1999

Neural Networks in the Analysis of Water-Soluble Sulfonylurea Herbicides Using an LC/MS

Joseph Michael Pompano

College of Humanities and Sciences
Virginia Commonwealth University

This is to certify that the dissertation prepared by Joseph M. Pompano entitled Neural Networks in The Analysis Of Water-Soluble Sulfonylurea Herbicides Using an LC/MS has been approved by his committee as satisfactory completion of the dissertation requirement for the degree of Doctor of Philosophy in Chemistry.

[redacted]

Dr. Sarah C. Rutan, Director of Dissertation

[redacted]

Dr. Fred M. Hawkridge, Committee Member

[redacted]

Dr. Albert T. Sneden, Committee Member

[redacted]

Dr. James Davenport, Committee Member

[redacted]

Dr. Charlene D. Crawley, Committee Member

[redacted]

Dr. Fred M. Hawkridge, Chairman, Department of Chemistry

[redacted]

Dr. Stephen D. Gottfredson, Dean, College of Humanities and Science

[redacted]

Dr. Jack L. Haar, Dean, School of Graduate Studies

September 10, 1999
Date

Neural Networks in the Analysis of Water-Soluble Sulfonylurea Herbicides Using an LC/MS

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

By

Joseph Michael Pompano

M.S. University of Massachusetts, Amherst, Massachusetts, 1982
B.A. University of Connecticut, Storrs, Connecticut, 1970

Director: Dr. Sarah C. Rutan, Ph.D., Professor,
Department of Chemistry

Virginia Commonwealth University
Richmond, Virginia
July 1999

Acknowledgments

I would like to give thanks to the following people.

To my wife, Deborah, for her support and patience throughout my educational experience. To my daughters, Rachel, Rebecca, and Laura for all the joy and love that they have given me throughout their lives.

To Dr. Sarah C. Rutan, my academic advisor, for all the help and encouragement and guidance she has given since I began in 1991.

To my committee members, for their support and guidance. Dr. Fred Hawkridge, Dr. Albert Sneden, Dr. James Davenport, and Dr. Charlene D. Crawley

To Dr. Jerry Bass for all his help with job hunting.

To the members of our group, both past and present: Rusty Poe, Neeta Kulkarni, Min Gui, Huiyan Lu, Zengbiao Li, Wendy Wells, and Kim Johnson.

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Subscripts

i	input layer of neural network
h	hidden layer of neural network
o	output layer of neural network

Symbols

$\mathbf{D}$	Data matrix
$\hat{\mathbf{D}}$	Data matrix reproduced from the truncated eigenvectors and eigenvalues
SVD	Singular value decomposition
$\mathbf{U}_{(m x n)}$	Matrix of abstract row factors, with m rows and n columns used in singular value decomposition
$\mathbf{V}^{T}{}_{(n x n)}$	Matrix of abstract column factors with n rows and n columns used in singular value decomposition
$\mathbf{S}_{(n x n)}$	Matrix of the square roots of eigenvalues used in singular value decomposition
$\mathbf{I}$	Identity matrix
R	Chromatographic resolution
TIC	Total ion chromatogram
α	Level of statistical significance
LC	Liquid chromatography
PB	Particle beam
MS	Mass spectrometry
LC/PB/MS	Liquid chromatograph/particle beam/mass spectrometer
HPLC	High performance liquid chromatography

MSE	Mean squared error
HP	Hewlett Packard
HP-PBM	Hewlett Packard Probability Based Matching
DAD	Diode array detector
m/z	Mass to charge ratio
ANN	Artificial neural network
K	The number of input sample for a neural network
H	The number of hidden nodes in a neural network
$\mathbf{w}_{(HxI)}$	Weight matrix connecting input layer i to hidden layer h
$\mathbf{w}_{(OxH)}$	Weight matrix connecting hidden layer h to output layer o.
$\mathbf{X}_{(IxK)}$	Input matrix to neural network
$\mathbf{Y}_{(KxJ)}$	Output matrix of neural network
$\mathbf{b}_1$	Bias for hidden layer
$\mathbf{b}_2$	Bias for output layer
$\mathbf{F}_1$	Activation function for hidden layer
$\mathbf{F}_2$	Activation function for output layer
ALS	Acetolactate synthase

Abstract

NEURAL NETWORKS IN THE ANALYSIS OF WATER-SOLUBLE

SULFONYLUREA HERBICIDES USING AN LC/MS

Joseph Michael Pompano, Ph.D.

