Wright State University

CORE Scholar

Browse all Theses and Dissertations

Theses and Dissertations

2007

An Evolutionary Programming Algorithm for Automatic Chromatogram Alignment

Bonnie Jo Schwartz
Wright State University

Repository Citation

AN EVOLUTIONARY PROGRAMMING ALGORITHM FOR AUTOMATIC CHROMATOGRAM ALIGNMENT

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering

By

BONNIE JO SCHWARTZ
B.S., Coastal Carolina University, 2001

2007
Wright State University

WRIGHT STATE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

April 3, 2007

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Bonnie Jo Schwartz ENTITLED An Evolutionary Programming Algorithm for Automatic Chromatogram Alignment BE ACCEPTED IN PARTIAL FULFLILMENT OF THE REQUIREMENT FOR THE DEGREE OF Master of Science in Computer Engineering.

Mateen Rizki, Ph.D.
Thesis Director

Forouzan Golshani, Ph.D.
Department Chair

Committee on
Final Examination

Thomas Hartrum, Ph.D.

Thomas Sudkamp, Ph.D.

Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies

ABSTRACT

Schwartz, Bonnie Jo. M.S.C.E, Department of Computer Science and Engineering, Wright State University, 2007. An Evolutionary Programming Algorithm for Automatic Chromatogram Alignment

Scientists use liquid chromatography/mass spectrometry (LC/MS) instruments to measure animals' metabolic responses to drugs, their environment, or diseases. These instruments produce large quantities of data that needs to be analyzed. The data, however, can be distorted due to changes in the testing environment and noise produced by the instrument. Automating the removal of these distortions is crucial in processing the data. The purpose of this thesis is to develop an algorithm that will automate the process.

The data produced by the LC/MS instrument were treated as images and image registration techniques were applied. A polynomial transformation function between chromatograms was assumed. An evolutionary programming algorithm was used to determine the coefficients of the polynomial. Based on observations of the data set, the data was manipulated in different ways to determine the best technique for registering. This thesis describes the data manipulation, details of the resolution of the algorithm and provides some experimental results.

The results show that the evolutionary programming algorithm is a reasonable solution for automating the registration of chromatograms produced by a liquid chromatography/mass spectrometry instrument. Very similar chromatograms were easy to register using the evolutionary algorithm while chromatograms with fewer similarities were more difficult to register. These results show that more work needs to be performed to fine-tune the algorithm to work on chromatograms that are highly distorted.

TABLE OF CONTENTS

LIST OF FIGURES

1. PURPOSE AND BACKGROUND

1.1 Liquid Chromatography/Mass Spectrometry

Liquid chromatography/ mass spectrometry is a powerful analytical tool that combines the capabilities of both liquid chromatography and mass spectrometry. “Liquid chromatography is a physical separation method in which the components to be separated are selectively distributed between two immiscible phases: a mobile phase is flowing through a stationary phase bed.” [1] More simply, liquid chromatography is the process of separating ions or molecules that are dissolved in a solvent. When the sample solution is in contact with another solid or liquid, differing degrees of interaction will occur due to differences in adsorption, ion-exchange, partitioning, or size. These differences are used to separate the mixture components, allowing the transit time of the solutes through a column to be determined. [7]

“Mass spectrometry is the production of ions that are subsequently separated or filtered according to their mass-to-charge ratio and detected.”[1] In simpler terms, mass spectrometry is the art of measuring atoms and molecules to determine their molecular weight. The information obtained from mass spectrometry is useful in identifying species. To perform this analysis, a charge is put on the molecules of interest, i.e., the analyte, and then the trajectory response of the resulting ions is measured. [7] Scientists can discover differences in samples by analyzing the data produced from these methods. These differences can provide important information about changes in the sample subjects’ body chemistry. [1]

The LC/MS instrument produces large quantities of data. Due to changes in environment and noise in the instrument, distortions occur in the data. In order for scientists to analyze this data, the distortions must be removed. This thesis revolves around automating the process of removing distortions so that large amounts of data can be processed quickly.

1.2 Metabonomics

Metabonomics is defined as “the quantitative measurement of the time-related multiparametric metabolic response of living systems to pathophysiological stimuli or genetic modification.”[9] In more general terms, metabonomics is the study of metabolic responses to drugs, environmental changes and diseases. Metabonomics identifies and quantifies molecules that are affected as the direct result of a disease, toxic insult, genetic modification, or other external stimulus. Knowledge about these compounds can potentially be used for “diagnosis, safety assessment screening, or to direct further research.” Metabonomics is used to identify up- or down-regulated metabolites and biomarkers as a result of a disease state, toxicity, genetic modification, and environmental factors.[10] The experiment described in this paper analyzes complex metabonomic data that is produced using a liquid chromatography/mass spectrometry instrument.

Due to its high sensitivity and resolution, LC/MS is the preferred analysis for bio-fluid samples such as serum, plasma, and urine. The LC/MS analysis produces three-dimensional information regarding the metabolites: retention characteristics, mass-charge-ratio (*m*/*z*), and peak intensities. The additional *m*/*z* information from LC/MS analysis is ideal for metabonomics. At the same time, more difficulties in alignment and deconvolution are encountered with the additional *m*/*z* information. [11] Errors and noise can be introduced in to the data sets produced by LC/MS analysis via instrument inconsistencies, human error and biological differences between samples. These errors pose the problem of how to extract the useful information from the raw data. This has become an obstacle for LC/MS in metabonomics applications. [11] Data in large databases can be compared if each sample is characterized by the same number of variables, each of those variables is represented across all observations and a variable in one sample has the same biological meaning in all other samples.[11] When this is the case, data from multiple samples can be registered to align corresponding points that are misaligned due to errors or noise.

1.3 Image Registration

Image registration is the process of spatially aligning two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. By overlaying the images, the centers of corresponding pixels can be matched. Differences between images (distortions) may be introduced due to different imaging conditions. [15][16] Images can be framed in many ways. Some of the most common types of transformation include rigid, affine, perspective and global polynomial. Rigid transformations account for object or sensor movement in which the objects maintain their relative shape and size. Affine transformations occur when the same object is shown from different angles. Perspective transformations occur when the same object is shown from different distances. Polynomial transformations take into account many types of distortions as long as they do not vary too much over the image. [4]

When the data collected from LC/MS analysis is treated like images, they can be registered using image registration techniques. The following figure shows an example of data represented as an image. With retention time on the y-axis and mass on the x-axis, the different shades of the points represent different peak intensities.

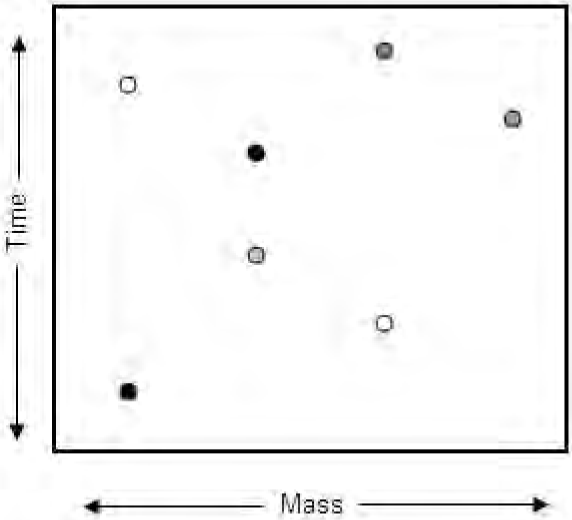


Figure 1: Data as an image

A common problem arises with images taken at different times or from different view points. If these images need to be compared to detect differences, they must first be aligned. To accomplish this alignment, a transformation must first be found so that the points in one image can be matched with their corresponding points in the other image. A transformation is a mapping of locations of points in one image to a new location in another. [4] Figure 2 demonstrates how this data might be misaligned. Although both images (data sets) are very similar, there are slight differences that need to be discovered and adjusted. A transformation would be used to match up corresponding points.

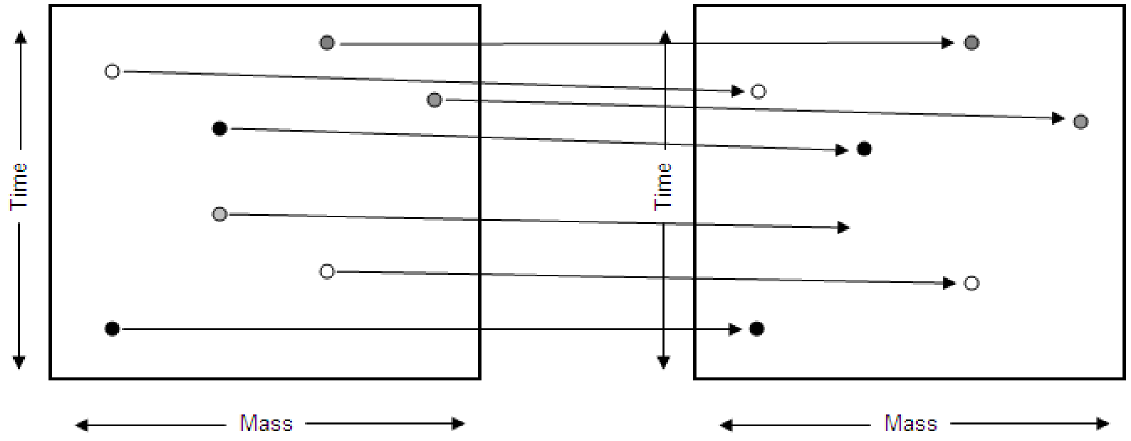


Figure 2: Slight misalignment

Transformations used to align images can be global or local. A global transformation is a single equation which maps the entire image. Local transformations map the image differently depending on the spatial location. [4] Both types of transformations were explored in this experiment. While the global transformation worked for many cases, the local transformations added extra precision to the alignment.

In images and also in the data from the LC/MS instrument, there are two different types of variations. One type is distortions. These are the variations that should be removed. They result from noise, shifting or skewing of data from inconsistencies in the instrument, measurement error or environmental effects. Other variations should not be removed because they represent natural variation of the underlying biological system. These are the ones that need to be detected. [4] The algorithm described in this paper checks each data point for accuracy using an evolutionary algorithm and it is assumed that the variations of interest will stand out while distortions will be removed.

The most general type of transformation is the polynomial transformation. [4] This works well for many types of distortions. A polynomial transformation is used in the algorithm described in chapter two.

1.4 Evolutionary Computation

Most registration techniques involve searching over the space of potential transformations to find the optimal transformation for a particular problem. [4] Evolutionary computation was used in this algorithm to search for the coefficients of the polynomial transformation function. Evolutionary algorithms simulate evolution to search for solutions to complex problems. [2] The common underlying idea behind all evolutionary computing techniques is the same: given a population of individuals, environmental pressure causes natural selection (survival of the fittest) and the overall fitness of the population grows. This process is easily viewed as optimization. [12]

Given a function to be maximized, a set of candidate solutions can be randomly created and the function can be used as an abstract fitness measure. This function is referred to as the fitness function. Using the fitness function, some of the better candidates are chosen to seed the next generation by applying recombination and/or mutation to some or all members of the population. Recombination can be achieved in many ways but in general, it is the combination of two or more existing parental solutions to produce one or more new candidate solutions, the children. Mutation is applied to one candidate and results in one new, slightly modified candidate solution. Applying recombination and mutation leads to an entire set of new candidates, the offspring. The offspring then compete with the previous generation for a place in the next generation. The winners are determined by their fitness. This process can be iterated until a solution is found or a previously set time limit is reached. [2][3] Figure 3 shows the general scheme of an evolutionary algorithm.

```
Initialize population with random individuals
Compute fitness of all individuals
WHILE stopping criteria not met
     Select parents
     Create offspring via recombination and mutation
     Compute fitness of offspring
     Replace some parents by some offspring
```

Figure 3: Basic evolutionary algorithm

"According to Darwin, the emergence of new species, adapted to their environment, is a consequence of the interaction between the survival of the fittest mechanism and undirected variations." Therefore, recombination and mutation must be stochastic. The pieces of each parent to be exchanged during recombination as well as the mutations are random. However selection of parents and new generation can be stochastic or deterministic. Stochastic selection gives even the weak individuals a chance to survive or become a parent while deterministic selection only keeps a pre-selected group of individuals— usually those with the best fitness. [12]

Evolutionary computation algorithms are considered generate-and-test, also known as trial-and-error, algorithms. The fitness function represents an estimation of solution quality. The search process is driven by recombination and mutation creating new candidate solutions. The selection operators are also key to the search process. However, evolutionary algorithms different from other generate-and-test algorithms because they are population based, i.e., they process a whole set of candidate solutions and create new solutions by using recombination to mix information from previous solutions. [12]

1.5 Evolutionary Programming (EP)

Evolutionary programming is a variation of evolutionary computing that is frequently used in optimization problems. EP does not rely on any form of recombination- only mutation. The typical selection method used in EP is to mutate each of the N members of the population to create N new offspring. The next generation typically contains the best N individuals of the 2N parents and offspring. [6] Evolutionary programming traditionally uses representations that are tailored to the problem domain. The type of mutation used is depends on the representation used and can also be adaptive, changing with each generation. [2] Figure 4 shows the basic evolutionary programming algorithm. The specific EP algorithm used for this research will be detailed in section 2.2.

```
Initialize population with random individuals
Compute fitness of all individuals
WHILE stopping criteria not met
     Create one offspring from each individual via mutation
     Compute fitness of offspring
     Replace some parents by some offspring
```

Figure 4: Basic evolutionary programming algorithm

The rest of this paper is laid out in the following format. Chapter two discusses the data that was used for this research. It also describes how the data was manipulated and the evolutionary search algorithm used. Chapter three provides details about the four experiments performed for this research. It details setup as well as results for each experiment. Chapter four discusses conclusions drawn from the results obtained in the four experiments described in chapter three. Chapter four also discusses future work.

