrain_col);

for (i=1:cantrain_row)
        temp = cancer_train(i,:);
        [Response,filter] = prepgaborfilter(temp,s_value,w_value,fsize);
        final_cancertrain(i,:) = Response;
end


%%%% Determine size of final_cancertrain
[cancertrain_row,cancertrain_col] = size(final_cancertrain);


%%%%%%%% Testing Data %%%%%%%%%

%%% Control test %%%
%%% Need to size the final_controltest matrix to fit the
%%% matrix that is calculated by the prepgaborfilter,
%%% which reduces the column size by 2*fsize

final_controltest = zeros(contest_row,(contest_col-(2*fsize)));
size(final_controltest);

%%% Cancer train
%%% Need to size the final_cancertest matrix to fit the matrix
%%% that is calculated by the prepgaborfilter, which reduces the
%%% column size by 2*fsize

final_cancertest = zeros(cantest_row,(cantest_col-(2*fsize)));
size(final_cancertest);


%%% Temp matrix
temp = zeros(1,contest_col);

fprintf('Running the Gabor Filter with s=%d, w=%d, fsize=%d on test
data....\n',s_value,w_value,fsize);


for (i=1:contest_row)
        temp = control_test(i,:);
        [Response,filter] = prepgaborfilter(temp,s_value,w_value,fsize);
        final_controltest(i,:) = Response;
end

%%%% Determine size of final_controltest
[controltest_row,controltest_col] = size(final_controltest);


%%% Temp matrix
temp = zeros(1,cantest_col);

for (i=1:cantest_row)
        temp = cancer_test(i,:);
        [Response,filter] = prepgaborfilter(temp,s_value,w_value,fsize);
        final_cancertest(i,:) = Response;
end


%%%% Determine size of final_cancertest
[cancertest_row,cancertest_col] = size(final_cancertest);



%%%%%%%%%% %%%%%%%%%%%%%%%%%%
```

```
%% Determine the code for each entry in the matrix %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%% Control Training %%%%%%%%%%%

fprintf('Determine the code for the control training set... \n');

code_controltrain = zeros(controltrain_row,controltrain_col*2);


current = 1;

temp = 0;

for (i = 1:controltrain_row)
        for (j = 1:controltrain_col)

                %%% Testing for 1st entry
                temp = j;

                if (j == 1)
                        current = hardlim(real(final_controltrain(i,j)));
                        current = cat(2, current,
hardlim(imag(final_controltrain(i,j))));

                else
                        current = cat(2, current,
hardlim(real(final_controltrain(i,j))));
                        current = cat(2, current,
hardlim(imag(final_controltrain(i,j))));
                end

        end

        %%% Set the value into the code_controltrain(i,temp)
        code_controltrain(i,:) = current;
end


%%%%% Determine the pattern for the control group,
%%%%% using a X% threshold as the criteria for either
%%%%% a zero or a one.  If the total is not greater
%%%%% than/equal to the threshold, then the value is nine

control_total = zeros(1,controltrain_col*2);
```

```
for (col = 1:controltrain_col*2)

    %% init the counts
    zero_count = 0;
    one_count  = 0;

    for (row = 1: controltrain_row)
        if (code_controltrain(row,col) == 0)
            zero_count = zero_count + 1;
        else
            one_count  = one_count + 1;
        end
    end

    %fprintf(fid1,'Column:%d  zero:%d   one:%d \n',col,zero_count,
one_count);


    %%%%%%% Determine the value for the column
    if ((zero_count/controltrain_row) >= threshold)
        control_total(1,col) = 0;
    elseif ((one_count/controltrain_row) >= threshold)
        control_total(1,col) = 1;
    else
        control_total(1,col) = 9;
    end

end

%%%%%%%%%%%% Cancer Training %%%%%%%%%%%%%%%
fprintf('Determine the code for the cancer training set.... \n');

code_cancertrain = zeros(cancertrain_row,cancertrain_col*2);


current = 1;

temp = 0;

for (i = 1:cancertrain_row)
    for (j = 1:cancertrain_col)