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 1999

Director: Dr. Sarah C. Rutan, Ph.D., Professor, Department of Chemistry

In this research, a hidden node pruning algorithm was developed for an artificial neural network (ANN) that automatically determined a more efficient size of the hidden layer, caused the ANN to re-size itself, and then continued to train using a standard back-propagation algorithm. The hidden-node pruning algorithm was based on determining the number of significant eigenvalues present in the matrix of values produced by the hidden layer, starting with an excessive number of hidden nodes.

Eight sulfonylurea herbicides were used as the target analytes in this study. The ability of an ANN to simplify the sample preparation needed for analysis using a liquid chromatograph/particle beam/mass spectrometer (LC/PB/MS) was evaluated.

The results derived from this research demonstrated that ANNs allow the clean-up procedure to be simplified, while still obtaining reliable identification of the sulfonylurea herbicides in complex matrices such as soil. Specifically, this was accomplished by using retention times from the LC and MS when the herbicides were injected individually in pure forms combined with MS data obtained from extracted samples. This information was used by a trained neural network to identify sulfonylurea herbicides as both individual components and components in a mixture.

Two different neural networks were created. One was trained with a single mass spectrum from each herbicide, resulting in an 8-training-sample network, and one was trained with five mass spectra of each herbicide, resulting in a 40-training-sample network. Both ANNs had 47 input nodes and eight output nodes. Starting with an excess of 20 hidden nodes, the networks resized themselves to contain 6 hidden nodes for the 8-training-sample network and 7 hidden nodes for the 40-training-sample network. An optimum sum-squared error (SSE) goal was determined to be 0.3 for the 8-training-sample network by using a statistical t-test to establish the smallest SSE where the standard error of prediction was not significantly greater than the standard error of calibration. Results demonstrated that the 8-training-sample ANN performed just as well as the 40-training-sample ANN. When compared to the Hewlett-Packard probability-based matching (HP-PBM) library searching system, both neural networks out-performed the HP-PBM system in the identification of unknown mass spectra.

CHAPTER I

INTRODUCTION

1. Overview and Objectives

The main goals of this research were to determine if artificial neural networks could simplify the clean up procedure necessary to identify sulfonylurea herbicides present in a complex matrix (soil) and to develop a hidden node pruning algorithm for the ANN that reduced the number of nodes in the hidden layer in order to improve the efficiency of the ANN.

Sulfonylureas are potent herbicides that inhibit acetolactate synthase, (ALS) which is an enzyme necessary in plants to produce the branched-chain amino acids; valine, leucine, and isoleucine. [1] (See Figure 1)

Sulfonylurea herbicides are particularly difficult to analyze. They are thermally labile, and have low volatility. They are also water-soluble and are used in doses up to 1000 times smaller than other herbicides.[2]. Because of these qualities, liquid chromatography is a better choice for analysis than the more traditional gas chromatography.

