The Analysis of Mass Spectrometry Data to Resolve and Quantify Peptide Peaks in Cerebral Stroke Samples: An Evolutionary Computation Approach. [*]

Julio J. Valdés
National Research Council Canada
Institute for Information Technology
M50, 1200 Montreal Rd.
Ottawa, ON K1A 0R6
julio.valdes@nrc-cnrc.gc.ca

Alan J. Barton
National Research Council Canada
Institute for Information Technology
M50, 1200 Montreal Rd.
Ottawa, ON K1A 0R6
alan.barton@nrc-cnrc.gc.ca

Arsalan Haqqani
National Research Council Canada
Institute for Biological Sciences
M50, 1200 Montreal Rd.
Ottawa, ON K1A 0R6
arsalan.haqqani@nrc-cnrc.gc.ca

ABSTRACT

A preliminary investigation of cerebral stroke samples injected into a mass spectrometer is performed from an evolutionary computation perspective. The detection and resolution of peptide peaks is pursued for the purpose of automatically and accurately determining unlabeled peptide quantities. A theoretical peptide peak model is proposed and a series of experiments are then pursued (most within a distributed computing environment) along with a data preprocessing strategy that includes *i)* a deisotoping step followed by *ii)* a peak picking procedure, followed by *iii)* a series of evolutionary computation experiments oriented towards the investigation of their capability for achieving the aforementioned goal. Results from four different genetic algorithms and one differential evolution algorithm are reported with respect to their ability to find solutions that fit within the framework of the presented theoretical peptide peak model. Both unconstrained and constrained (as determined by a course grained preprocessing stage) solution space experiments are performed for both types of evolutionary algorithms. Good preliminary results are obtained.

Keywords

mass spectroscopy, proteomics, medicine, genetic algorithms, differential evolution, evolutionary computation, model fitting

1. INTRODUCTION

Stroke is the second leading cause of death and the most common cause of disability in the world. To relieve the heavy burden of stroke, we need to understand its mechanisms that will form the basis of improved prevention and treatment. Stroke-specific protein changes may serve as surrogate markers of pharmacodynamic efficacy, toxicity, or outcome in preclinical and clinical development of new medicines. Proteomics platforms that can efficiently identify and quantify changes in proteins related to disease (e.g., stroke) offer great promise for advancing biomedical research and the development of novel medicines.

Mass spectrometry is an analytical technique used to measure the mass-to-charge ratio (m/z) of ions. It is most generally used to find the composition of a physical sample by generating a mass spectrum representing the masses of sample components. The technique has several applications, including: *i)* identifying unknown compounds by the mass of the compound and/or fragments thereof, *ii)* determining the isotopic composition of one or more elements in a compound, *iii)* determining the structure of compounds by observing the fragmentation of the compound *iv)* quantifying the amount of a compound in a sample using carefully designed methods (mass spectrometry is not inherently quantitative), *v)* studying the fundamentals of gas phase ion chemistry (the chemistry of ions and neutrals in vacuum), *vi)* determining other physical, chemical or even biological properties of compounds with a variety of other approaches.

Two of the most commonly used methods for quantitative proteomics are (i) two-dimensional electrophoresis (2DE) coupled to either mass spectrometry (MS) or tandem mass spectrometry (MS/MS) and (ii) liquid chromatography coupled to mass spectrometry (LC-MS).

In the 2DE-based approach, intact proteins are separated by 2DE, and the abundance of a protein is determined based on the stain intensity of the protein spot on the gel. The identity of the protein is now generally determined by MS analysis peptides after proteolysis of the protein spot. Since its inception in the mid-1970s, the 2DE-based approach has been routinely used for large scale quantitative proteomics analysis. The 2DE method, however, is limited in sensitivity and can be inefficient when analyzing hydrophobic proteins or those with very high or low mass. In addition, 2DE approach is difficult to automate and has a limited detection capacity for proteins with extreme ranges in pI values (the isoelectric point of proteins, which is the pH at which the net charge of the protein is zero), and for low abundance proteins.

The LC-MS-based approach, on the other hand, can be automated and can identify proteins with extreme masses and pI values.

[*] This research was conducted within the scope of the BioMine Project in the Integrated Reasoning Group (National Research Council Canada, Institute for Information Technology).

This approach is also more sensitive and can detect very low abundant peptide peaks. However, to correctly quantify the low abundant peaks, they need to be properly resolved from the background ”noise”. The LC-MS/MS based approach often uses stable isotope labeling techniques, e.g. with 15N, 13C, stable isotope labeling by amino acids in cell culture (SILAC), and isotope-coded affinity tags (ICAT), to provide relative quantification. While potentially providing the greatest accuracy, isotopic labeling has some disadvantages. Labeling with stable isotopes is expensive, and some labeling procedures involve complex processes and yield artifacts.

A ”label-free” LC-MS approach is based on the principle that the MS signal intensity of each peptide in a substantially similar sample analyzed under identical conditions is proportional to the abundance of the peptide within the dynamic range of the instrument. Therefore one may evaluate the relative abundance of a peptide in different, related samples by analyzing the samples under identical LC-MS conditions and by comparing MS signal intensity of the same peptide in different LC-MS runs. A disadvantage of such a label-free approach is that biological samples are usually very complex, and as a result, overlapping peptide peaks are often observed, which may be difficult to resolve. In order to accurately quantify peptide levels in LC/MS sample, not only do we need to identify and subtract the background noise but also need to deconvolve overlapping peaks.