        %%% Testing for 1st entry
        temp = j;

        if (j == 1)
            current = hardlim(real(final_cancertrain(i,j)));
            current = cat(2, current,
hardlim(imag(final_cancertrain(i,j))));

        else
            current = cat(2, current,
hardlim(real(final_cancertrain(i,j))));
            current = cat(2, current,
hardlim(imag(final_cancertrain(i,j))));
        end

    end

    %%% Set the value into the code_cancertrain(i,temp)
    code_cancertrain(i,:) = current;
end

%%%%% Determine the pattern for the control group,
%%%%% using a X% threshold as the criteria for either
%%%%% a zero or a one.  If the total is not greater
%%%%% than/equal to the threshold, then the value is nine

cancer_total = zeros(1,cancertrain_col*2);

for (col = 1:cancertrain_col*2)

    %% init the counts
    zero_count = 0;
    one_count  = 0;

    for (row = 1: cancertrain_row)
        if (code_cancertrain(row,col) == 0)
            zero_count = zero_count + 1;
        else
            one_count  = one_count + 1;
        end
    end

    %fprintf(fid1,'Column:%d  zero:%d   one:%d \n',col,zero_count,
one_count);


    %%%%%%% Determine the value for the column
    if ((zero_count/cancertrain_row) >= threshold)
        cancer_total(1,col) = 0;
    elseif ((one_count/cancertrain_row) >= threshold)
        cancer_total(1,col) = 1;
    else
```

```
                    cancer_total(1,col) = 9;
          end

end


%%%%%%%%%%%% Control Testing %%%%%%%%%%%%%%
fprintf('Determine the code for the control testing set.... \n');

code_controltest = zeros(controltest_row,controltrain_col*2);


current = 1;

temp = 0;

for (i = 1:controltest_row)
          for (j = 1:controltest_col)

                    %%% Testing for 1st entry
                    temp = j;

                    if (j == 1)
                              current = hardlim(real(final_controltest(i,j)));
                              current = cat(2, current,
hardlim(imag(final_controltest(i,j))));

                    else
                              current = cat(2, current,
hardlim(real(final_controltest(i,j))));
                              current = cat(2, current,
hardlim(imag(final_controltest(i,j))));
                    end

          end

          %%% Set the value into the code_controltest(i,temp)
          code_controltest(i,:) = current;
end

%%%%%%%%%%%% Cancer Testing %%%%%%%%%%%%%%%

fprintf('Determine the code for the cancer testing set.... \n');

code_cancertest = zeros(cancertest_row,cancertest_col*2);
```

```
current = 1;

temp = 0;

for (i = 1:cancertest_row)
        for (j = 1:cancertest_col)

                %%% Testing for 1st entry
                temp = j;

                if (j == 1)
                        current = hardlim(real(final_cancertest(i,j)));
                        current = cat(2, current,
hardlim(imag(final_cancertest(i,j))));

                else
                        current = cat(2, current, hardlim(real(final_cancertest(i,j))));
                        current = cat(2, current,
hardlim(imag(final_cancertest(i,j))));
                end

        end

        %%% Set the value into the code_cancertest(i,temp)
        code_cancertest(i,:) = current;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%% Calculate the Hamming Distance %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


fprintf('Calculating the Hamming Distance.... \n');


%%% Compare the Control training versus the Control testing
hd_control_control = zeros(controltest_row,1);

for (i = 1:controltest_row)

        %% Init the Hamming distance variables to be zero
        hd_controltest1 = 0;
        %nine_count1 = 0;

        for (j = 1:controltest_col*2)
                if (control_total(1,j) == 9)
```

```
          %                     nine_count1 = nine_count1 + 1;
                    else
                              if (control_total(1,j) ~= code_controltest(i,j))
                                        hd_controltest1 = hd_controltest1 + 1;
                              end
                    end
          end