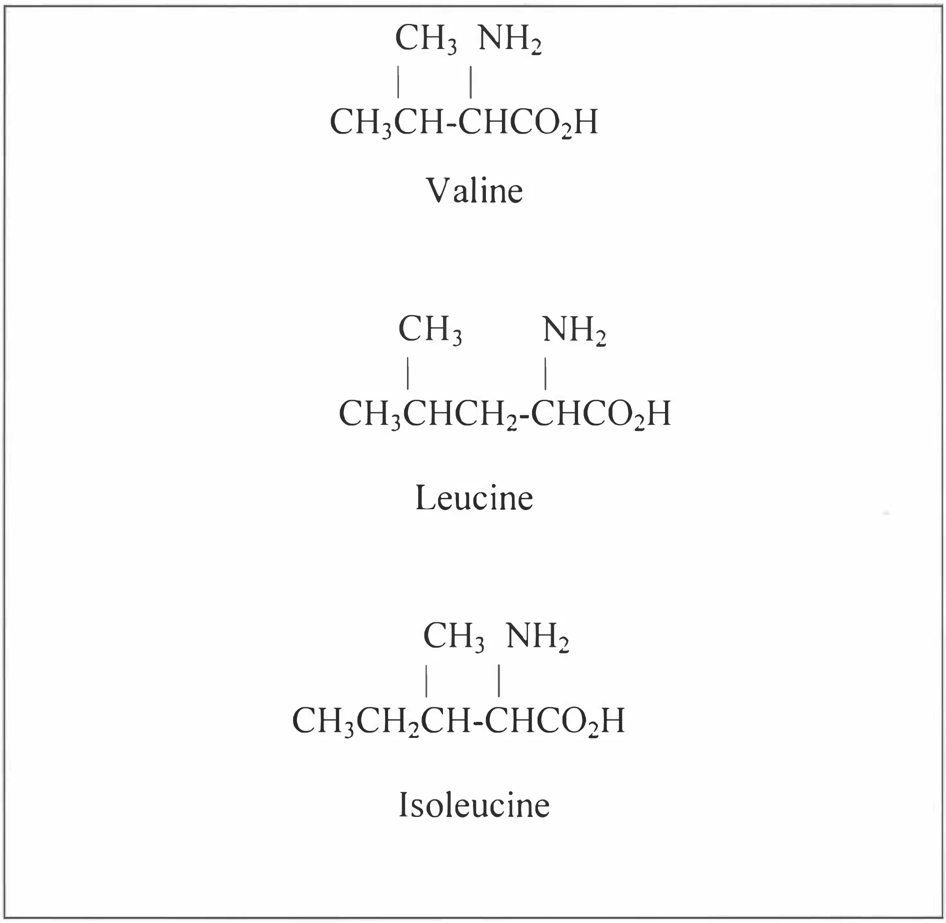


Figure 1. Structure of the Branched-Chain Amino Acids Valine, Leucine, and Isoleucine.

High performance liquid chromatography can be applied to nonvolatile analytes. Applications of HPLC coupled to a mass spectrometer in the quantification and determination of pesticide residues are rapidly increasing. [3]

LC/MS using a particle beam interface allows for the use of electron impact (EI) ionization. EI allows the use of computerized library searches of MS databases of unknown compounds. The particle beam interface can also be used for compounds having a wide range of polarities.

One sub-field of chemometrics, artificial neural networks (ANNs), has seen explosive growth in the past decade. In the area of nonlinear data analysis, ANNs are said to be capable of modeling nonlinear relations. ANNs are mathematical tools that attempt to duplicate some of the functions of biological neurons. ANNs are also very good at handling nonlinear data or outliers in data. [4] They are also useful when a problem cannot be described by a specific model or when there is noise associated with the data, such as the data produced by the particle beam interface.

In ANNs, subgroups of processing elements, neurons, (mathematical functions that sum a series of inputs and operate on that sum), arranged in layers, make independent computations and then pass the result to different subgroups in other layers. This process is repeated until a subgroup of processing elements determines the output. The first subgroup is known as the "input layer," and the last is known as the "output" layer. Any layers between the input and output layers are known as "hidden layers."

One major problem with ANNs is the determination of the appropriate number of nodes in the hidden layer. If there are too many hidden nodes, the ANN has a tendency to model noise in the data. If there are too few hidden nodes, the ANN will not be capable of being trained. The development of a hidden node pruning algorithm is described in the Experimental section and is shown in the Results section to successfully reduce the number of nodes in the hidden layer to an appropriate level. The pruning algorithm is based on the technique of singular value decomposition (SVD), which determines the number of significant eigenvalues in a matrix. SVD was used on the matrix produced by the hidden layer of an ANN initially containing an excessive number of hidden nodes.

Two different ANNs were developed in this research. One used a single mass spectrum of each herbicide, and one used five mass spectra of each herbicide as training samples. Once trained, both ANNs were used to detect the presence of sulfonylurea herbicides extracted from soil. Using a modified version of a technique developed by E.I.duPont de Nemours and Co. in cooperation with the Minnesota Department of Agriculture to extract sulfonylurea herbicides from soil, both ANNs performed at least as well as the Hewlett Packard Probability Based Matching (HP-PBM) system (library searching) in identifying sulfonylurea herbicide from mass spectra, and in some cases, the ANNs were successfully able to identify particular herbicides when the HP-PBM system could not make an identification. However, the ANN trained with the greater number of training samples did not outperform the ANN trained with fewer samples.