2. DATA AND ALGORITHM DESCRIPTION

2.1 Data Description

The raw data provided from the liquid chromatography-mass spectrometry instrument can be processed into a matrix of mass values along with a corresponding matrix of peak intensities and array of retention times. Figure 5 shows these three elements plotted together with time on the y-axis, mass on the x-axis and the shade of each point representing the peak intensity. The arrow in figure 5 represents a scan line. The data in a scan line consists of the peak intensity of each mass at a single point in time. There are approximately 100 scan lines per minute in the data provided for this experiment.

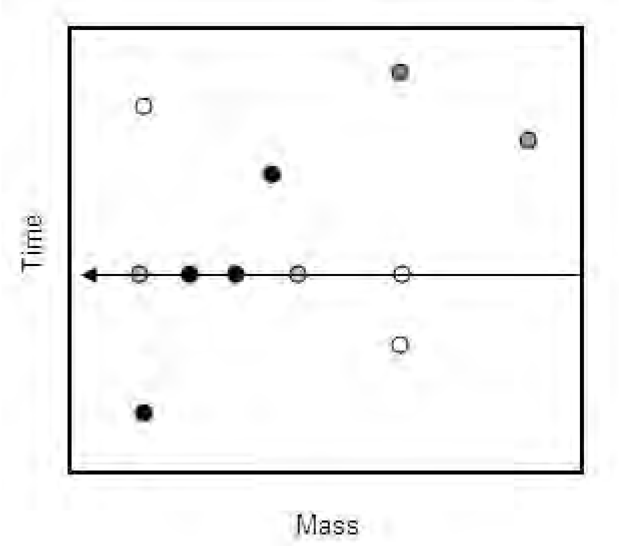


Figure 5: Plotted data; scan line represented by arrow

For this experiment, it was assumed that there is little to no distortion in mass so no attempts were made to adjust the mass values. The intensity matrix is summed to the time axis to create an array of summed intensities. Each element of the intensity array corresponds to the same element of the retention time array. The following figure illustrates this process of data organization.

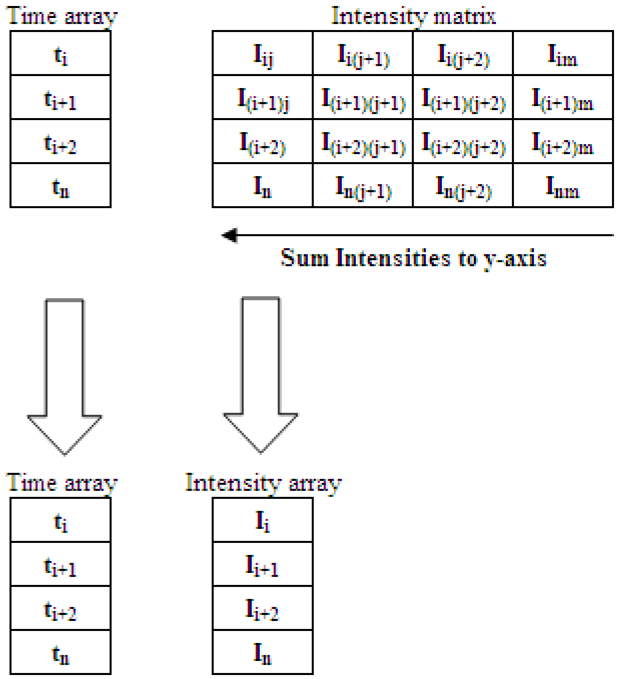


Figure 6: Data organization for analysis

Plotting the intensity array against the time array produces a chromatogram. A chromatogram is a graph relating concentration (intensity) of solute leaving a chromatographic column, against time, and takes the form of a series of peaks. [5] To remove variations in the chromatograms, the data was normalized before analysis. Before summing the intensity matrix, all intensities were scaled by the maximum intensity value. This produced an intensity scale of zero to one for each chromatogram. As mentioned in section 1.2, it is also possible to normalize using a variable that has the same biological meaning throughout every sample. For example, creatinine is a chemical waste molecule that is generated from muscle metabolism. [14] Creatinine has a known mass value (114.1271 m/z) and a known retention time of 1.0 – 1.5 minutes. While normalization using the maximum intensity worked well for this experiment, using creatinine to normalize was another option.

The data used for this research came from 10 minutes of tests. There were 1000 scan lines of data—600 (6 minutes) of testing and 400 (4 minutes) of flushing the tube. The four minutes of flushing was ignored so each time array contained 600 elements, approximately 100 elements (scan lines) per minute. The following figure shows an example of a chromatogram produced using the methods detailed in figure 6.

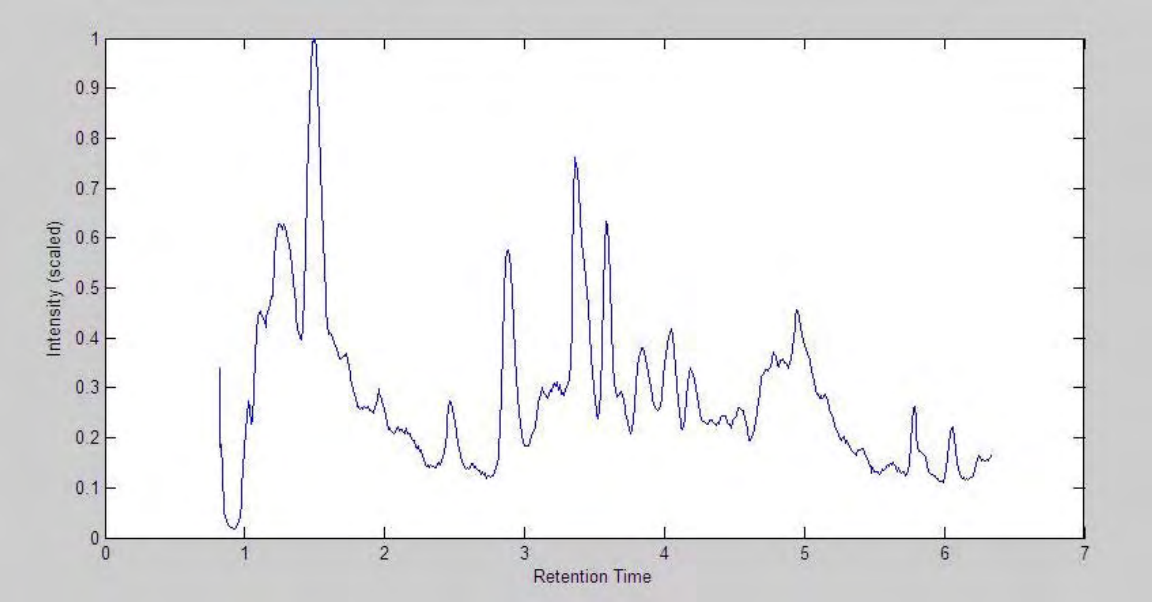


Figure 7: Sample chromatogram

It was suspected that once a transformation function was found for the chromatograms, that one polynomial could be used to adjust the entire data set—a global transformation. The goal of this experiment was to find a polynomial transformation function that can be used to register two chromatograms. Once registered, differences in data could be discovered. This process is performed on a pair wise basis. Figure 8 shows how a large set of samples would be processed. One chromatogram is chosen as the base case. All other samples are then registered individually to that base case. One transformation function is found for each sample that will register it to the base case.

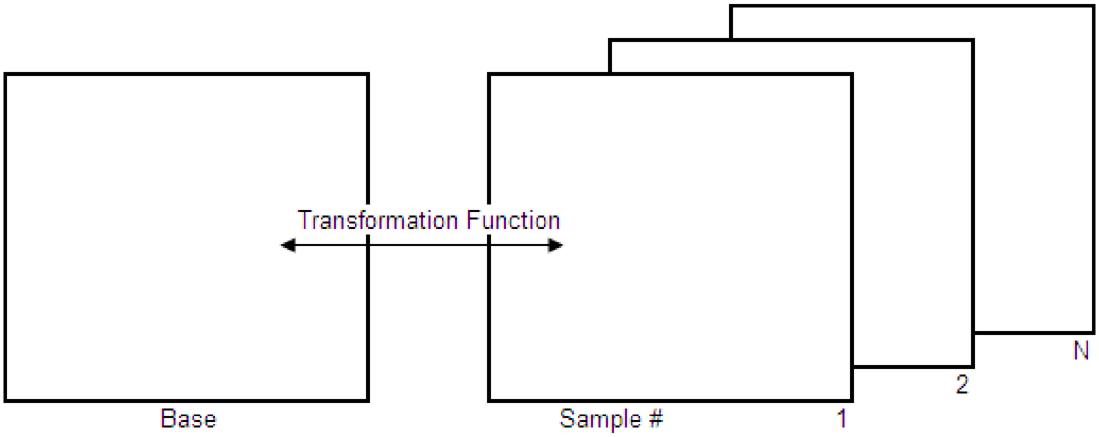


Figure 8: Registering a large set of samples

The data set provided contained approximately 410 samples from varying animals at varying times with varying dosages. Dosages ranged from 0 to 100 and data were collected from each animal at 0, 24 and 48 hours. The data were assumed to only be distorted along the time access. By using only the time variable, calculations and analysis were simpler than if shifts in both time and mass were considered.

Samples that were taken from the same animal were visually close to being aligned. The following figure shows two samples taken from the same animal with a dosage of zero. Since these two chromatograms should be the same, the distortion seen (mostly after 3 minutes) is probably caused by noise from the instrument or measurement errors. It is these slight distortions that need to be compensated for during the registration process.

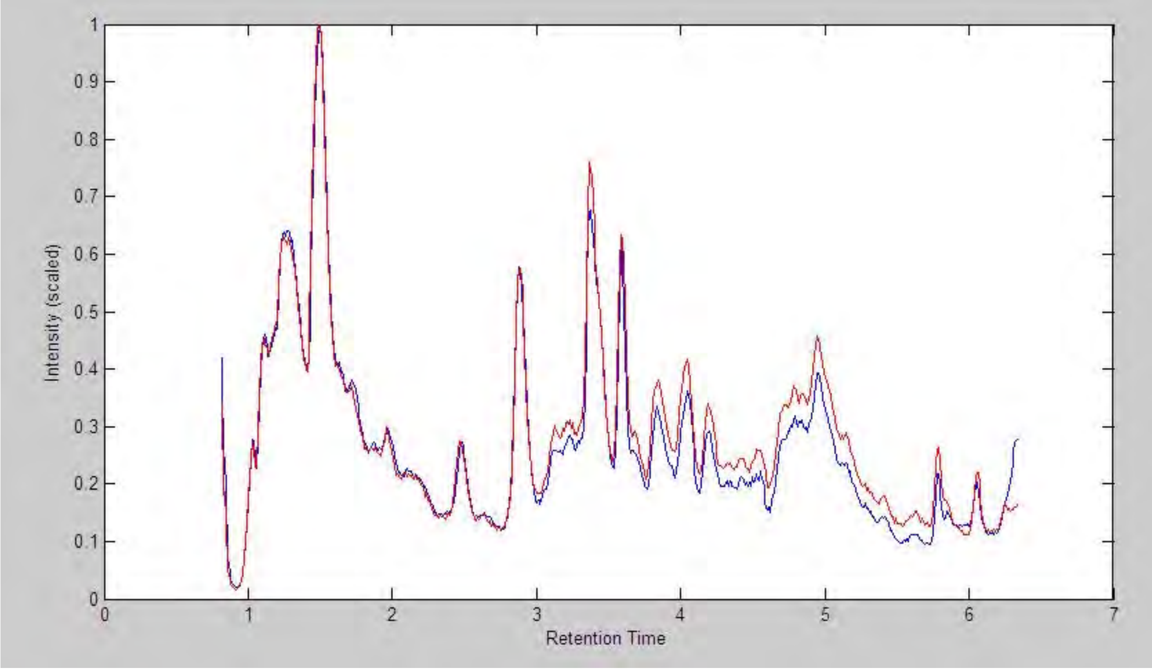


Figure 9: Two chromatograms taken from same animal

Samples taken from different animals, regardless of the day or dosage, could be similar or very different. Figure 10 demonstrates how different those chromatograms could be. Since these chromatograms come from different animals, some biological variation is assumed as well as distortion from the instrument or measurement errors.