          % Set the value to the hd_control_control
       hd_control_control(i,1) = hd_controltest1;

          fprintf(fid1,'Control testing/Control training %d:  %d\n',i,hd_controltest1);
end


%%% Compare the Control training versus the Cancer testing
hd_control_cancer = zeros(controltest_row,1);

for (i = 1:controltest_row)

          %% Init the Hamming distance variables to be zero
          hd_controltest2 = 0;
          %nine_count2 = 0;

          for (j = 1:controltest_col*2)
                    if (cancer_total(1,j) == 9)
          %                     nine_count2 = nine_count2 + 1;
                    else
                              if (cancer_total(1,j) ~= code_controltest(i,j))
                                        hd_controltest2 = hd_controltest2 + 1;
                              end
                    end
          end


          % Set the value to the hd_control_cancer
       hd_control_cancer(i,1) = hd_controltest2;

          fprintf(fid1,'Control testing/Cancer training %d:  %d\n',i,hd_controltest2);
end


%%% Compare the Cancer training versus the Control testing
hd_cancer_control = zeros(cancertest_row,1);

for (i = 1:cancertest_row)
```

```
    %% Init the Hamming distance variables to be zero
    hd_cancertest1 = 0;
    %nine_count3 = 0;

    for (j = 1:cancertest_col*2)
        if (control_total(1,j) == 9)
    %           nine_count3 = nine_count3 + 1;
        else
            if (control_total(1,j) ~= code_cancertest(i,j))
                hd_cancertest1 = hd_cancertest1 + 1;
            end
        end
    end

    % Set the value to the hd_cancer_control
   hd_cancer_control(i,1) = hd_cancertest1;

    fprintf(fid1,'Cancer testing/Control training %d:  %d\n', i,hd_cancertest1);
end


%%% Compare the Cancer training versus the Cancer testing  %%%

hd_cancer_cancer = zeros(cancertest_row,1);

for (i = 1:cancertest_row)

    %% Init the Hamming distance variables to be zero
    hd_cancertest2 = 0;
    %nine_count4 = 0;

    for (j = 1:cancertest_col*2)
        if (cancer_total(1,j) == 9)
    %           nine_count4 = nine_count4 + 1;
        else
            if (cancer_total(1,j) ~= code_cancertest(i,j))
                hd_cancertest2 = hd_cancertest2 + 1;
            end
        end
    end

    % Set the value to the hd_cancer_control
   hd_cancer_cancer(i,1) = hd_cancertest2;

    fprintf(fid1,'Cancer testing/Cancer training %d:  %d\n', i,hd_cancertest2);
end
```

```
tn = 0;
fn = 0;
tp = 0;
fp = 0;

fprintf('Calculate the confusion matrix.... \n');

for (k=1:controltest_row)
        if (hd_control_control(k,1) < hd_control_cancer(k,1))
                tn = tn + 1;
        else
                fn = fn + 1;
        end
end

for (k=1:cancertest_row)
        if (hd_cancer_cancer(k,1) < hd_cancer_control(k,1))
                tp = tp + 1;
        else
                fp = fp + 1;
        end
end

%% Print the confusion matrix
fprintf(fid,'tn: %d\n',tn);
fprintf(fid,'fn: %d\n',fn);
fprintf(fid,'tp: %d\n',tp);
fprintf(fid,'fp: %d\n',fp);

fprintf(fid,'\n');
fprintf(fid,'Sensitivity: %4.2f\n',tp/(tp+fn));
fprintf(fid,'Specificity: %4.2f\n',tn/(tn+fp));
fprintf(fid,'Accuracy: %4.2f\n',(tn+tp)/(tp+tn+fp+fn));

fprintf('\n');
fprintf('\n');
fprintf('Sensitivity: %4.2f\n',tp/(tp+fn));
fprintf('Specificity: %4.2f\n',tn/(tn+fp));
fprintf('Accuracy: %4.2f\n',(tn+tp)/(tp+tn+fp+fn));

fprintf('The time and date is %s\n',datestr(now));

fprintf('Done.... \n');