CHAPTER II

LITERATURE REVIEW

1. Background

In the United States, pesticide use in agriculture became widespread by the middle of the twentieth century. The term "pesticide" means "pest killer" and includes insecticides, fungicides, herbicides, and rodenticides. The use of pesticides has contributed to a huge increase in the yields for crops grown. [2]

However, the use of pesticides can have a negative aspect as well. In a study of two hundred North Carolina migrant farm workers and 42 non-farm workers, the farm workers applying pesticides had significantly lower erythrocyte cholinesterase levels than non-farm workers. [5] In a study performed on farmers in central Italy, stomach, rectal, and pancreatic cancer were increased among licensed pesticide users with greater than ten years experience. [6]

Pesticides used in crop production are a major source of exposure to pesticide residues. It is the combined responsibility of the Food and Drug Administration (FDA), the Environmental Protection Agency (EPA), and the Food

Safety and Inspection Service of the United States Department of Agriculture (USDA) to ensure that pesticide residues in foods are not present at levels that pose a hazard to people. [7] Legislation enacted by Congress to regulate pesticide exposure and ensure a safe food supply includes the Federal Insecticide, Fungicide, and Rodenticide Act (FIFRA), and the Federal Food, Drug, and Cosmetic Act (FFDCA). [8]

Data on dietary levels of pesticide residues are combined with total diet studies and food consumption surveys to allow the EPA to provide estimates of human exposure to pesticide residues. Therefore, accurate assessments of pesticide residues in a variety of matrices, such as soil and food are essential.

2. Analysis of Pesticides

Methods of analysis of pesticides fall into two general categories, single residue methods and multi-residue methods. Single residue methods are used to identify and quantify a single pesticide or any of its metabolites. Multi-residue methods detect and quantify more than one pesticide present in food at a time. Multi-residue methods may be single-class or multi-class. Single-class multi-residue methods are capable of determining multiple residues all within the same class of pesticides. Multi-class, multi-residue methods are capable of detecting many pesticide residues from a variety of classes. The FDA's multi-residue methods are published in the Pesticide Analytical Manual.[9] The Pesticide Analytical Manual is published as "a repository of the analytical methods used in FDA laboratories to examine food for pesticide residues for regulatory purposes."

The general approach for the analysis of many pesticides consists of sampling the material of interest, extracting the residue, removing any interferences, then identifying or quantifying the pesticide or pesticides of interest using gas or liquid chromatography. The exact nature and complexity of each stage depend on the nature of the pesticide and its sample matrix. Many multi-residue methods depend on extraction of pesticides into an organic solvent for elimination of interferences and maximization of sensitivity for trace level analysis.

3. Sulfonylurea Herbicides

Water-soluble species, such as sulfonylurea herbicides, (Tables 1 and 2) present a particular challenge to the analyst. These herbicides, first introduced in 1982, are used in doses that are 100 to 1000 times smaller than other herbicides, [2] are thermally labile, and have low volatility. Because of these qualities, liquid chromatography is a better choice for analysis than the more traditional gas chromatography.

Table 1. Sulfonylurea Herbicides: Their Trade names, Chemical Formulas, Molecular Masses, Common Names, Melting Points, and Solubilities.

Oust - $C_{15}H_{16}N_4O_5S$ M.W. 364 g/mol Sulfometuron methyl m.p 203-205 °C Solubility - H_2O 0.184 g/L	Classic - $C_{15}H_{15}ClN_4O_6S$ M.W. 414 g/mol Chlorimuron ethyl m.p. 158 °C Solubility - H_2O 1.2 g/L	Ally - $C_{14}H_{15}N_5O_6S$ M.W. 381 g/mol Metsulfuron methyl m.p. 181 °C Solubility - H_2O 0.1 g/L
Glean $C_{12}H_{12}ClN_5O_4S$ M.W. 358 g/mol Chlorsulfuron m.p. 170-173 °C Solubility - H_2O 0.1 g/L	Harmony $C_{12}H_{13}N_5O_6S_2$ M.W. 387 g/mol Thifensulfuron m.p. 176-178 °C Solubility - H_2O 0. 24 g/L	Londax $C_{16}H_{18}N_4O_7S$ M. w. 410 g/mol Bensulfuron methyl m.p. 185-188 °C Solubility - H_2O 0.12 g/L
Express $C_{15}H_{17}N_5O_6S$ M.W. 395 g.mol Tribenuron methyl m.p. 141 °C Solubility - H_2O 0.050 g/L	Accent $C_{15}H_{18}N_6O_6S.H_2O$ M.W. 428 g/mol Nicosulfuron m.p. 141-144 °C Solubility - H_2O 12 g/L	