The central dogma of Biology revolves around the idea that DNA molecules give rise to RNA molecules which give rise to protein molecules through a very complex and currently not completely understood process. One aspect that is being tackled, is that of attempting to understand the differences in quantity of protein molecules between different cells or tissues of various organisms. This problem is important because it has been noted that a lot of the changes in proteomics data are very subtle, but may lead to large phenotypic differences.

The purpose of this paper is to evaluate the possibilities of evolutionary algorithms, in particular, genetic algorithms and differential evolution in the detection and quantification of relevant peaks associated with peptides from mass-spectrometry data. The automation of modeling in the context of the high throughput mass spectrometry equipment will allow extensive data mining with high quality interpretation, thus facilitating the knowledge discovery process. The idea is to describe the mass spectrum with a simple mathematical model, and explore the performance of evolutionary algorithms in the adjustment of the model parameters. A specific goal is to decompose the spectrum into its constituent peaks isolated from the background noise. Such decomposition will allow the independent identification and characterization of the peptides present. If these operations can be performed reliably by an computational algorithm, the high throughput of the mass spectrometry equipment used for this kind of research could be pipelined and the process automated. This should be considered a a preliminary step in that direction.

2. THE DATA AND ITS PREPROCESSING

Data was collected after one biological sample (containing peptides extracted from brain synapse of a stroke-induced animal) was injected into a mass spectrometer operating in survey mode. MassLynx software (available from `http://www.waters.com`) was used to generate peak lists for each of the MS survey scans (usually $2,000-4,000$ per sample). Each list contains three types of information: *i)* mass over charge, which is very accurate with an error of ± 0.05 Daltons, *ii)* intensity (ion counts) and *iii)* time, which can have a high error of ± 10 min. Fig.1 shows an example of raw mass spectrometry data for a set of eluting peptides from one sample.

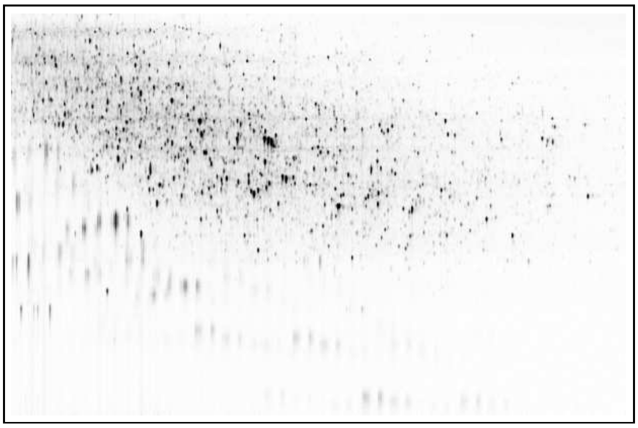

Figure 1: Raw Mass Spectrometry data collected from one biological sample and represented as an image. Y-axis: Peptide elution. X-axis: Mass to charge.

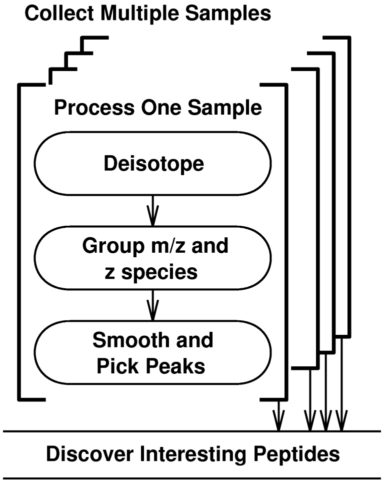

Figure 2: Outline of Mass Spectrometry Data Processing. The current approach focusses on the more accurate determination of measurements for one sample.

The general overview of the data analysis that was then performed on one sample is described in Fig.2. The first step involves a deisotoping process, which determines the specific charge values for each of the measured ions and removes all of the peaks with intensity less than 150. This threshold was selected due to the fact that the mass spectrometer would not be able to properly sequence lower abundant peptides. Then a series of heuristics were applied to filter the data in order to decrease the complexity of the later analysis (no data values were altered, only deleted): i) a series of partitioning steps were performed, leading to a set of partitions that each represent a measured peptide. ii) the partitions were then analysed in terms of the number of consecutive missing values – if more than a threshold were contiguous, then the partitions were divided. The intention is that if the mass spectrometer did not have sufficiently high measurements for an extended duration, then the peptide was probably not eluting, and so it would be safe to consider consecutive partitions as belonging to different peptides at this stage of processing (they could be merged later). iii) a weighted mass to charge value was then calculated for each partition, and iv) a filter was applied in order to remove multiple measurements collected for the same scan (time) and small partitions. Fig.3 shows the first three isotopic peaks and their sum, for one of the largest partitions. The first monoisotopic peak was selected from Fig.3 as being representative of a spectrum (i.e. small, medium and large peaks) that could then be more thoroughly investigated using evolutionary computation techniques.