%%%% Close the file
fclose(fid);
fclose(fid1);
```

VITA

Stuart M. Morton

4366 Greenthread Drive
Zionsville, IN 46077
Tel: (317) 769-2500
Email: smorton22@tds.net

WORK EXPERIENCE:

Covance Inc, Indianapolis, IN. 2005 – 2006

Project Details:

Java Developer: **August 2005 – August 2006**

Developed applications for the Covance Central Labs that allows customers to create, maintain and retrieve results of clinical trials.

As a **Senior Java Developer**, my responsibilities included:

- Creating web-based and GUI applications using Java J2EE, C# and JTML
- Providing technical support for the Covance clinical trial sites
- Developing software to allow Covance the ability to data mine clinical trial information

As a **Database Administrator**:

- Maintained the clinical trial database
- Provided data support for Covance clinical trial kits

Software: **J2EE, JHTML, C#, Microsoft SourceSafe, JSP, HTML, Perl**
Operating System: **UNIX, Windows 2000 and XP**

Lucent Technologies, Naperville, IL. 1997 – 2005

Project Details:

Packet Core DB Developer: **Jan 2001 – July 2005**

The Packet Core project allows a customer the ability to utilize secure Internet connections to route cellular data calls rather than using local carriers, and thus reducing their costs.

As a **Senior Java Developer/Architect**, my responsibilities included:

- Creating Java based software, Dynamic Update Agent that allows dynamic data to be sent to customer's equipment without the need to reboot the equipment in order to read the new DB values.
- Providing updates to the DUA when the customer requests new dynamic data to be added to their system
- Developing thread monitoring for Java processes that has a patent pending
- Unit testing of the Dynamic Update Agent
- Creating Java based DB backup and restore software that allows a customer to backup and restore their configuration data for disaster recovery.
- Responsible for the software architecture of the DB system for the Packet Core project

- Developing and maintaining an SQL based DB that allows a customer to provision their Packet Core equipment via a web based provisioning interface
- Unit testing of the DB
- Providing scripts to populate the DB for lab testing by other development teams
- Training classes for developers in China who will take ownership of the Packet Core DB system
- DB lab support for other development teams

Software: **C/C++, JNI, J2SE, JDBC, CCMS, Apache Web Server, JSP, HTML, JavaScript, Perl**
Operating System: **UNIX, Windows 2000**

ECP TDMA Edge DB Developer: Feb 2000 – Dec 2000

The TDMA cellular system utilizes a time division algorithm to allow multiple cell phone users to use the same radio channel, but in different time allocations. EDGE is an enhanced data rate for cellular phones that use the Global System for Mobile Communications (GSM)

As a SQL/**XML Developer**, my responsibilities included:

- Developing a prototype database system that utilized XML to generate SQL that would be used by the customer to provision their TDMA system
- Creating scripts to simulate DB creation, DB field population and failure scenarios
- Testing the prototype DBMS in a cellular lab environment
- Writing customer documentation

Software**: SQL, XML, XML Schema, DTD, HTML**
Operating System**: UNIX, Windows 2000**

ECP CDMA DB Designer: Jul 1997 - Jan 2000

The ECP is the Executive Cellular Processor that is responsible for processing cellular telephone calls, billing and performance monitoring of a cellular customer's wireless telephone equipment. The provisioning system used a text based menu system that stored the data in a link list database. CDMA is a cellular technology that utilizes a spread spectrum form of modulation requiring a contiguous block of spectrum (1.25 MHz) rather than channels as used by analog of TDMA telephones.

As a team leader of five **C** database developers, my responsibilities included:

- Taking the data requirements from the customer and creating a design document , and then assigning the developer's their piece of the development