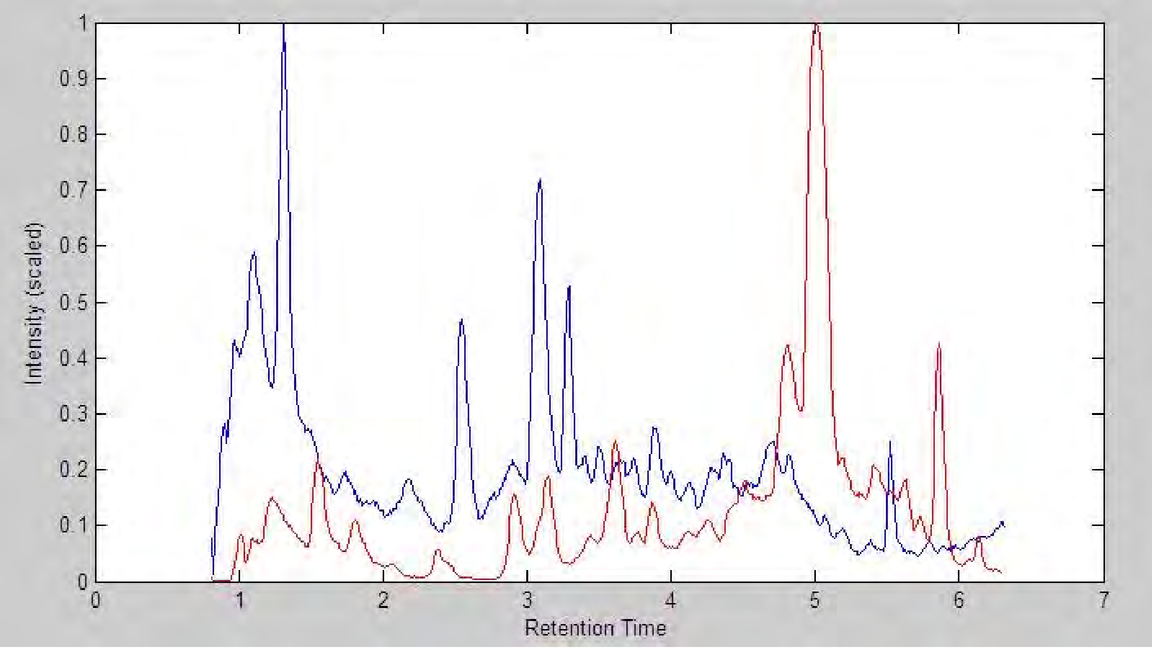


Figure 10: Two chromatograms taken from different animals

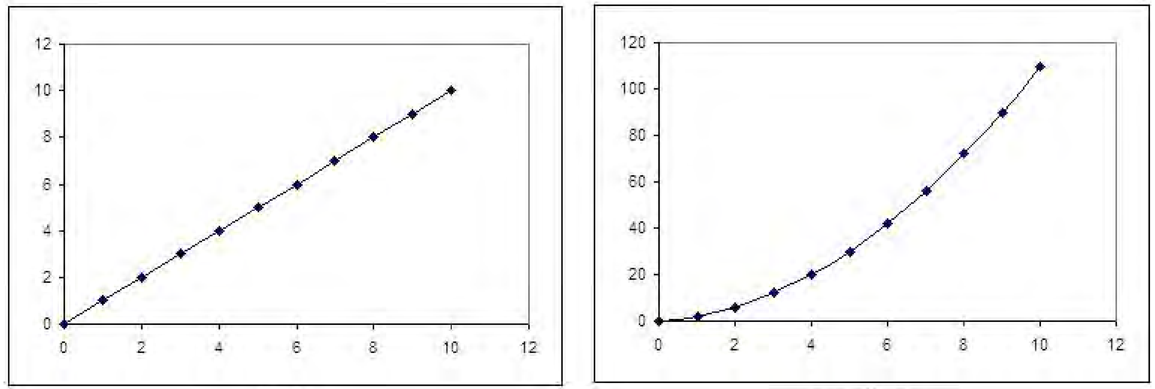
Section 3.2 will discuss registration of chromatograms that initially only contain small distortions while section 3.3 will discuss issues that arise while attempting to match data with many dissimilarities.

2.2 Algorithm Description

The hypothesis of this experiment is that there exists an n-degree polynomial,

$$c_1 * t^n + c_2 * t^{n-1} + \ldots + c_n * t + c_{n+1},$$

where the coefficients c_1 through c_{n+1} are unknown. This polynomial represents the transformation along the time axis between two chromatograms. An evolutionary algorithm was used to find the optimal combination of values for those coefficients. Each individual used in the search consisted of a set of coefficients. The figure 11 shows samples of how this transformation works. For example, in the case where the chromatograms are a perfect match, the base time and the sample time are equal. This linear relationship is displayed in the left plot in figure 11. In the case of a shift only, the sample time is some number (the coefficient) multiplied by the base time ($t_s = c * t_b$). The second plot in figure 11 shows the relationship between base time and sample time when a second degree polynomial transformation is used. For this experiment, second or greater degree polynomial transformations are used because as time increases, the chromatograms become more misaligned. Section 3.4 will discuss these changes in chromatograms over time in more detail.



$t_s = t_b$

$t_s = t_b^2 + t_b$

Figure 11: Base time and sample time relationships

The algorithm used was a basic evolutionary programming algorithm. Each individual consisted of a set of coefficients that would potentially define an n-degree polynomial transformation function. Figure 12 shows an example of a single individual where c_1 through c_{n+1} represent the coefficients in the transformation function.

c_1	c_2	…	c_{n+1}

Figure 12: One individual

As figure 9 demonstrated, it was visually obvious that some sample chromatograms were initially very similar to the base chromatogram with which they were to be registered. Using this domain knowledge, it was determined that the population should be initialized using a linear transformation of the base chromatogram as a starting point for each candidate solution. The initial population was generated by adding a normally distributed random variable with a mean of zero and standard deviation of one to all coefficients except the first degree coefficient where 1 plus a normally distributed random variable was the starting value. Figure 13 illustrates how the population was initialized.

$c_1 = 0$	$c_2 = 1$	$c_3 = 0$

$c_1 = 0 + N(0, 1)$	$c_2 = 1 + N(0, 1)$	$c_3 = 0 + N(0, 1)$

Figure 13:
Top: Linear solution
Bottom: Initialization of candidate solutions

The individuals used in this experiment consisted of three to six real values and no recombination of chromosomes was used. Mutation was the only method of variation in the population. To create a new child solution from a parent solution via mutation, a scaled, normally distributed random number with a mean of zero and a standard deviation of one was added to each coefficient in the parent solution.

$$c_c = c_p + \alpha * N(0, 1)$$

The scaling constant, α, determined the size of the mutation. For example, if the random number generated was 1.2 and α was 0.01, the child coefficient would only be increased by 0.012. By using this scaling factor, the degree of change from generation to generation could be controlled. Figure 14 shows how a child candidate solution was produced from its parent solution.

$c_{1(p)}$	$c_{2(p)}$	$c_{3(p)}$

$c_1 = c_{i(p)} + \alpha * N(0, 1)$	$c_2 = c_{2(p)} + \alpha * N(0, 1)$	$c_3 = c_{3(p)} + \alpha * N(0, 1)$

Figure 14:
Top: Parent candidate before mutation
Bottom: Child candidate after mutation

Each parent produced exactly one offspring and only the best N individuals were kept around for the next generation, where N is the size of the initial population. Elite selection was used because if one parent candidate solution is a high quality solution, a slight mutation to that individual's coefficients could be an even better solution. However, if a parent candidate solution was of poor quality, the slight mutation used in this algorithm would not be likely to create a significantly better quality solution. Population size and number of generations varied throughout the experiment to test for the optimal situation. Those variables will be discussed in chapter 3 in relation to each individual experiment. The algorithm used in this experiment is outlined in figure 15.

```
randomly generate an array of N candidate solutions
evaluate fitness of each solution
sort population_array by fitness
for (number of generations)
{
   for (size of initial population)
   {
     mutate population_array[i] to create new solution
     evaluate fitness of new solution
     add new solution to back of population_array
   }
   sort population_array (now size = 2N)
   crop population_array back down to size N, keeping only best
}
```

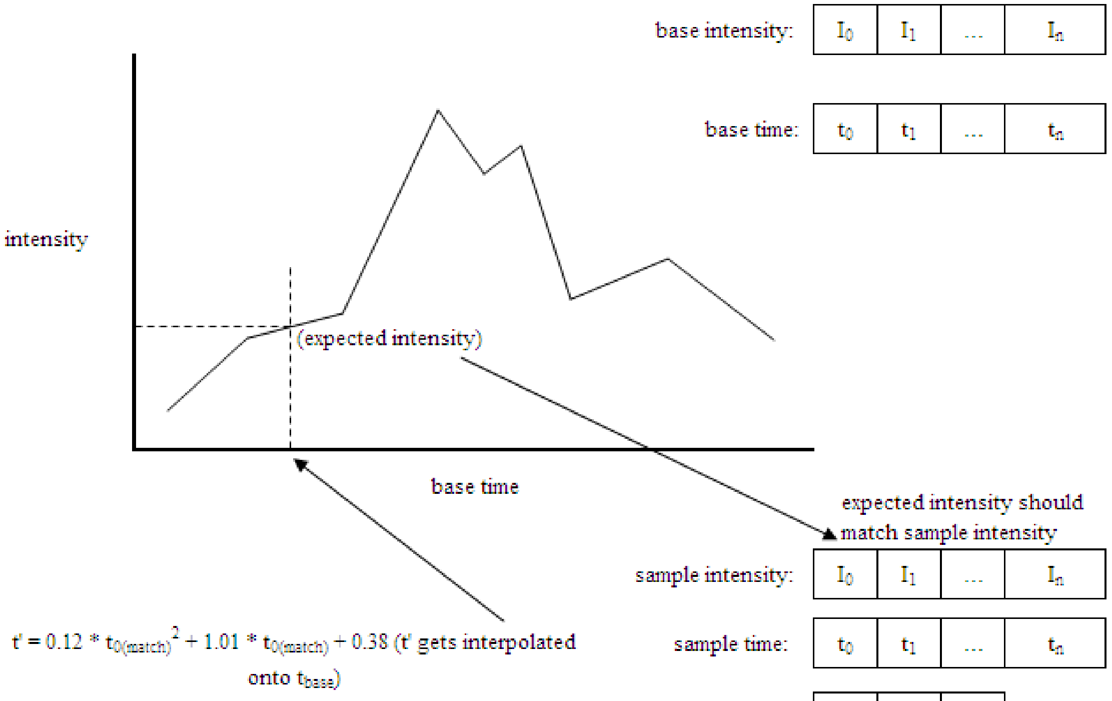
Figure 15: EP algorithm used

Figure 16 shows the algorithm used to evaluate the fitness of a candidate solution in this experiment.

```
for each element of the sample time array
{
     evaluate with coefficients to get a new time
     interpolate new time onto base chromatogram
     find the expected intensity at the new time point
     sum = sum + (expected intensity – sample intensity)$^2$
}
fitness = sum / total num of elements
fitness = sqrt(fitness)
```

Figure 16: Fitness algorithm

The fitness algorithm produced the mean squared error of each set of coefficients when evaluated along the entire sample time array. The fitness, therefore, was maximized at zero because a mean squared error of zero indicates a perfect match between the sample chromatogram data and the base chromatogram data. Figure 17 illustrates the fitness algorithm.


base intensity:
I_0 | I_1 | ... | I_n
base time:
t_0 | t_1 | ... | t_n
intensity
(expected intensity)
base time
expected intensity should match sample intensity
sample intensity:
I_0 | I_1 | ... | I_n
sample time:
t_0 | t_1 | ... | t_n
chromosome:
0.12 | 1.01 | 0.38
$t' = 0.12 * t_{0(match)}^2 + 1.01 * t_{0(match)} + 0.38$ (t' gets interpolated onto t_{base})

$$\text{fitness} = \text{sqrt}\left(\text{sum}\left(\text{expected intensity}_{0\text{-}n} - \text{sample intensity}_{0\text{-}n}\right)\right)$$

Figure 17: Fitness function

Figure 18 is a simple numerical example of the fitness function algorithm.

Step 1: t' is calculated from the sample chromatogram using the candidate solution (top of figure.) t' is then interpolated onto the base chromatogram.

Step 2: The expected intensity is derived from the base chromatogram.

Step 3: The actual sample intensity is derived.

Step 4: Fitness is calculated using the fitness function and the intensities found in steps 2 and 3.

Since the fitness calculated in this example is 0, the candidate solution, (0, 1, -0.5), evaluates to a perfect match with the base case chromatogram.

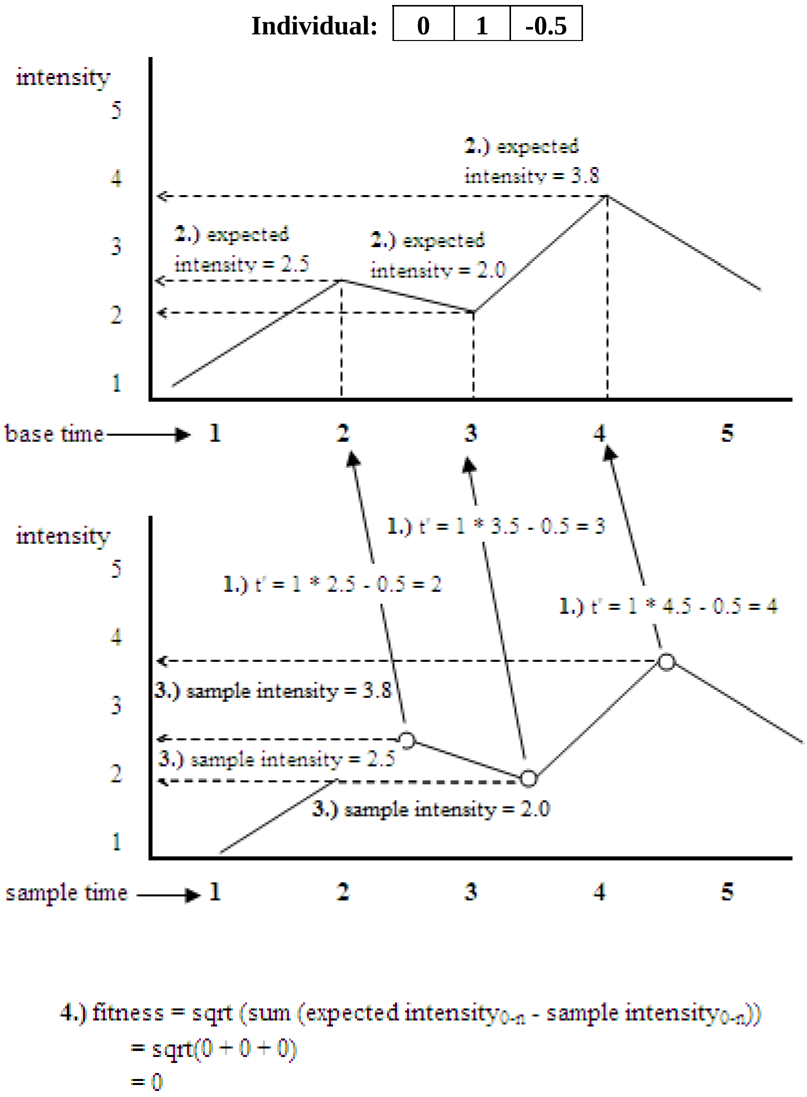

Individual: 0 1 -0.5
intensity
5
4
3
2
1
2.) expected intensity = 3.8
2.) expected intensity = 2.5
2.) expected intensity = 2.0
base time
1
2
3
4
5
1.) t' = 1 * 3.5 - 0.5 = 3
1.) t' = 1 * 2.5 - 0.5 = 2
1.) t' = 1 * 4.5 - 0.5 = 4
intensity
5
4
3
2
1
3.) sample intensity = 3.8
3.) sample intensity = 2.5
3.) sample intensity = 2.0
sample time
1
2
3
4
5
4.) fitness = sqrt (sum (expected intensity0-n - sample intensity0-n))
= sqrt(0 + 0 + 0)
= 0

Figure 18: Simple numerical example of fitness function

2.3 Similarity Analysis

In some of the following experiments, similarity must be quantified. Similarity of chromatograms before registration and similarity of solution chromatograms need to be numerically measured to confirm results. Two similarity measurements were developed for these experiments.