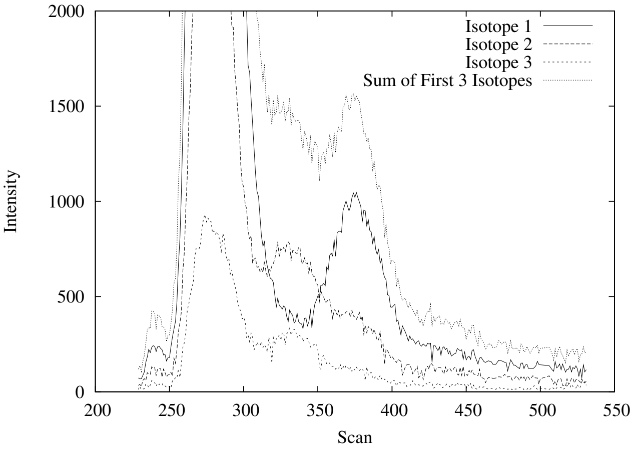

Figure 3: View of one representative spectrum for the first three isotopic peaks and their sum as extracted from the raw data (Fig.1) using the described preprocessing. The spectrum has been truncated at $2,000$, but rises to approx. $11,000$.

Numerically the spectrum is given by a collection of N values $\{x_i, y_i\}$, where $i \in [1, N]$. Usually the differences $x_i - x_{i-1}$, for $i \in [2, N]$ are not equal, and the spectrum has to be transformed into a sequence of regularly sampled values by interpolation. There are many procedures available for function interpolation, and the one used here is based on the computation of the best predicted value in a minimum expected squared error sense. The predicted value is given by the conditional expectation given x [8], [11]. For the general multivariate case in which a dependent variable y is a function of a vector $\vec{x}$, the conditional expectation is given by

$$E_{Y|\vec{X}}(\vec{X}) = \frac{\int_{-\infty}^{\infty} f \cdot f_{\vec{X}Y}(\vec{X}, y)dy}{\int_{-\infty}^{\infty} f_{\vec{X}Y}(\vec{X}, y)dy}$$

where $f_{\vec{X}Y}(\vec{X}, y)$ is the joint density. In practice, this function is unknown but it can be approximated using a Parzen estimator. If distance functions for $\vec{X}$ and y are defined as

$$D_{\vec{X}}(\vec{x}, \vec{x_i}) = \sum_{j=1}^{p} \left(\frac{x_j - x_{ij}}{\sigma_j} \right)^2$$

where $\vec{x}$ is an observed case and $\vec{x_i}$ are a training cases, and

$$D_Y(y, y_i) = \left(\frac{y - y_i}{\sigma_y} \right)^2$$

then the Parzen approximation is given by

$$g(\vec{x}, y) = \frac{1}{N c_{\vec{X}} c_Y} \sum_{i=1}^{N} e^{-D_X(\vec{x}, \vec{x_i})} e^{-D_Y(y, y_i)}$$

where N is the sample size and $c_{\vec{X}}$, c_Y are two normalizing constants ensuring integrability to unity. Using Parzen's estimate for the joint density, the predicted value for the dependent variable is given by

$$\hat{y}(\vec{x}) = \frac{\sum_{i=1}^{N} y_i \cdot e^{-D_X(\vec{x}, \vec{x_i})}}{\sum_{i=1}^{N} e^{-D_X(\vec{x}, \vec{x_i})}}$$

This scheme was used for constructing 303 regularly sampled spectrum values covering the range of unidimensional x values defined by its minimum and maximum. This interpolated spectrum was smoothed with a moving averages operator of length 7 (larger values might have removed narrow peaks and smaller values would not lead to smooth enough spectra). Other digital filtering schemes may be used as an alternative for moving averages, but in this preliminary investigation a parsimonious approach was adopted, thus leading to the selection of a simple smoothing operator.

Then the smoothed interpolated spectrum was explored for the occurrence of maxima, with their corresponding numeric characterization. The procedure used is given by the following steps:

i) Search for local maxima. If τ_{max} is a specified distance threshold measured in sampling interval units, a maximum is located at x_0 if $\forall x_i \in [x_0 - \tau_{max}, x_0 + \tau_{max}], y(x_0) \geq y(x_i)$. The collection of all maxima found is the set $\mathcal{P}$

ii) For each $p \in \mathcal{P}$ the derivatives at the flanks are approximated as $D = y_n - y_{n-1}$ (the sampling interval is assumed as 1, which is guaranteed by the interpolation procedure) and successively computed at either side of the analyzed maximum p.

iii) If $D \leq T_h$, where T_h is a pre-specified threshold, it is assumed that the background level has been reached. This is performed at each side of p determining its limits A and B (Fig-4).

iv) The peak amplitude and base width are obtained as shown in Fig-4. Other peak descriptors are also computed.

Screening strategies of this kind have been applied successfully for a long time in processing continuous measurements. For example, the detection and characterization of anomalies or interesting events in airborne electromagnetic data, is very similar to what is required in the case of the mass-spectrum for finding meaningful spectral peaks. The above described pre-processing procedure is inspired by the one introduced in [9] in the context of geophysical prospecting.

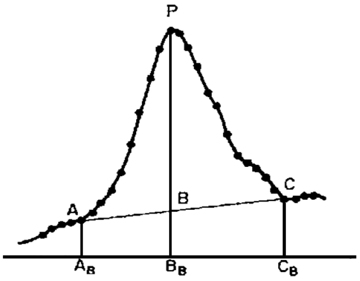

Figure 4: An example of a single peak. The distance PB is the peak amplitude and $A_B C_B$ the base width.