Table 2. Sulfonylurea Herbicides: Their Trade and Systematic Names

Glean - 2-chloro-N-[[(4-methoxy-6-methyl-1,3,5-triazin-2-yl)amino]carbonyl] benzenesulfonamide
Classic - ethyl 2-[[[[(4-chloro-6-methoxypyrimidin-2-yl)amino]carbonyl]amino]sulfonyl] benzoate
Ally - methyl2-[[[[(4-methoxy-6-methyl-1,3,5-triazin-2-yl)amino]carbonyl] amino]sulfonyl]benzoate
Accent - 3-carboxamide-N,N-dimethyl2[[[[(4,6-dimethoxy pyrimidin-2-yl)amino]carbonyl]amino]sulfonyl] pyridine monohydrate
Titus - 3-ethylsulfonyl 2[[[[(4,6-dimethoxypyrimidin-2-yl)amino]carbonyl]amino]sulfonyl] pyridine
Oust - methyl 2-[[[[(4,6-dimethylpyrimidin-2-yl)-amino]carbonyl]amino] sulfonyl] benzoate
Express - methyl 2-[[[[4-methoxy-6-methyl-1,3,5-triazin-2-yl)methyl]amino]carbonyl] amino]-sulfonyl]benzoate
Harmony- 2-methylcarboxylate, 3-[[[[(4-methoxy-6- methy l-1,3,5-triazin-2-yl)amino]carbonyl]amino]sulfonyl] thiophene
Londax - methyl 2-[[[[[(4,6-dimethoxypyrimidin-2-yl)amino]carbonyl] amino]sulfonyl]methyl]benzoate

The general structure of a sulfonylurea herbicide is illustrated in Figure 2. The greatest herbicidal activity occurs when the aryl segment contains a substituent ortho to the bridge, and the heterocyclic segment is a symmetrical pyrimidine or a symmetrical triazine containing short chain alkyl or alkoxy substituents. [2]

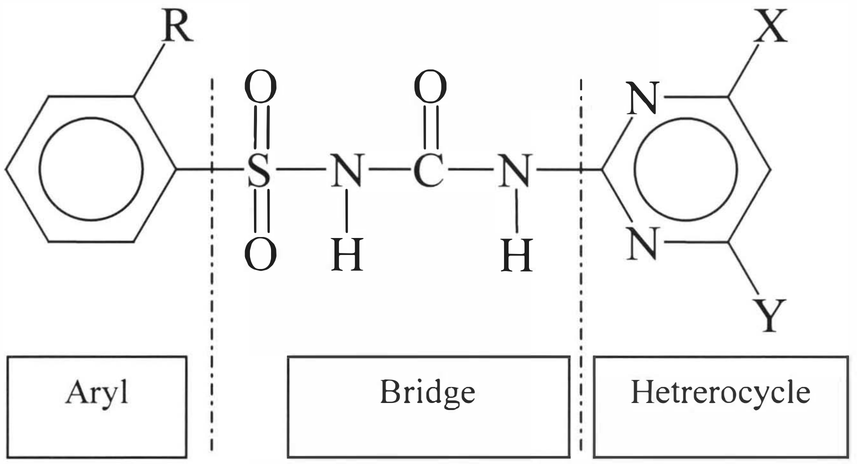


Figure 2. General Structure – Sulfonylurea Herbicides, composed of three segments, an aryl segment, a heterocyclic ring segment, and a bridge connecting the two.

Several researchers have studied sulfonylurea herbicides. From an environmental perspective, Fletcher et al. [10] reported on the effects of low application rates of chlorsulfuron, the active ingredient in the herbicide Glean^{TM}, shown in Figure 3. They demonstrated reduced yield of four plants; canola,