The first of these similarity measures is called an initial similarity measure. This refers to how similar two chromatograms are prior to any analysis. The algorithm used to determine the degree of similarity found the difference between intensities at each scan line and averaged the differences over the entire set of scan lines. The algorithm is outlined in figure 19.

```
sum = 0
for (each scan line)
{
        add intensity difference to sum
}
sum = sum / # of scan lines
```

Figure 19: Algorithm for determining similarity between two chromatograms

Figure 20 illustrates what is being measured by this initial similarity measurement. Given the first chromatogram set, the dark sections in the second set (bottom) represent the differences between the two. The dark sections are quantified with this similarity measure.

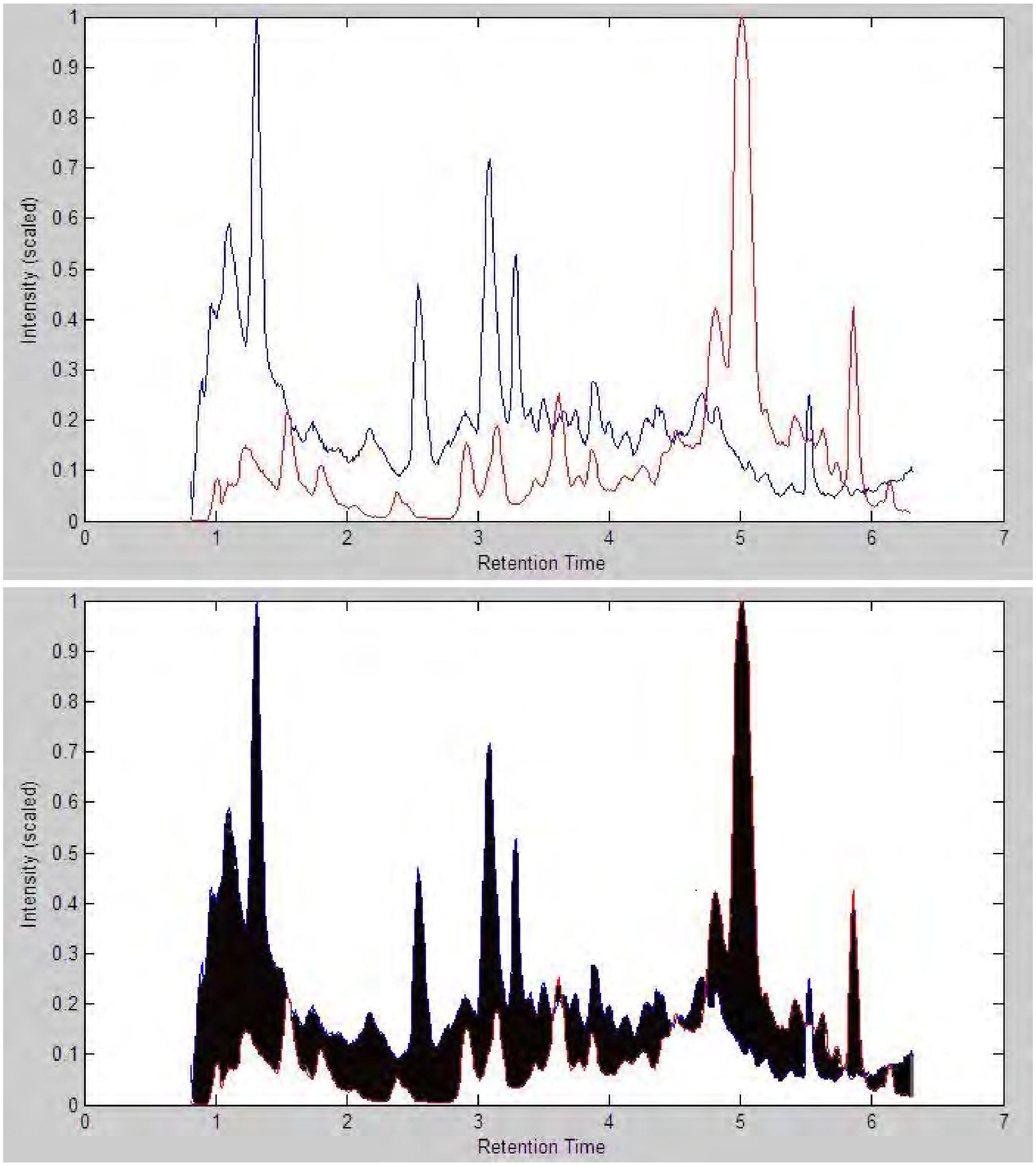


Figure 20: Illustration of similarity measure

To confirm the algorithm, the numerical rankings of similarity were compared with visual rankings of similarity. Figure 21 shows the range of numerical values associated with each category.

Initial Similarity	Category
0.00 - 0.02	Very Similar
0.02 - 0.03	Similar
0.03 – 0.1	Least Similar
0.1 +	Very Dissimilar

Figure 21: Numerical rankings of similarity

This measure is used throughout experiments 2, 3 and 4 and is referred to as initial similarity.

A second similarity measure was introduced in experiment 3 to measure the quality of alignment of a solution produced by the EP algorithm. Because distortions were assumed to only be along the time axis, this similarity measure compared the values in the base case time array with the values in the solution time array. The algorithm used for this measure, shown in figure 22, was similar to the initial similarity measure. The results of this measure were confirmed using data from experiments 1 and 2 (discussed in sections 3.1 and 3.2) whose alignment was very close.

```
diff = 0
for ( each element in time array)
{
        add time difference to diff
}
diff = diff / #of elements in array
```

Figure 22: Algorithm for solution alignment quality

This measure was essentially the average time difference between the solution and the base chromatogram with which it was supposed to align. Tests conducted using this algorithm with data from experiment 1 and experiment 2 concluded that an alignment quality value of less than 0.05 was acceptable alignment and as the value grew the alignment was of poorer quality.

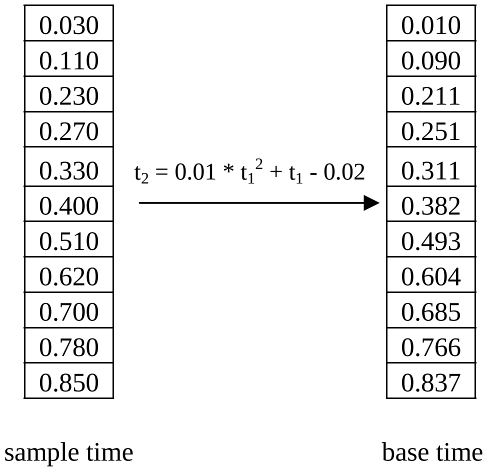
3. **EXPERIMENT DESCRIPTIONS AND RESULTS**

3.1 Experiment 1: Algorithm Validation

As an initial test of the evolutionary programming algorithm being used, one set of real data was distorted with a known second order polynomial:

$$c_1 * t^2 + c_2 * t + c_3$$

where the c_1, c_2 and c_3 represent the coefficients that will be searched for using the evolutionary algorithm. Each element in the sample case time array was distorted using the distortion function to create the base case array. Figure 23 shows how the distortion was performed.



$t_2 = 0.01 * t_1^2 + t_1 - 0.02$

sample time	base time
0.030	0.010
0.110	0.090
0.230	0.211
0.270	0.251
0.330	0.311
0.400	0.382
0.510	0.493
0.620	0.604
0.700	0.685
0.780	0.766
0.850	0.837

Figure 23: Known distortion used to test algorithm

Intensities were assumed to be unchanging because the algorithm only calculates transformations along the time axis. Therefore, the intensity array was the same for both chromatograms. Since the distortion function was known, the goal of the algorithm was to reproduce the coefficients of the function. This experiment verified the algorithm before it was used on two sets of real data whose distortion function was unknown.

The known distortion functions were second-order polynomials. Therefore, the search was designed to find a second-order transformation polynomial. The results of this initial experiment were favorable. Multiple distortion functions were tested. Sets of coefficients were chosen such that each coefficient was tested individually and all were tested together to confirm that there were no biases in the algorithm. Each distortion function was tested with varying numbers of individuals and varying numbers of generations. Figure 24 shows the test plan for experiment 1. The table on the left shows an "x" in the position of the coefficient(s) being tested. The table on the right shows the test runs for each distortion function. For each quantity of individuals, the algorithm was allowed to run for 100, 200, 300, 500 and 1000 generations. Some runs showed poor results immediately (low numbers of individuals and/or low numbers of generations) so only a few tests were run. For any combinations that showed potential, at least 10 tests were run.

	distortion	
c1	c2	c3
0	1	0
0	1	**x**
0	**x**	0
0	**x**	**x**
x	1	0
x	1	**x**
x	**x**	0
x	**x**	**x**

Individuals: 100				
100	200	300	500	1000
Individuals: 200				
100	200	300	500	1000
Individuals: 300				
100	200	300	500	1000
Individuals: 500				
100	200	300	500	1000
Individuals: 1000				
100	200	300	500	1000

Figure 24: Test plan for experiment 1

The combination of 500 individuals and 500 generations produced the best results in each case. Due to the stochastic nature of the initialization and mutation, it was expected that any combination of distortion coefficients would be equally difficult for the algorithm to locate. The following table shows five tests. For each run, the best fitness of the terminal generation was recorded. Each test represents the average over ten runs. The largest difference in fitness shown in figure 25 is 0.328779. Figure 26 will demonstrate that this is not a significant difference.

Expected			Calculated			Average Fitness
c1	c2	c3	c1	c2	c3	
0	1	0	-0.001615	0.804126	0.003122	0.381196 ± 0.14
0.1	1	0	0.098824	1.004844	-0.001186	0.145632 ± 0.09
0	0.75	0	0.000591	0.740612	0.010335	0.489382 ± 0.05
0	1	0.25	-0.001774	1.005522	0.260270	0.458169 ± 0.13
0.01	1.1	0.003	0.008986	1.102776	0.003500	0.233379 ± 0.03

Figure 25: Average results of experiment 1

The worst result (as seen in the above table) had an average fitness of 0.474411. Figures 26 and 27 show the visual results of this worst average. In figure 26, line #2 (solid) is distorted along the time axis with the function, 0.75 * t, to produce line #1(dotted.) In figure 27, the third line (added alongside line #1) represents the worst result of this experiment. Line #3 is supposed to align with line #1. It almost completely aligns which is why it is difficult to discern two lines.

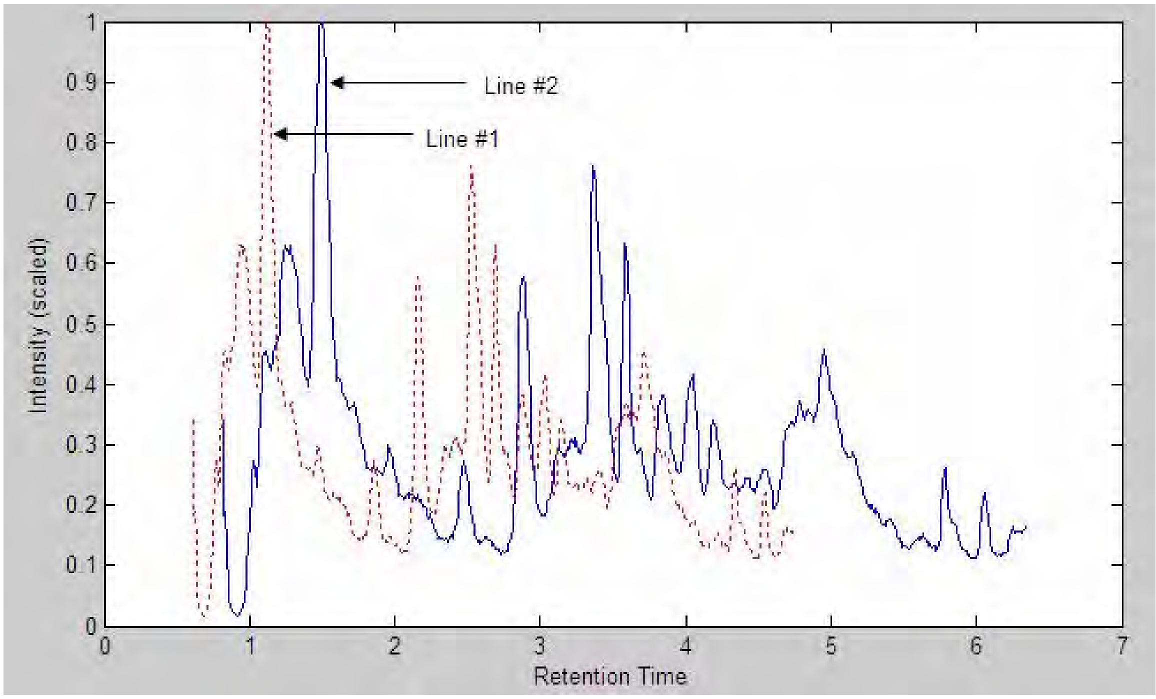


Figure 26: Time of line #1 = 0.75 * (time of line #2)