3. A MODEL FOR MASS SPECTRA

A simple model of a spectrum in the context of the studied problem is that of a collection of peaks corresponding to the peptides (possibly overlapping), molecule fragments, and a background noise. Most elution peptide profiles seem to follow a gaussian distribution and therefore an obvious first modeling candidate to analytically describe a spectral peak would be that of a gaussian function

$$g(x, A, x_0, \sigma) = A\ e^{-(\frac{x-x_0}{\sigma})^2} \qquad (1)$$

where A is the peak's amplitude, x_0 its location and σ a spread factor. When a collection of n-peaks is considered, the model is naturally extended as an additive aggregation of the constituent peaks

$$y(x) = \sum_{i=1}^{n} A_i\ e^{-(\frac{x-x_{0i}}{\sigma_i})^2}$$

where A_i, x_{0i} and σ_i are the corresponding amplitude, location and spread factor for the individual peaks. However, in many cases asymmetric peaks are experimentally observed and their shape depends upon the high pressure liquid chromatography (HPLC) and the type of chromatographic column used. Hence, the spreads or rates of decay are not the same at each side of the peak, which can be described by a model consisting of a collection of asymmetric peaks. Accordingly, the single and multiple peaks models turns into

$$\hat{g}(x, A, x_0, \sigma_L, \sigma_R) = \begin{cases} A\ e^{-(\frac{x-x_0}{\sigma_L})^2} & \text{if } x \leq x_0 \\ A\ e^{-(\frac{x-x_0}{\sigma_R})^2} & \text{otherwise} \end{cases} \qquad (2)$$

and

$$y(x) = \sum_{i=1}^{n} \hat{g}(x, A_i, x_{0i}, \sigma_{Li}, \sigma_{Ri}) \qquad (3)$$

The spectral background is slightly more complex than a constant intensity level. Usually a baseline is observed in the spectra, which slightly deviates from linear. It can be described as a low order power polynomial and the simplest non-linear model would be one with a single curvature. A spectrum model containing all of the previous features would be

$$y(x) = (ax^2 + bx + c) + \sum_{i=1}^{n} \hat{g}(x, A_i, x_{0i}, \sigma_{Li}, \sigma_{Ri}) \qquad (4)$$

This model will be used for approximating the observed spectrum used in the experiments.

4. EVOLUTIONARY COMPUTATION TECHNIQUES IN FINDING MODEL PARAMETERS

The problem of fitting the theoretical model given by Eq-4 to an observed mass spectrum can be approached using many optimization techniques: classical, evolutionary and hybrid. In the first case deterministic methods like Powell, Fletcher-Reeves or Levenberg-Marquardt could be applied. In the second, techniques like genetic algorithms, evolution strategies and particle swarm optimization are typical choices. In this preliminary approach, genetic algorithms (GA) and differential evolution (DE) were considered. In the case of genetic algorithms two variants were investigated: *i)* using real-valued chromosomes, with a fixed length given by $3 + (4 \cdot k)$, where 3 is the number of coefficients of the 2-nd order polynmomial trend and k is the number of expected spectral peaks (each one has 4 parameters), and *ii)* real-valued chromosomes with variable length [6], [5]. In this case, k is assumed unknown and has to be estimated during the evolutionary process like the other model parameters. In the case of differential evolution only a single strategy was applied.

Genetic algorithms are the most popular representative of the evolutionary computation family of algorithms [**?**], [3], [1], [2]. In this paper four types of GAs were considered: *i)* The standard simple genetic algorithm as described in [**?**]. This algorithm uses non-overlapping populations and optional elitism and each generation the algorithm creates an entirely new population of individuals. *ii)* a steady-state genetic algorithm that uses overlapping populations. In this variation, it is specified how much of the population should be replaced in each generation. *iii)* The incremental genetic algorithm, in which each generation consists of only one or two children. The incremental GA allow custom replacement methods to define how the new generation should be integrated into the population (for example, a newly generated child could replace its parent, replace a random individual in the population, or replace an individual that is most like it). *iv)* the 'deme' genetic algorithm. This algorithm evolves multiple populations in parallel using a steady-state algorithm. Each generation the algorithm migrates some of the individuals from each population to one of the other populations. The GA implementation used is the one described in [16].

Differential Evolution [12], [10], [7] is a kind of evolutionary algorithm working with real-valued vectors, and it is relatively less popular than GAs. However, it has proven to be very effective in the solution of complex optimization problems. Like GA, evolution strategies and other EC algorithms, it works with populations of individual vectors (real-valued), and evolves them. Many variants have been introduced, but the general scheme is as follows:

ALGORITHM 1. General Differential Evolution Scheme

(0) Initialization: Create a population $\mathcal{P}$ of random vectors in $\Re^n$, and decide upon an objective function $f : \Re^n \rightarrow \Re$ and a strategy $\mathcal{S}$, involving vector differentials.
(1) Choose a target vector from the population $\vec{x}_t \in \mathcal{P}$.
(2) Randomly choose a set of other population vectors $\mathcal{V} = \{\vec{x}_1, \vec{x}_2, \ldots\}$ with a cardinality determined by strategy $\mathcal{S}$.
(3) Apply strategy $\mathcal{S}$ to the set of vectors $\mathcal{V} \cup \{\vec{x}_t\}$ yielding a new vector $\vec{x}_{t'}$.
(4) Add $\vec{x_t}$ or $\vec{x_{t'}}$ to the new population according to the value of the objective function f and the type of problem (minimization or maximization).
(5) Repeat steps 1-4 to form a new population until termination conditions are satisfied.