- Implementing the data design to provide the customer a text based interface that allows them to provision their cellular equipment and have it stored in a database that is accessed using C structures.
- Testing of the code in a cellular lab environment
- Providing support of the provisioning system at the customer's site
- Generating weekly status reports on the status of the DB development
- Maintaining the group and department web pages
- Leading a Quality Improvement team to develop a standardized Data System Requirements document template. The plan was to reduce the interval and provide consistent data requirements across projects
- Co-wrote the online Database Process, which involved working with multiple sources including the Design and Code Process Management team, the Data Subsystem owners and the Data developers.

Software**: C, HTML, Perl, Java, JavaScript, Word, Excel, PowerPoint, Change Management System**
Operating System **: UNIX, Windows NT, 2000**

- Current role involves managing a team of 7 developers on Lucent's Internet Protocol project, while also developing, testing and integrating Java code for the same project that will be used to dynamically update data on customer's telephone equipment.
- Created and presented a remote testing class for Lucent employees that demonstrates the ability to test and develop software from any remote location.

**Purdue University Department of Computer Science at Indianapolis, IN.
1994 - 1997**

- Research Assistant – Research involved transaction management in distributed medical database systems. *
- Teaching Assistant – Instructor for two Internet development classes and a Pascal programming class.

Eli Lilly and Company, Research Laboratories, Indianapolis, IN. Summer 1990

- Worked as an analyst in the Department of Chemistry.

EDUCATION:

PhD Bioinformatics
Started program August 2006
Indiana University
Indianapolis, IN 46202

M. S. Bioinformatics
Projected Graduation September 2006
Indiana University
Indianapolis, IN 46202
Grade Point Average: 4.0/4.0

Masters Thesis, Indiana University (Jan 2005-Sept 2006):

- **“**Gabor Wavelet Phase Quantization and Binary Coding for Ovarian Cancer Serum Protein Profiling” **(Submitted to SAC2007)**
    - Analyzed proteomic ovarian spectra generated by Mass Spectrometry instruments
    - Utilized Gabor filters for detecting optimal localizations in the spectra
    - Proposed a binary coding scheme to classify ovarian and non-ovarian samples

M. S. Computer Science, August 1997
Purdue University
Indianapolis, IN 46202
Grade Point Average: 3.6/4.0

PUBLICATIONS (Available at http://home.indy.rr.com/smorton/abstr.html)

- S. Morton, O. Bukhres, E. Vanderdijs, P. Zhang, M. Mossman, C. Crawley, J. Platt. “A Proposed Mobile Architecture for a Distributed Database Environment” *Proceedings of the 5th Euromicro Workshop on Parallel and Distributed Processing*, 1997.

- S. Morton, O. Bukhres. “Utilizing Mobile Computing in the Wishard Memorial Hospital Ambulatory Service” *Proceedings of the 12th ACM Symposium on Applied Computing (ACM SAC 1997*). S. Morton, O. Bukhres. “Mobile Transaction Management in Distributed Medical Databases” *Proceedings of the 10th IEEE Symposium on Computer-Based Medical Systems (CBMS 1997*).
- S. Morton, O. Bukhres, M. Mossman. “Mobile Computing Architecture for a Battlefield Environment” *Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications,* 1996.
- S. Morton, O. Bukhres. “Mobile Transaction Recovery in Distributed Medical Databases” *Proceedings of the IASTED Eighth International Conference on Parallel and Distributed Computing and Systems,* 1996.

ADDITIONAL UNIVERSITY CLASSES:
Indiana University at Indianapolis (1994):

- **Graduate**: Immunology
- **Undergraduate**: Introduction to Microbiology, Cell Biology, Computer Applications in Biology and Medicine, Genetics & Molecular Biology, Introduction Java Programming

B. A. Computer Science, May 1993
Pre-Med program
DePauw University
Greencastle, IN 46135
Grade Point Average: 3.25/4.0

RELEVANT COMPUTER SKILLS (1993- 2006):

- **Programming Languages:** C, C++, Java, C#, JNI, HTML, UML, XML, SQL, Perl
- **Operating Systems:** UNIX, Windows, Mac
- **Other Applications**: MS Office, Netscape, Internet Explorer, Latex, MATLAB