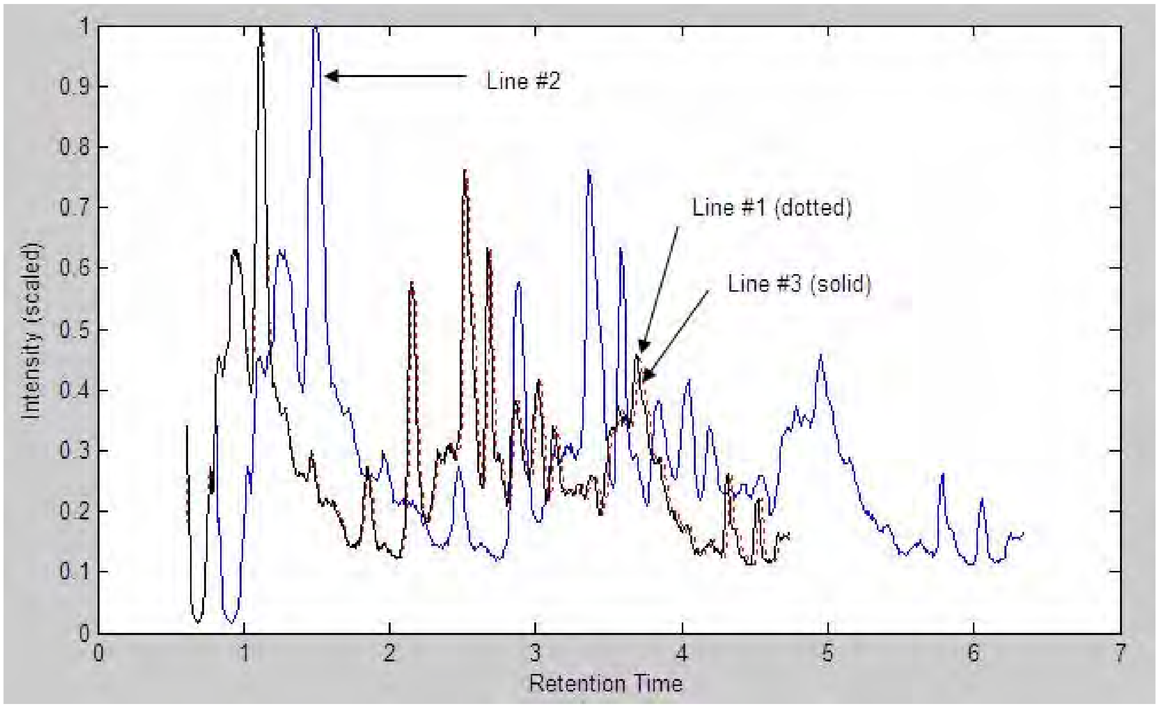


Figure 27: Worst results of experiment 1; Line #3 should match line #1

Figure 27 represents the worst results of experiment 1. The excellent quality of even the worst results demonstrates that experiment 1 performed well. Experiment 1 successfully found a second-degree polynomial transformation function. The transformation function found was almost equal to the distortion function used to produce the base chromatogram. The results of experiment 1 demonstrated that the algorithm was functioning properly and could be used to compare two sets of real data.

Because the distortion function was known in experiment 1, no similarity measures were necessary. A simple comparison of distortion function coefficients and the coefficients generated by the EP algorithm was enough to confirm quality.

3.2 Experiment 2: Comparing Initially Similar Real Data

The purpose of the second experiment was to use the algorithm with two sets of actual data. Data sets from samples taken from the same animal are similar. Some are more similar than others. Some of the difference in these samples are due to measurement error and/or noise in the instrument. The other differences are the biological variations that are caused by the experimental methods and need to be pinpointed.

The chromatogram data used for experiment 2 was divided into three sections based on their initial similarity measures: very similar data, similar data, and least similar data. Very similar data was almost an exact match while similar data has a few shifts and skews. Least similar data contained many shifts, skews and differences. Data falling into the very dissimilar category will be tested in experiment 3. Eight chromatograms in different combinations were used for this experiment. The chromatograms were all from the same animal with the same dosage but at different times- 0, 24 and 48 hours.

Figure 28 shows a visual example of each category of similarity used for this experiment. In the first case, the chromatograms are considered very similar. With a similarity value of 0.0092524, they are almost equal. The second case shows chromatograms that are similar. Their similarity value is 0.025315. The third set of chromatograms in figure 28 has a similarity value of 0.059379 and is considered least similar.

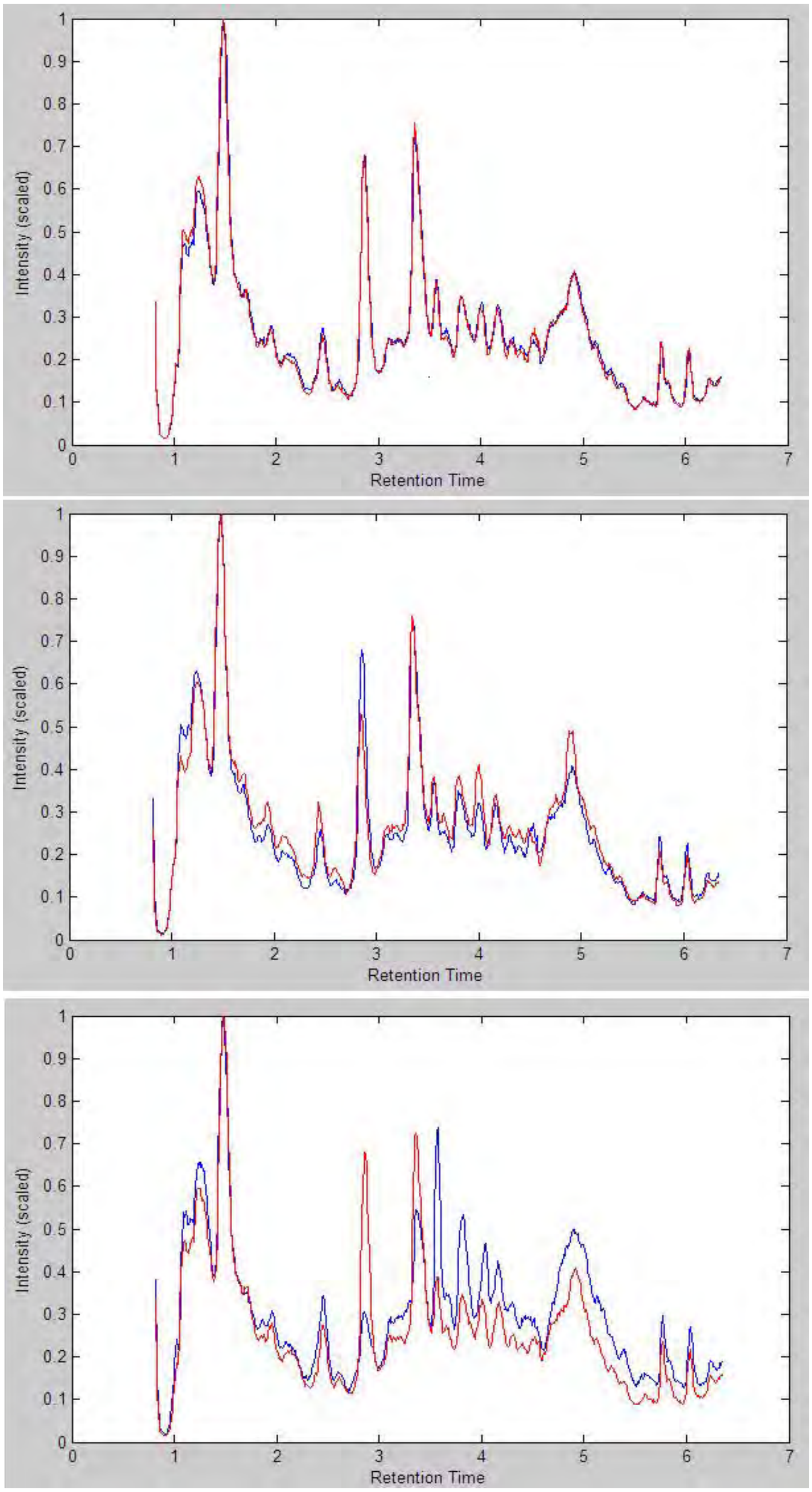


Figure 28: Examples of Very Similar, Similar and Least Similar Chromatograms

Even the least similar chromatograms in experiment 2 were initially much more closely matched than some of the distortions performed in experiment 1. Due to the initial similarities of chromatograms in experiment 2, a second degree polynomial was assumed.

In this experiment, the general test plan from experiment 1 was used. Two chromatogram sets from each category were tested with at least 10 runs of each individual-generation combination. Figure 29 shows the test plan for experiment 2.

Very Similar (1)
Very Similar (2)
Similar (1)
Similar (2)
Least Similar (1)
Least Similar (2)

Individuals: 100				
100	200	300	500	1000
Individuals: 200				
100	200	300	500	1000
Individuals: 300				
100	200	300	500	1000
Individuals: 500				
100	200	300	500	1000
Individuals: 1000				
100	200	300	500	1000

Figure 29: Experiment 2 test plan

Because this experiment used two real data sets rather than one real set and one set distorted with a known function, it took higher numbers of generations and individuals to find a good solution. With small numbers of individuals (<500), the algorithm prematurely converged to a solution that was not optimal. Due to the size of the search in this experiment, it was necessary to start the algorithm with many options for potential solutions. The bad candidate solutions were weeded out quickly, leaving many good potential solutions. Also, few generations (< 500), did not give the algorithm enough time to reach a good solution. More than 1000 generations did not produce significantly higher quality results and therefore were considered unnecessary. The evolutionary programming algorithm worked best with populations of 500 to 1000 individuals and 1000 generations.

Because the noise and errors caused by the instrument and humans were even slighter than the distortions introduced in experiment 1, experiment 2 performed very well. Due to the similarities in these data sets, the algorithm typically settled on an approximately linear distortion function. The table below shows the average results for the three types of data sets.

	Average			
	c_1	c_2	c_3	fitness
Very Similar	-0.00097	0.99999	0.01592	0.02617
Similar	-0.00255	1.00466	0.01035	0.03959
Least Similar	-0.00185	1.00582	0.00443	0.05943

Figure 30: Average results for experiment 2

As figure 30 demonstrates, although all results for experiment 2 were very favorable, the fitnesses of the candidate solutions for the least similar chromatograms were of lower quality than those of very similar chromatograms. Figures 31, 32 and 33 show samples of results from each category. In all three figures, line #1 is the sample chromatogram while line #2 is the base which is being aligned with. Line #3 represents the solution produced by the evolutionary programming algorithm. Figure 31 shows the results from the very similar category. The chromatograms are initially almost equal. The 3 lines are aligned so closely that they almost appear to be a single line. However, the close-up view shows that they are in fact three individual lines. The solution alignment value for this result is 0.0055. Because the difference between scan lines is approximately 0.6 seconds, the difference shown is not significant.

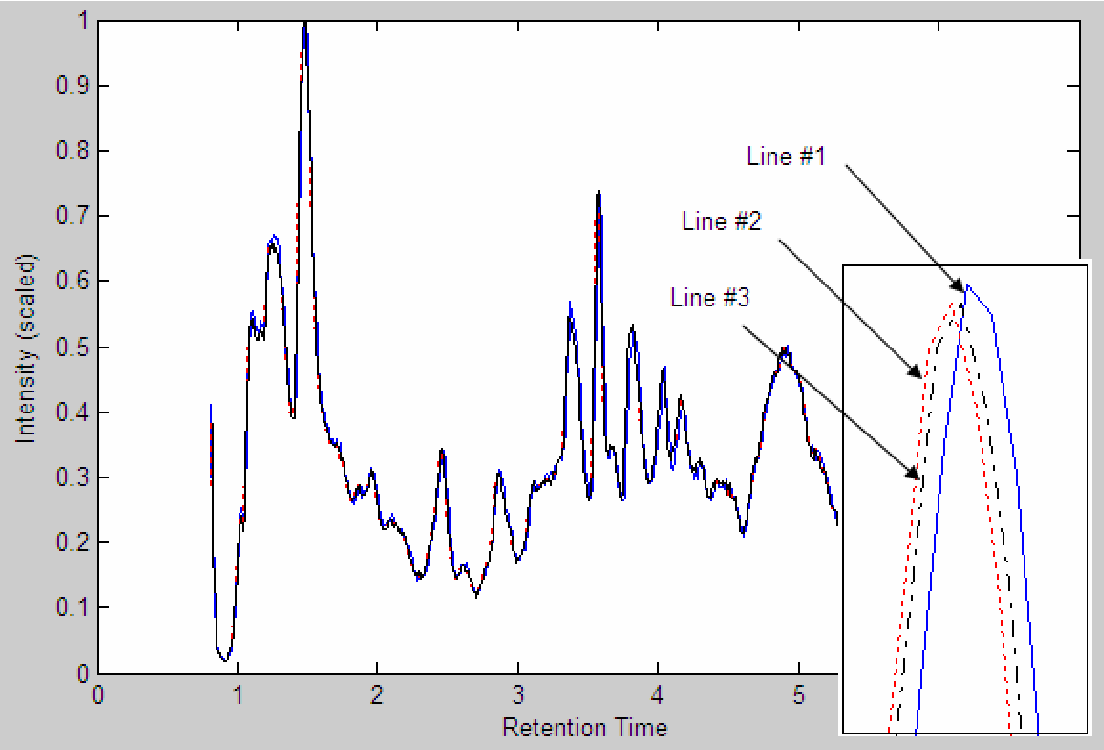


Figure 31: Very similar chromatograms and results from EP algorithm

Figure 32 shows a result from chromatograms in the similar category. While the initial chromatograms were slightly less similar than in figure 31, the results were still very favorable. A very low solution alignment value (0.0057) was found for this solution too. Again, the difference is less than a scan line and therefore could not be a close match.