— End of Algorithm —

In particular, DE was applied using the DE/rand/1/exp strategy which proceeds as follows:

ALGORITHM 2. Strategy $\mathcal{S} = DE/rand/1/exp$

Let F be a scaling factor, $\mathcal{C}_r \in \Re$ be a crossover rate, D be the dimension of the vectors, $\mathcal{P}$ be the current population, $N_p = card(\mathcal{P})$ be the population size, $\vec{v}_i$, $i \in [1, N_p]$ be the vectors of $\mathcal{P}$, $\vec{b}_{\mathcal{P}} \in \mathcal{P}$ be the population's best vector w.r.t. the objective function f and r_1, r_2, r_3 be random numbers in $(0, 1)$ obtained with a uniform random generator function $rnd()$ (the vector elements are $\vec{v}_{ij}$, where $j \in [0, D)$).
Then the transformation of each vector $\vec{v}_i \in \mathcal{P}$ is performed by the following steps:

(1) Initialization: $j = (r_1 \cdot D)$, $L = 0$
(2) $\vec{v}_{ij} = \vec{v}_{r_1j} + F \cdot (\vec{v}_{r_2j} - \vec{v}_{r_3j})$
(3) $j = (j + 1)$ mod D

```
(4) $L = L + 1$
(5) repeating (1) to (4) until($\neg((rnd() < \mathcal{C}_r)\&(L < D))$)
— End of Algorithm —
```

5. EXPERIMENTAL SETTINGS

Two groups of experiments were performed. In the first, the behaviour of 4 genetic algorithms were investigated via a total of $10,240$ constrained and unconstrained (constraints were placed on the evolved model parameters) experiments in a distributed computing environment. Distributed and Grid computing involves coordinating and sharing computing, applications, data, storage, or network resources across dynamic and geographically dispersed organizations. The use of grid technologies is an obvious choice for many data mining tasks within the knowledge discovery process. Condor [15], [13], [14], (`http://www.cs.wisc.edu/condor/`) is a specialized workload management system for compute-intensive jobs in a distributed computing environment, developed at the University of Wisconsin-Madison (UW-Madison). It provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. The distributed experiments in this paper were conducted on a Condor pool of the Institute for Information Technology, National Research Council Canada. The GA experimental settings for the first group are reported in Fig.1.

In the second group of experiments, the behaviour of a differential evolution algorithm strategy was investigated via a total of 10 constrained and unconstrained experiments. The DE experimental settings for this second group are reported in Fig.2.

For both groups of experiments, the raw fitness values were used. That is, the objective function values were used directly as the measure of fitness. In these experiments, root mean squared error (RMSE) was used as the objective function, which is one of many possible measures of difference between the observed raw spectrum values and the theoretical spectrum derived from a chromosome, in the case of the GA algorithms or derived from a vector, in the case of the DE algorithm.

The particular constraints imposed upon both the genetic algorithm chromosomes and the differential evolution vectors, when the respective algorithms were actually constrained, are reported in Fig.3. All algorithms were constrained by the same boundary values, which were determined via the preprocessing that was performed.

6. RESULTS

The application of the preprocessing procedure described in Section-2 produced the results shown in Table-4. A total of 4 peaks were found and their parameters were roughly estimated. The amplitude of the 4-th was too small to be considered as representative of a real peptide and most likely is related with the Yule-Slutzky effect (moving averages may generate an irregular oscillation even if none exists in the original data) [4]. This fourth peak was excluded, thus reducing the set to 3.

The distribution of the fitness values for the fixed-length, constrained genetic algorithm experiments in which the number of generations is 600 or greater is shown in Fig-5 (left). It is left-skewed with the mode around 180, indicating that in general, the algorithm tends to produce results with low RMSE values. However, there is a small secondary mode around 250 suggesting a mixture of two population of results. When the distribution is segregated according to the type of GA this behavior can be appreciated more clearly. Fig-5 (right) shows boxplots of the corresponding distributions for each of the individual GA types. The distribution of the RMSE

Chromosomes were	Constrained and Unconstrained
Genetic algorithm	Simple, SteadyState, Incremental, Deme
Termination condition	number of generations
Optimization direction	minimization
Scaling scheme	linear
Linear scaling multiplier	1.2
Selection scheme	Rank, RouletteWheel, Tournament, Uniform
Score freq. 1	1
Score freq. 2	100
Score freq.	1
Number of generations	200 to 1000 by 200
Crossover probability	0.6 0.7 0.8 0.9
Mutation probability	0.01 0.02 0.04 0.06
Population size	50
Number of populations	5
Percent replacement	0.25
Number replacement	5
Number of best genomes	1
Flush frequency	0
Elitism	yes
Number of offspring	2
Percent migration	0.1
Number migration	1
Random seeds	101 8943 98431 84375

Table 1: Experimental settings for the $10,240$ genetic algorithm (GA) experiments.

Vectors were	Constrained and Unconstrained
Strategy	DE/rand/1/exp
Number of Generations	600
Vector dimension	15
Size of the population	100
Control Constant (F)	0.1 0.2 0.3
Crossing Over factor ($\mathcal{C}_r$)	0.4 0.5 0.6
Random seed	319

Table 2: Experimental settings for the 10 differential evolution (DE) algorithm experiments.

model coeff.	minimum	maximum
a	-0.05	0
b	0	10
c	-1200	0
number of peaks	3	3
1 position	232	247
1 amplitude	1	200
1 σ_{left}	0	15
1 σ_{right}	0	15
2 position	247	327
2 amplitude	1	6500
2 σ_{left}	0	40
2 σ_{right}	0	40
3 position	343	409
3 amplitude	1	2000
3 σ_{left}	0	40
3 σ_{right}	0	40