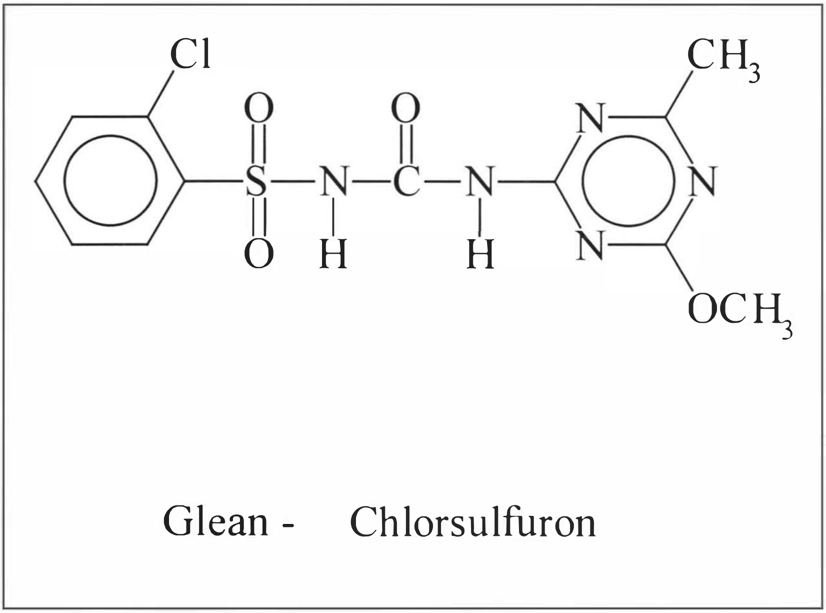


Figure 3. Structure of the sulfonylurea herbicide, Glean

smartweed, soybean, and sunflower by the application of the herbicide in concentrations approximately 1000 times less than the highest exposure recommended by the U.S. Environmental Protection Agency. In a similar study, Fletcher et al. [11] have shown that residues from this sulfonylurea herbicide may severely reduce both crop yields and fruit development on cherry trees.

Analytically, McNally and Wheeler [12] developed a supercritical fluid extraction (SFE) procedure for separation of sulfonylurea herbicides. Coupling supercritical fluid chromatography with supercritical fluid extraction for on-line analysis, the authors successfully extracted 100% of the analytes and their metabolites. The samples were extracted from soil, ground soybean, several wheat matrices, and a cell culture medium containing salts, wheat germ, and the amino acid L-cysteine. Prince and Guinivan [13] used normal phase, high performance liquid chromatography (HPLC) and a photoconductivity detector to determine the residual concentration of chlorimuron ethyl, the active ingredient in the herbicide Classic^{TM}, shown in Figure 4. The LC separation was accomplished using either a 4.6 mm I.D.x 25.0 cm DuPont Zorbax SIL column or a Waters μ-Porasil 3.9 mm I.D. x 30 cm column. The mobile phase was 75% hexane, 12.5% propanol, 12.5% methanol, with 2 mL of acetic acid and 1 mL of water added per liter of mobile phase. The authors presented four methods of extraction and sample cleanup. The minimum detection level was 0.01 ppm and recoveries averaged 90% for samples spiked at the 0.01 to 0.1 ppm levels.

Nilvé and Stebbins [14] studied enrichment of metsulfuron methyl, the active ingredient in Ally^{TM}, shown in Figure 5, and chlorsulfuron (Glean^{TM}) in natural waters using an extraction and back-extraction across an immobilized liquid membrane. They used an automated flow system and monitored the sulfonylureas using UV-detection and obtained a detection limit of 10 ng/L after 500 minutes

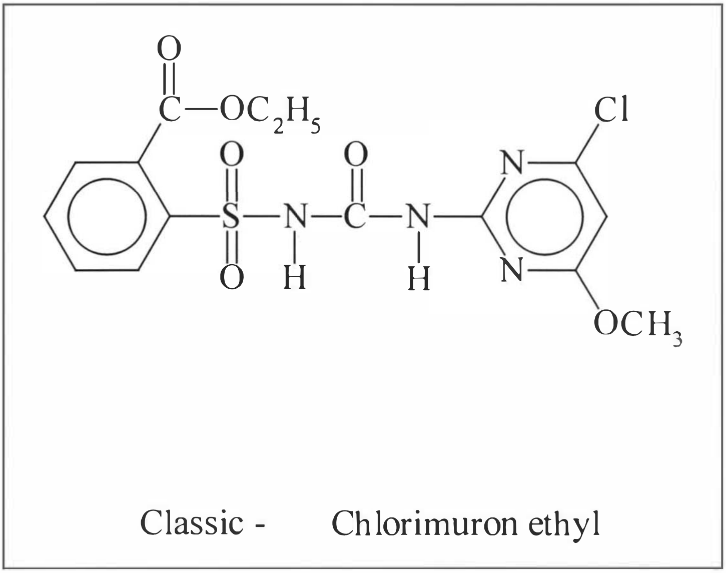


Figure 4. Structure of the sulfonylurea herbicide, Classic.