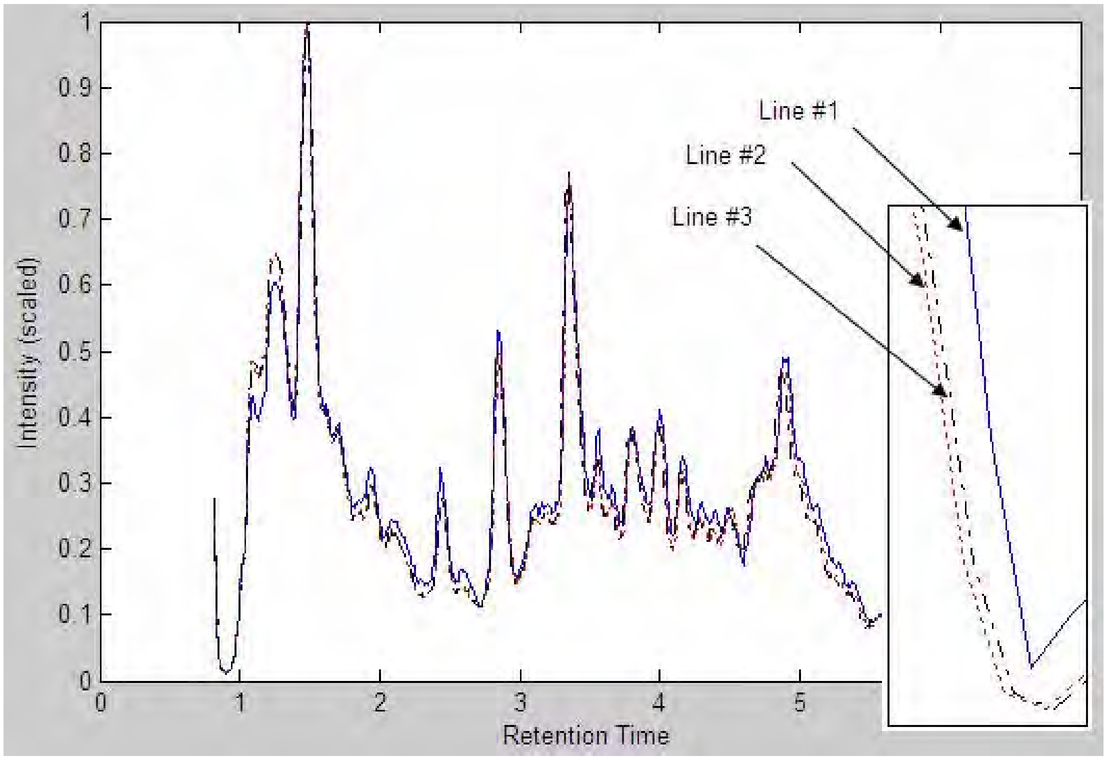


Figure 32: Similar chromatograms and results from EP algorithm

Lastly, figure 33 shows a result from chromatograms that were categorized into the least similar category. Again, a low solution alignment value was found (0.032). While this value is not as low as the previous cases, the match is still excellent.

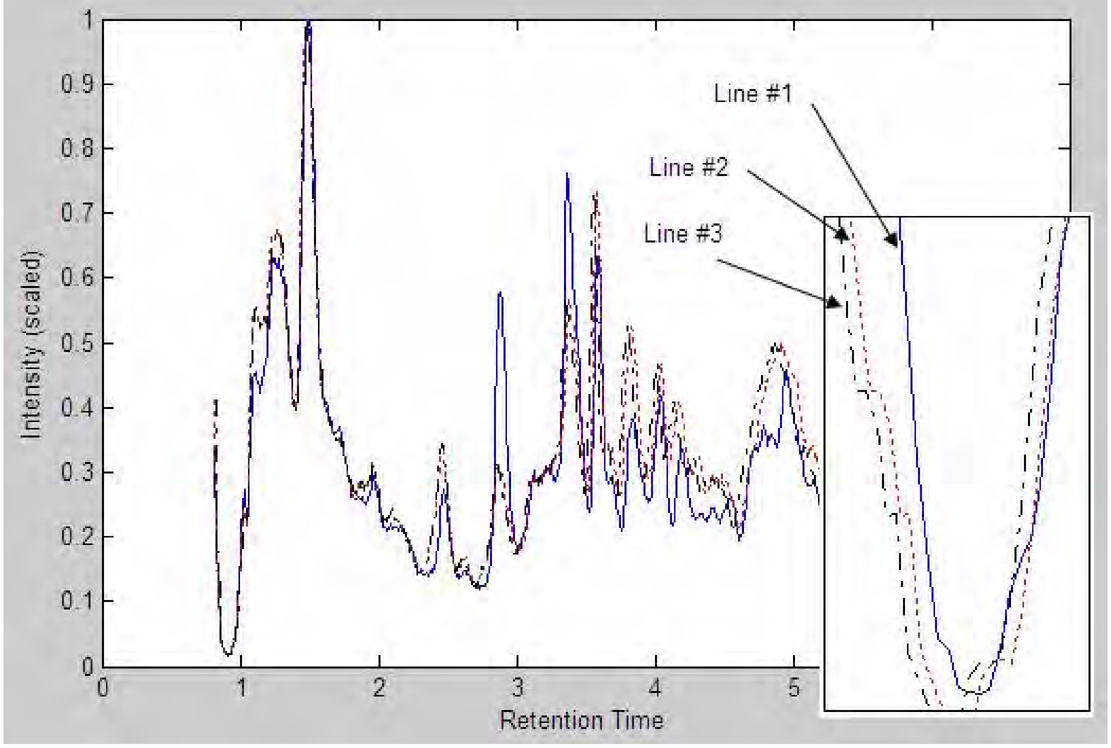


Figure 33: Least similar chromatograms and results from EP algorithm

An interesting anomaly was observed during this experiment. As time increased, the quality of the match of the results was increasingly worse. It is speculated that these changes over time are somehow related to changes in state as the solution moves through the LC/MS instrument. This anomaly is addressed in section 3.4.

3.3 Experiment 3: Comparing Initially Dissimilar Real Data

The third experiment tested data from different animals. These chromatograms are, for the most part, very different. While the chromatograms in experiments one and two could be visually confirmed to be similar, the chromatograms in experiment three were not as simple. The initial similarity measure used in experiment 2 was also used for initial data in experiment 3.

In experiment 2, all chromatograms were in the least similar category or better because they had initial similarity measures of less than 0.1. The chromatograms used in experiment 3 had initial similarity measures of 0.1 and greater. Figure 34 shows how different two of these chromatograms can be.

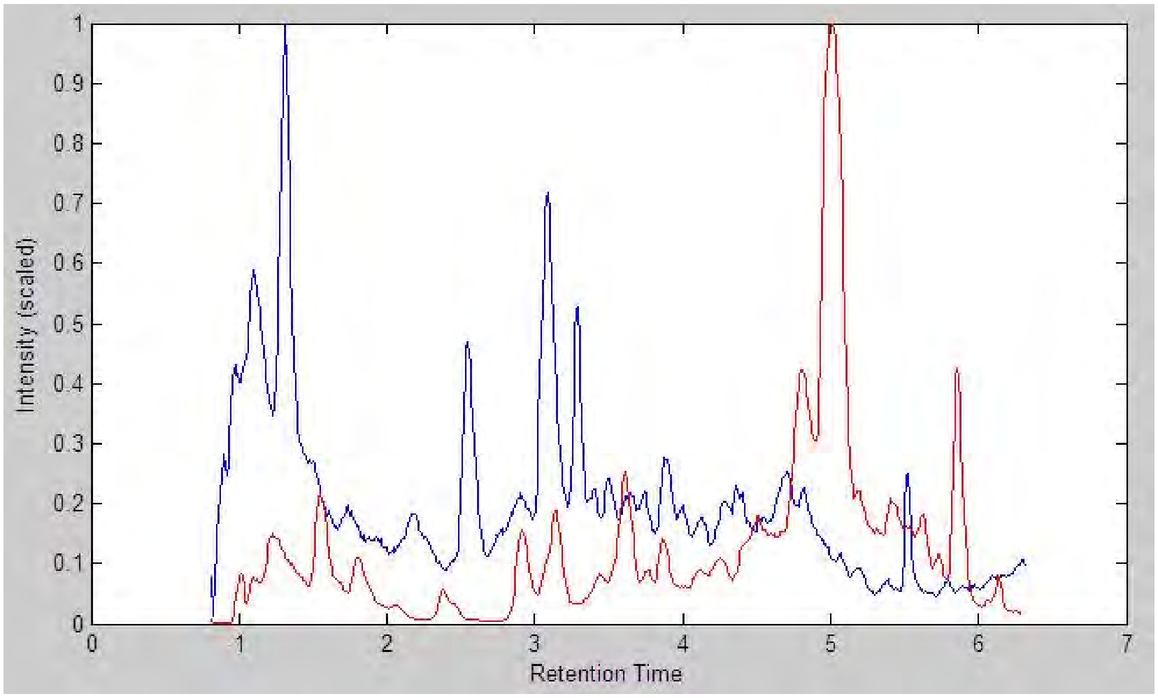


Figure 34: Two chromatograms used in experiment 3 – similarity measure = 0.16717

Due to the poor quality of similarity measures for all data used in experiment 3, there was no division into very similar, similar and least similar categories. Five very dissimilar chromatogram sets were used for this experiment. The test plan used in experiments 1 and 2 was used as a starting point for experiment 3 and a second-degree polynomial transformation function was a starting assumption. Figure 35 shows the test plan for experiment 3.

Very Dissimilar (1)
Very Dissimilar (2)
Very Dissimilar (3)
Very Dissimilar (4)
Very Dissimilar (5)

Individuals: 100				
100	200	300	500	1000
Individuals: 200				
100	200	300	500	1000
Individuals: 300				
100	200	300	500	1000
Individuals: 500				
100	200	300	500	1000
Individuals: 1000				
100	200	300	500	1000

Figure 35: Experiment 3 test plan

It was immediately obvious that experiment 3 would require both an increase in individuals and an increase in generations. The fitness of the candidate solutions was improving slightly but the results were still poor compared to the results found in experiments 1 and 2. An additional test plan was added to experiment 3. Figure 36 shows the additional tests that were run.

Very Dissimilar (1)
Very Dissimilar (2)
Very Dissimilar (3)
Very Dissimilar (4)
Very Dissimilar (5)

Individuals: 1000				
1000	2000	3000	4000	5000
Individuals: 1200				
1000	2000	3000	4000	5000
Individuals: 1500				
1000	2000	3000	4000	5000
Individuals: 1700				
1000	2000	3000	4000	5000
Individuals: 2000				
1000	2000	3000	4000	5000

Figure 36: Additional tests

These tests proved to be very time-intensive yet they did not produce better quality solutions. Regardless of the individual-generation combination, these 5 chromatograms continuously produced alignment quality values of greater than 0.05. In many cases, the alignment quality values were greater than 1.0. These values led to the conclusion that the algorithm would not find a solution for these very different chromatograms. This problem is potentially due to the fitness function and should be investigated in future works. There is also a possibility that a solution does not exist at all. Figure 37 shows one example of the quality of results achieved in experiment 3. Line #1 is the sample chromatogram while line #2 is the base chromatogram. Line #3 is the chromatogram resulting from the values obtained from the evolutionary programming algorithm.

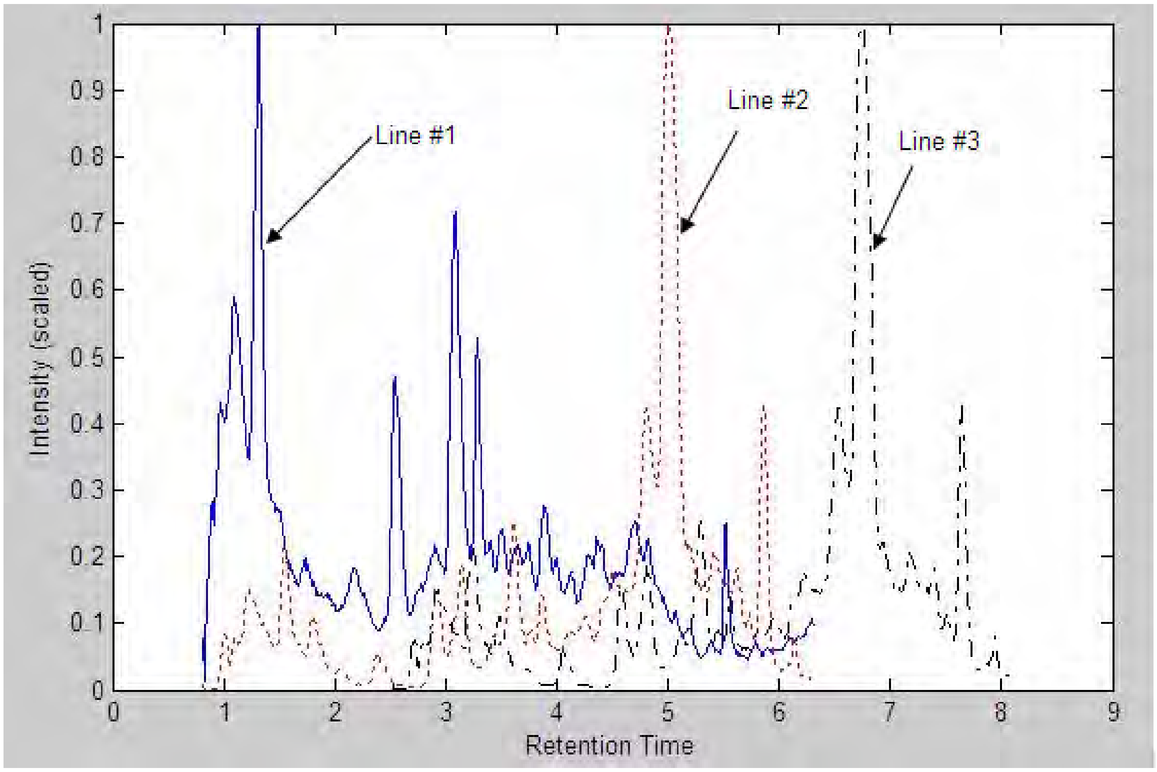


Figure 37: Results from exercise 3- alignment quality value = 1.5443

Although none of the results from experiment 3 were high quality, it was noted that the more similar the data sets, the better the results of the algorithm.