Table 3: Model boundary constraints for both the four genetic and one differential evolution algorithms. Constraints determined by preprocessing.

x_0	spectrum value	Start	End	A	σ_L	σ_R
241	232.8	232	247	74.2	10.10	6.73
276	6593.9	247	327	6314.2	15.45	27.17
375	1017.9	343	409	678.6	31.79	33.78
501	143.7	499	502	2.8	17.54	8.77

Table 4: Results of the preprocessing procedure applied to the observed spectrum. Start and End refer to the x-values delimiting the peak. The notation for the other parameters is that of Eq-2.

values for the Deme and the Simple GA have narrower ranges than those of the Steady-state and the Incremental algorithms which not only cover a broader range, but have the median and the 25 and 75 quartiles at considerable higher levels. The medians for the Deme and the Simple are both small and comparable, but the $25-75$ interquartile distance is considerably smaller in the case of the Deme, which also has a very small range Table-5. This results indicate that in the context of the present problem, the Deme was clearly the best among the family of genetic algorithms. Individually, the overall best GA result (i.e. the chromosome with minimum RMSE) also corresponds to the Deme.

Alg	MinFitness	MaxFitness		Num. of exp.	
		$ng \geq 600$	unb	$ng \geq 600$	unb
Deme	95.32	218.97	919.46	768	2560
Simple	95.63	391.14	1286.68	768	2560
Steady-state	96.54	687.85	1430.75	768	2560
Incremental	101.42	748.17	1430.75	768	2560

Table 5: Minimum and maximum fitness per type of genetic algorithm, broken up into bounded ($ng >= 600$, where ng is the number of generations) and unbounded (unb) results, with their associated number of experiments.

The comparison between the observed and theoretical spectra according to the best GA results, as well as the estimated background are shown in Fig-6(left). There is a good match (RMSE= 95.32) and the three spectral peaks are identified. They are shown individually with the observed spectrum in Fig-6(right).

In particular, the smallest observed peak was retrieved. Resolving such peaks is usually a challenging task since they are very close to the background.

The results corresponding to the application of Differential Evolution with and without constraining the model components are shown in Table-6.

Exp	F	$\mathcal{C}_f$	ng	Fitness	
				constrained	unconstrained
1	0.2	0.5	600	94.37	69.14
3	0.2	0.6	”	94.43	71.63
4	0.1	0.5	”	94.52	79.60
5	0.3	0.5	”	95.00	75.40
2	0.2	0.4	”	95.17	76.87

Table 6: Fitness for the DE experiments, broken up into runs with constrained and unconstrained vectors. $ng = 600$, where ng is the number of generations. F is the DE weighting factor and $\mathcal{C}_f$ is the crossover constant.

Only 5 experiments for each case were performed, all of them with 600 generations. The controlling parameters F and $\mathcal{C}_f$ do not cover wide ranges, but some combinations involve low values of F with higher of $\mathcal{C}_f$ and conversely. However, the fitness values obtained for the constrained and unconstrained were correspondingly all of the same order, and rather low. If only RMSE (fitness) is considered as model quality measure, then the experiments with unconstrained model parameters seems to have outperformed the constrained counterpart (and also all of the GA results). However, in this case, *i)* negative amplitudes were obtained for some peaks, and *ii)* the first small peak at the initial part of the observed spectrum was not retrieved. Instead, the algorithm combined two gaussians for approximating the second peak (the largest in amplitude). This was due to a numeric effect, since the large values of the largest peak affect considerably the mean sum of squared differences in comparison with the other two peaks. It is known that in some cases apparently single spectral peaks might be in reality composed by two or more individual peaks corresponding to peptides which can not be resolved at the level of precision of the given observations. In this case multi-peak spectral approximation would be a desirable feature of any algorithm, in the sense of suggesting previously unnoticed peptides. However, if not properly constrained, these algorithms may produce physically unrealistic results, like spectral peaks with negative amplitudes or too many close peaks describing a single observed one. These elements indicate on one hand the important of data preprocessing, as well as the need of introducing more elaborate constraint handling and more appropriate model quality measures. In the later case, the use of weighted combinations of different model quality measures as fitness functions, or the formulation of the problem as multi-objective optimization may lead to more appropriate solutions.

The comparison between the observed and theoretical spectra according to the best DE results (experiment 1), with the estimated background are shown in Fig-7(left). There is a good match (RMSE= 94.37) and the three spectral peaks are identified as well. They are shown individually in Fig-7(right) with the observed spectrum.

It is interesting to compare the GA and the DE model results obtained (Table-7). From the point of view of the fitness w.r.t. the observed spectrum, both approaches perform similarly; with DE having a slightly smaller value. However, the best GA variant required 800 generations as opposed to DE, which needed 25% less. Moreover, the best GA model emerged from a total of 3072 experiments as opposed to only 5 in the case of DE, indicating further potential for improvement. Another element to consider is the greater simplicity of DE over GA from the point of view of the number of algorithm controlling parameters. Both evolutionary computation techniques succeeded in the challenging task of resolving peaks which are very close to the background.

7. CONCLUSIONS

Both families of algorithms, GA and DE, were able to correctly identify the 3 peaks existing in the observed spectrum despite their relatively large amplitude differences. The fitness of the theoretical models with respect to the observed data was good. The spreads of the identified peaks were also very accurate and both algorithms successfully identified the background trend. This allows a more accurate determination of the peptide levels.