3.4 Experiment 4: Additional Tests

Based on results observed in the previous three experiments, it was determined that additional tests might provide more information about the data sets and potential alternative solutions to registering the chromatograms. These additional tests explored the data sets more thoroughly. It was hoped that they would provide a better understanding of how the data is laid out and what might be done to correlate data sets that are very dissimilar. Unless otherwise noted, these experiments were performed with the previously observed optimal individual-generation combination for the data set.

3.4.1 Adding terms to the polynomial

The first test attempted to add additional terms to the polynomial in hopes of getting better results from the algorithm. More terms in the polynomial will add more complexity and potentially lead to better quality matches. In experiment 2, it was observed that the match quality decreased as time increased. Figure 11 showed a second degree polynomial. As base time increased, sample time increased more rapidly. More terms in the polynomial would make this increase even more rapid. These increases in times seemed to be occurring in chromatogram registration.

While this seemed like a reasonable idea, the algorithm produced coefficients, although very small, for the higher order elements of the polynomial. This caused very unpredictable results in the solution chromatogram and solution quality values of 0.1 or greater. When using more than two terms in the polynomial, if a perfect solution was not found, it was more likely to be a very poor solution. The following example shows why this could lead to many problems.

Polynomial:	$a * t^5 + b * t^4 + c * t^3 + d * t^2 + e * t + f$
Solution:	*a = 0; b = 0; c = 0.01; d = 0; e = 1.2; f = 1*
Solution from EP:	*a = 0.001; b = 0.001; c = 0.03; d = 0.0001; e = 1.2; f = 1.1*
Assume:	*t = 0.5; 1.5; 4.0*
Expected results:	*1.60125; 2.83375; 6.44000*
EP results:	*1.70386; 3.01413; 9.10160*

The example shows that as time increases, the expected results and the EP-produced results become increasingly more different. Figure 38 illustrates this example. The broken line shows the expected results while the solid line shows the results produced by the EP. Although the differences are not initially significant, as time increases, the results from the EP are very different from the expected results.

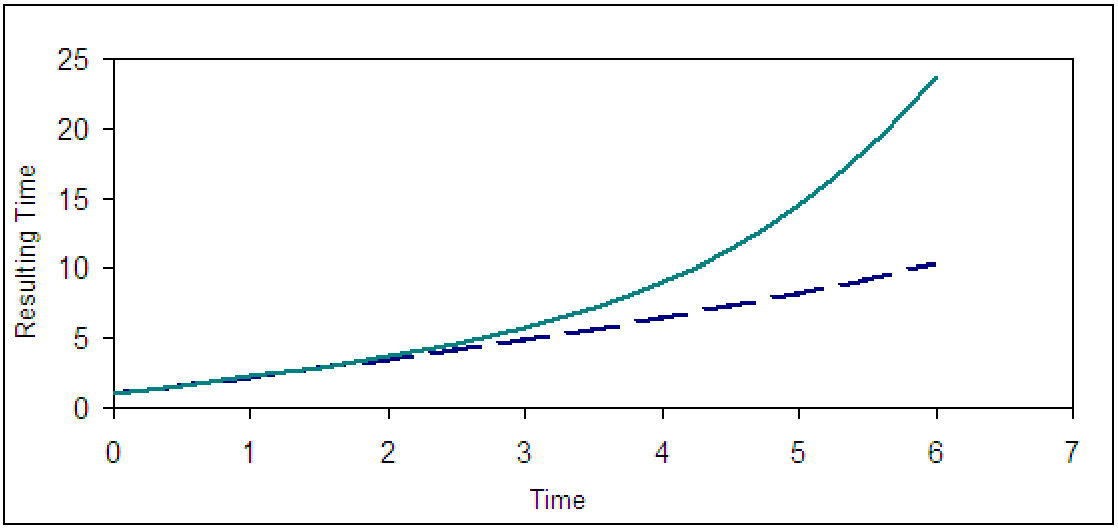


Figure 38: Example of adding more coefficients to polynomial

This experiment suggested that unless an exact solution was found, the second-order polynomial was the best transformation function for registering the chromatograms.

3.4.2 Considering saturation of intensity

In this research, it was known that the peak intensity values were saturated at values over 200. This knowledge led to the hypothesis that the values over 200 might be significant in some way. It was concluded that these values might be the key to matching these chromatograms or that they might be hampering the algorithm from finding a match if they are included in the chromatogram.

First, all values fewer than 200 were excluded from the chromatogram analysis. Next, values greater than 200 were excluded from the chromatogram analysis. Much of the data tested in experiments 2 and 3 was reevaluated using this method.

Visual evaluation of this data showed that when excluding values greater than 200, the initial similarity of the chromatograms was of better quality. Also, when only the values over 200 were analyzed, the visual analysis showed worse quality. Figure 38 shows the similarity value results for this experiment. Eight of the chromatogram sets from experiment 2 are shown in the first table. All 5 of the chromatogram sets from experiment 3 are shown in the second table. Positive percent differences represent an improvement in similarity. Those values are highlighted in the tables.

Similarity Values (experiment 2)				
all intensities	intensities < 200	% difference	intensities > 200	% difference
0.00925	0.00927	-0.18590	0.01655	-78.90493
0.01894	0.02695	-42.29226	0.02020	-6.61493
0.02328	0.02044	12.21974	0.04808	-106.51147
0.02532	0.01621	35.95497	0.03694	-45.90954
0.02591	0.02184	15.70701	0.05558	-114.47978
0.03800	0.04141	-8.97966	0.03185	16.16707
0.04804	0.03179	33.82941	0.08313	-73.03097
0.05938	0.05302	10.71759	0.05468	7.91694

Similarity Values (experiment 3)				
all intensities	intensities < 200	% difference	intensities > 200	% difference
0.10401	0.12287	-18.13287	0.09450	9.14239
0.15374	0.13544	11.90321	0.13716	10.78444
0.16717	0.11932	28.62356	0.23050	-37.88359
0.16782	0.13381	20.26576	0.19130	-13.99118
0.20567	0.15944	22.47776	0.23680	-15.13590

Figure 39: Similarity values when some intensities are removed

Overall, chromatogram sets constructed with only intensities greater than 200 did not show significant improvement over the initial sets that used all intensities. Only about 25% showed improvement and the improvements were not as great as in the reverse experiment. When these new chromatograms were analyzed with the evolutionary programming algorithm, the resulting candidate solutions' fitnesses were all worse than the fitnesses of the same chromatograms including all intensity values. Figure 40 shows one result of removing values with intensities less than 200. Overall, the initial match quality is poor as well as the solution quality. Figure 42 shows the solution qualities of these results.

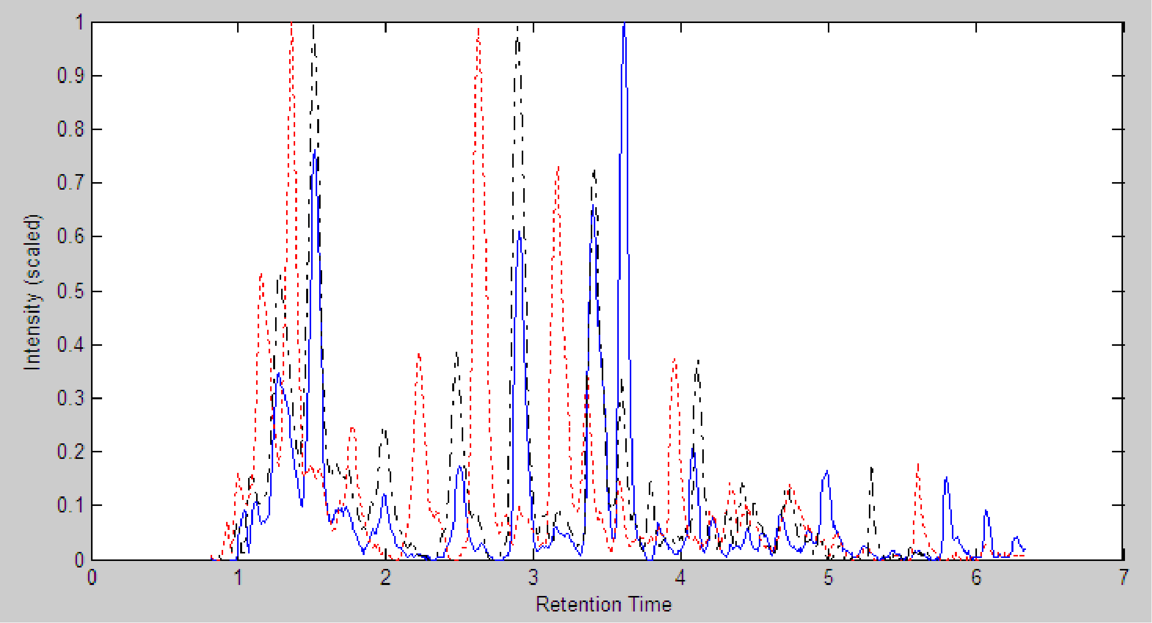


Figure 40: Results from removing values < 200

Interesting results were observed by eliminating values over 200. Since intensities over 200 were determined to be saturated, it was strongly suspected that these values could be inaccurate and therefore could be causing some of the noise and misalignment in the chromatograms. In approximately 72% of the cases, the chromatograms containing only values under 200 had better initial similarity values than the chromatograms containing all values. When these chromatograms were tested with the EP algorithm, some showed better solution alignment quality. On average though, the solution quality was about the same (or slightly worse) for chromatograms with good initial similarity values but was better for chromatograms whose initial alignment was poor. Figure 41 shows an example of chromatograms that had originally been classified as least similar based on their similarity value. The first set of chromatograms includes all values while the second set contains only intensity values below 200. As in previous figures, line #1 is the sample chromatogram while line #2 is the base chromatogram. Line #3 is the solution produced by the EP algorithm and is supposed to align with line #2.

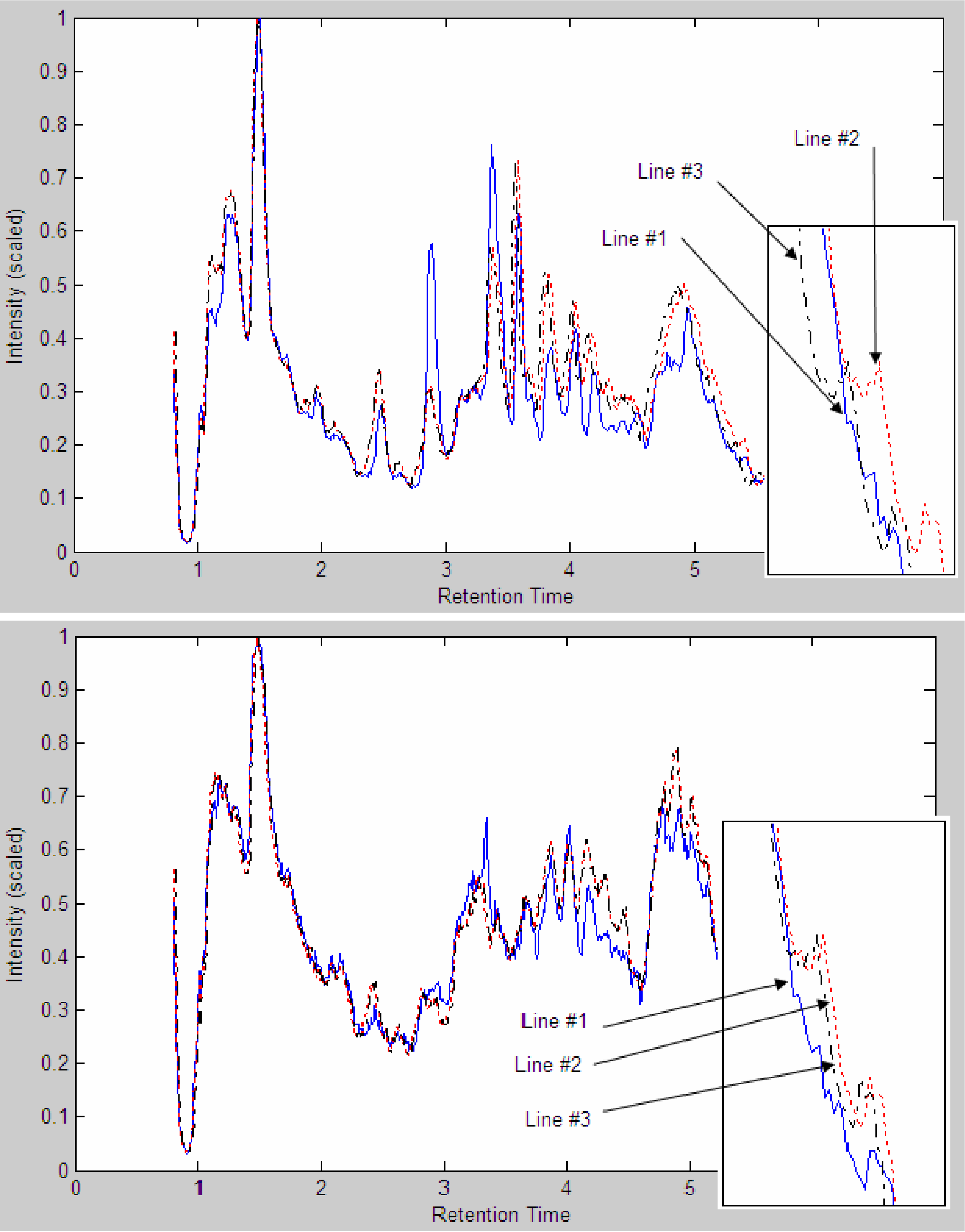
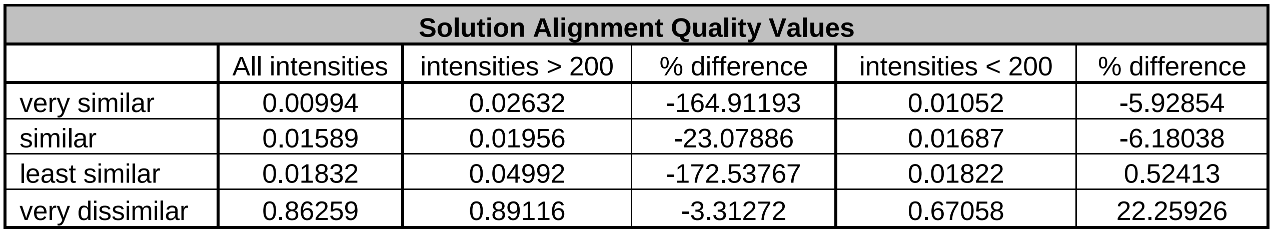


Figure 41: Top: A least similar example including all values
Bottom: The same least similar example excluding values over 200

Figure 42 shows the average solution quality values for the chromatograms (categorized by initial alignment) tested using these methods.