The experiments indicate the importance of data preprocessing, as well as the need of introducing more elaborate constraint handling and more appropriate model quality measures. In the later case, the use of weighted combinations of different model quality measures as fitness functions, or the formulation of the problem as multi-objective optimization may lead to more appropriate solutions. It was observed that DE obtained its solutions using

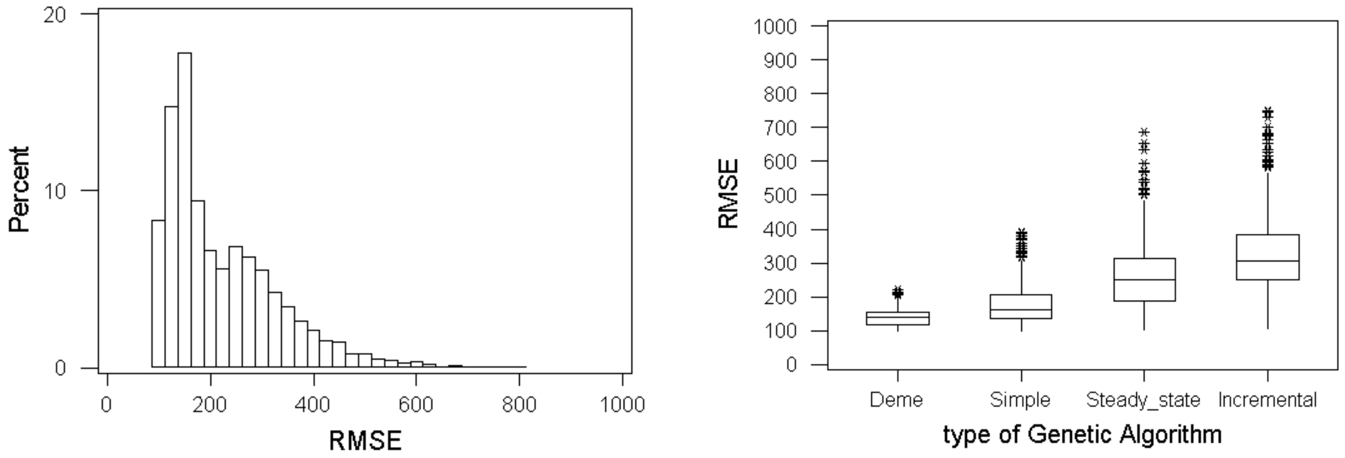

Figure 5: General characteristics of the results obtained with Genetic Algorithms (fixed length vectors, constraints and 600 generations or more). Left: RMSE distribution. Right: RMSE distributions according to the type of genetic algorithm.

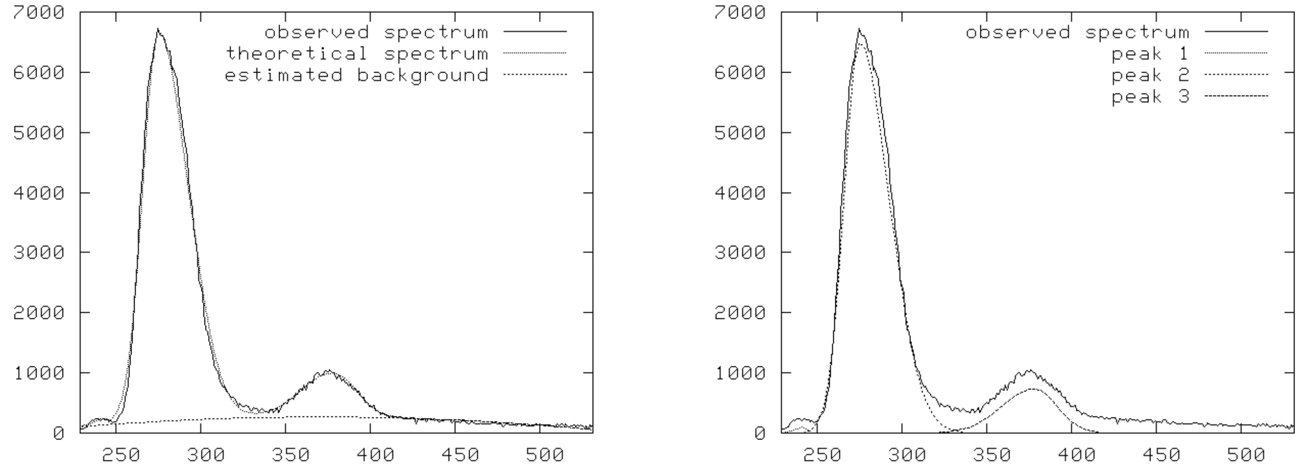

Figure 6: Best results obtained with Genetic Algorithms with fixed length chromosomes and constraints. Left: observed and theoretical spectra. Right: observed spectrum and the individual peaks found by the algorithm.

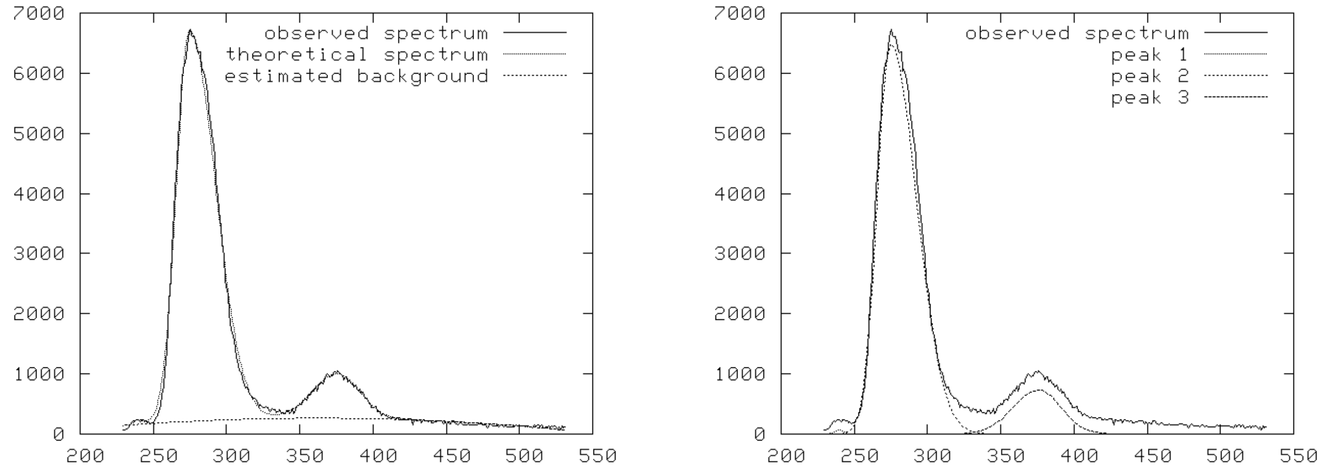

Figure 7: Best results obtained with Differential Evolution with fixed length vectors and constraints. Left: observed and theoretical spectra. Right: observed spectrum and the individual peaks found by the algorithm.

Alg	Exp	Peak 1	Peak 2	Peak 3
GA	8314	$< 243, 109.72, 7.91, 1.88 >$	$< 275, 6497.22, 13.00, 24.71 >$	$< 378, 730.52, 27.10, 18.83 >$
DE	1	$< 239, 80.58, 3.43, 4.69 >$	$< 275, 6498.62, 12.64, 25.00 >$	$< 376, 734.83, 24.40, 21.52 >$
		A,B,C	Fitness	ng
GA	8314	-0.0081, 5.99, -841.72	95.32	800
DE	1	-0.0069, 5.01, -643.83	94.37	600

Table 7: Best models found by each type of algorithm (Alg) in a particular experiment (Exp). Each peak is represented as a tuple $< position, amplitude, \sigma_{left}, \sigma_{right} >$.

fewer computational resources; furthermore, the number of controlling parameters is much smaller than those of GA. It was also noticed that of the four types of GA investigated, the Deme proved be clearly superior.

The ability to accurately quantify the levels of a peptide is very important in order correctly compare its levels in multiple samples. To do this, a method capable of identifying the background noise and also capable of resolving both low abundant and overlapping peptide peaks was desired. The methods described in the current paper are potentially valuable since they address most of the problems. Most of the methods are able to accurately ”mimic” the profiles of the peptides being eluted from High Pressure Liquid Chromatography. They can identify the background to enable accurate quantification. However, only one of the methods enabled resolution of very low abundant peptide. Thus, although the methods need improvement, they are very valuable for accurate quantification.

Thus, the algorithms proved to be capable of accurately identifying low abundant peaks in the presence of background noise and show great potential in quantifying peptide levels in brain tissues from samples with and without stroke. Further experiments are necessary, including

8. ACKNOWLEDGMENTS

The authors would like to thank Jeff Cameron from the University of Waterloo for his help with the numeric experiments, and to Robert Orchard from the Integrated Reasoning Group (National Research Council Canada, Institute for Information Technology) for his constructive criticism of the first draft of this paper.

9. REFERENCES

[1] T. Bäck, D.B.Fogel, and Z. Michalewicz. *Evolutionary Computation 1. Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol and Philadelphia, 2000.

[2] T. Bäck, D.B.Fogel, and Z. Michalewicz. *Evolutionary Computation 2. Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol and Philadelphia, 2000.

[3] T. Bäck, D. B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford Univ. Press, New York, Oxford, 1997.

[4] P. Brockwell and R. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer, 1986.

[5] D. E. Goldberg, K. Deb, and B. Korb. Messy genetic algorithms revisited: Nonuniform size and scale. *Complex Systems*, 4:415–444, 1990.

[6] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3:93–530, 1989.

[7] R. S. K. Price and J. Lampinen. *Differential Evolution : A Practical Approach to Global Optimization*. Natural Computing Series. Springer Verlag, 2005.

[8] T. Masters. *Advanced Algorithms for Neural Networks*. John Wiley & Sons, 1993.

[9] G. Palacky and G. West. Computer processing of airbone electromagnetic data. *Geophysical Prospecting*, 22(3):490–509, 1974.

[10] K. Price. Differential evolution: a fast and simple numerical optimizer. In J. K. J. Y. M. Smith, M. Lee, editor, *1996 Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS*, pages 524–527. IEEE Press, New York, June 1996.

[11] U. Schioler H, Hartmann. Mapping neural networks derived from the parzen window estimator. *Neural Networks*, 5(6):903–909, 1992.

[12] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI, March 1995.

[13] D. Thain and M. Livny. Building reliable clients and servers. In I. Foster and C. Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2003.

[14] D. Thain, T. Tannenbaum, and M. Livny. Condor and the grid. In F. Berman, G. Fox, and T. Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons Inc., December 2002.

[15] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.

[16] T. Wall. *GaLib: A C++ Library of Genetic Algorithm Components*. Mechanical Engineering Dept. MIT, `http://lancet.mit.edu/ga/`, 1996.