Solution Alignment Quality Values

	All intensities	intensities > 200	% difference	intensities < 200	% difference
very similar	0.00994	0.02632	-164.91193	0.01052	-5.92854
similar	0.01589	0.01956	-23.07886	0.01687	-6.18038
least similar	0.01832	0.04992	-172.53767	0.01822	0.52413
very dissimilar	0.86259	0.89116	-3.31272	0.67058	22.25926

Figure 42: Average solution alignment quality values

This experiment revealed that the values with intensities over 200 may be having a negative effect on the initial alignment of the chromatograms and potentially the ability of the algorithm to find the coefficients of the polynomial translation function.

3.4.3 Separating chromatograms

Experiment 2 led to the observation that alignment of the chromatograms is better at the low end of the time scale than the high end. Figure 43 (top) illustrates this concept. The first half of the resulting chromatogram aligns so closely with the base chromatogram that they form one line (zoomed in picture on the left.) As time increases, the alignment becomes worse (zoomed in picture on the right.)

This alignment issue led to the last additional experiment. To solve this problem, the chromatograms were divided into sections and each section was run through the algorithm as an individual chromatogram. After results were gathered for each section, they were combined to produce a single result chromatogram. For this experiment, chromatogram data was divided into halves and each half was treated like a separate chromatogram. As with previous experiments, this method led to improvements in some chromatograms but not all. While there are still slight flaws, this method occasionally created better results than using the entire chromatogram. Figure 43 shows the results when the chromatograms were split into halves.

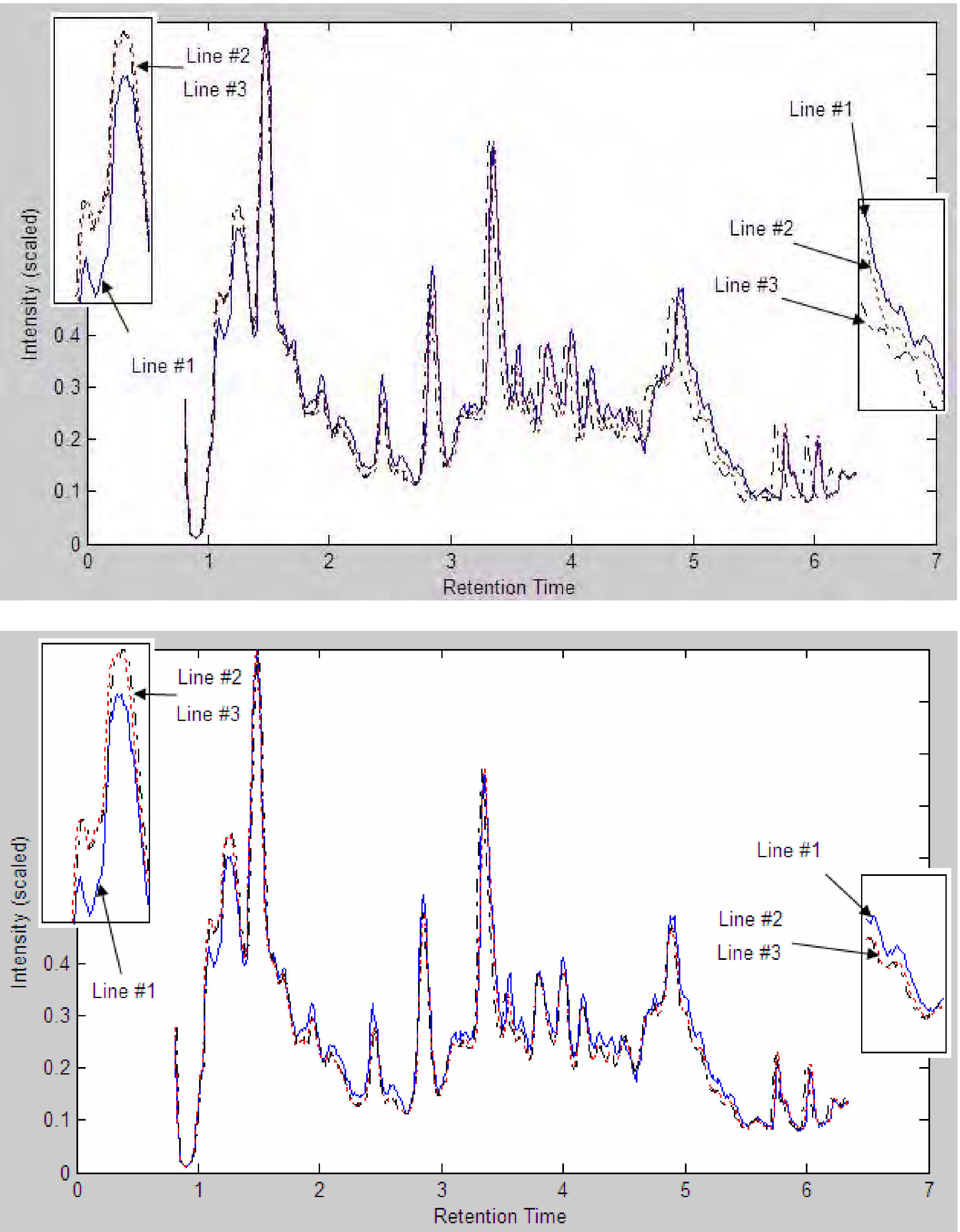
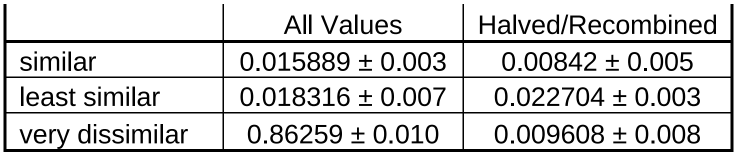


Figure 43: Top: Alignment worsens as time increases
Bottom: Results of splitting chromatograms in halves

Figure 44 shows the average solution quality results for this experiment.

Solution Alignment Quality Values



	All Values	Halved/Recombined
similar	0.015889 ± 0.003	0.00842 ± 0.005
least similar	0.018316 ± 0.007	0.022704 ± 0.003
very dissimilar	0.86259 ± 0.010	0.009608 ± 0.008

Figure 44: Solution alignment quality values for chromatogram halving

While the average quality got significantly worse in each case, this experiment showed potential with some chromatograms (as seen in figure 42.) The key is to divide the chromatograms in enough sections and to get a good quality match in each section. If all divisions produce a good solution alignment except for one, that one could cause the entire chromatogram to have a poor solution alignment quality value.

4. CONCLUSIONS AND FUTURE WORK

This thesis demonstrated that a polynomial transformation function between two similar chromatograms can be found using evolutionary programming. However, when chromatograms were not initially similar it was difficult or even impossible to find a good quality alignment using just the chromatograms. Experiment 2 demonstrated that the algorithm would work with real data and that the more similar the chromatograms the better the algorithm performed. Experiment 3 showed that the algorithm was not capable of aligning very dissimilar chromatograms.

The data provided was assumed to only be distorted along the time axis. This somewhat explains the poor results of experiment 3. It might be unreasonable to assume all chromatograms are able to be registered. Chromatograms produced from samples taken from different animals which were given different dosages should be expected to be biologically different. Further investigation into the usefulness of aligning these chromatograms should be conducted before much more work is put into aligning these chromatograms. If these different chromatograms are to be correlated in some way, future work on the algorithm should consider differences in intensities and masses along with retention times.

The additional tests in experiment 4 provided some insight into the data provided by the LC/MS instrument. With the exception of adding terms to the polynomial, each of the experiments showed potential with at least a few chromatograms. None, however, worked consistently well with all chromatogram sets. Future work with this research should incorporate these experiments. One way to incorporate these is to add components to the algorithm that try these tests and use them only if they improve the initial similarity value or the quality of the solution.

In experiment 4, dividing the chromatogram into halves showed promise. For dissimilar chromatograms, dividing into several sections could improve alignment. Future work with the algorithm should include methods for determining when a chromatogram should be divided into smaller sections and how many sections are necessary. It would be wasteful of resources to automatically divide each chromatogram into a certain number of sections so decisions about division should be based on a predetermined value, perhaps the initial similarity value. A reasonable maximum number of sections should also be determined to prohibit chromatograms from being divided into many sections.

With additional work in some areas, the recommended solution for improving this research is the following. Initial alignment measures should be taken for all intensity data values, just intensity values under 200 and just intensity values over 200. Whichever of these receives the best initial alignment value should be used to find the polynomial transformation function. The chromatograms should always be divided into two separate chromatograms (first half, second half.) Based on the initial alignment value, chromatograms should be divided into additional sections with the maximum number of sections being the number of minutes— so there are no fewer than 100 scan lines per section.

The algorithm developed in this experiment will ultimately lead to a solution that will register the entire matrix of intensities rather than their resulting chromatograms.

Like the chromatogram, the matrix may need to be divided into sections, potentially both vertically and horizontally. The algorithm for registering the matrix should include a method for determining what divisions are needed. The full package, when developed, will hopefully be a useful tool for the scientific community to analyze large quantities of data quickly. Besides the liquid chromatography/ mass spectrometry and metabonomics application, this package could be used to automate many other data analysis that will help scientists further their research.

5. REFERENCES

1. Wilfried M. A. Niessen, *Liquid Chromatography-Mass Spectrometry*, 2nd Ed., New York: Marcel Dekker, Inc., 1999, pp 3 - 34.

2. William M. Spears, Kenneth A. DeJong, Thomas Baeck, David B. Fogel, Hugo de Garis, "An Overview of Evolutionary Computation," 1993 European Conference on Machine Learning, 1993.

3. Darrell Whitley, "An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls," *Information and Software Technology*, Vol. 43, No.14 (2001), pp 817 – 831.

4. Lisa Gottesfeld Brown, "A Survey of Image Registration Techniques," *ACM Computing Surveys*, Vol. 24, No. 4 (1992), pp 325 – 376.

5. "Chromatogram," Internet: Library for Science, http://www.chromatography-online.org/topics/chromatogram.html, accessed January 19, 2007.

6. A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing,* New York, Springer, 2003.

7. Par Jonsson, Stephen J. Bruce, Thomas Moritz, Johan Trygg, Michael Sjostrom, Robert Plumb, Jennifer Granger, Elaine Maibaum, Jeremy K. Nicholson, Elaine Holmes, Henrik Antti, "Extraction, interpretation and validation of information for comparing samples in metabolic LC/MS data sets", *The Analyst*, Vol 130 (2005), pp.701 – 707.

8. "History of Mass Spectrometry," Internet: Scripps Center for Mass Spectrometry, http://masspec.scripps.edu/MSHistory/whatisms.php#Basics, accessed March 1, 2007

9. "Definition of Metabonomics," Internet: MedicineNet, http://www.medterms.com/script/main/art.asp?articlekey=38634, accessed March 1, 2007

10. John Lindon, "Metabonomics – Techniques and Applications", *Business Briefing: Future Drug Discovery*, 2004.

11. Jun Yang, Xinjie Zhao, Xiaoli Liu, Chang Wang, Peng Gao, Jiangshan Wang, Lanjuan Li, Jianren Gu, Shengli Yang, Guowang Xu, "Performance Liquid Chromatography—Mass Spectrometry for Metabonomics: Potential Biomarkers for Acute Deterioration of Liver Function in Chronic Hepatitis B," *Journal of Proteome Research*, Vol. 5 (2006), pp. 554-561.

12. A.E. Eiben, M. Schoenauer, "Evolutionary Computing", *Information Processing Letters*, Vol 82 (2002), pp. 1-6.

13. Ji-Hui Zhang, Xin-He Xu, "An Efficient Evolutionary Programming Algorithm", *Computer Operation Research*, Vol. 26 (1999) pp. 645 – 663.

14. "Creatinine", Internet: Answers.com, http://www.answers.com/creatinine, Accessed March 3, 2007

15. Barbara Zitova, Jan Flusser, "Image Registration Methods: A Survey," *Image and Vision Computing*, Vol 21 (2003), pp.997-1000.

16. Bharti Temkin, Sreeram Vaidyanath, Eric Acosta, "A High Accuracy, Landmark-based, Sub-pixel Level Image Registration Method," *International Congress Series*, Vol. 1281 (2005) pp.254-259.