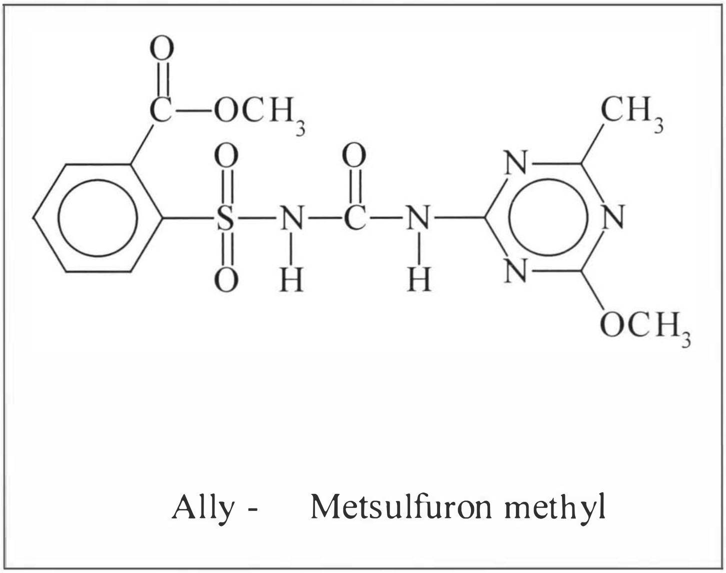
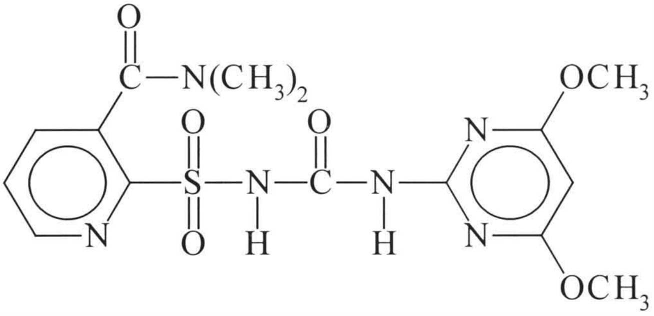


Figure 5. Structure of the sulfonylurea herbicide, Ally

enrichment. Chlorsulfuron has also been determined in soil extracts by enzyme immunoassay [15].

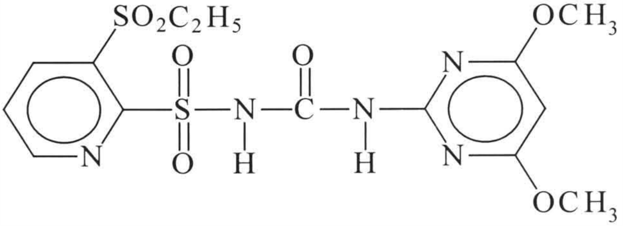
Shalaby et al. [16] used thermospray LC/MS for residue analysis in soil of two sulfonylurea herbicides, nicosulfuron, the active ingredient in the herbicide Accent^{TM}, shown in Figure 6, and rimriduron, the active ingredient in the herbicide Titus^{TM}, shown in Figure 7. The authors also analyzed for a major metabolite of each herbicide, 4,2-(aminosulfonyl)-N,N-dimethyl-3-pyridinecarboxamide, the metabolite of nicosulfuron, and 3,N-(4,6-dimethoxy-2-pyrimidinyl)-N-[3-(ethylsulfonyl)-2-pyridinyl]urea, the metabolite of rimriduron. Their HPLC system was equipped with a Whatman Partisil C-8 column, (25 cm x 4.6 mm I.D.). The gradient mobile phase ranged from 100% 0.1M acetic acid to to (45:55) acetonitrile/0.1M acetic acid. Samples were extracted into an extraction solvent of (80:20) acetonitrile/water. Detection limits of 0.02 ppm were reported.

Shalaby and George [17] reported the optimization of thermospray LC/MS for the analysis of six sulfonylurea herbicides. The analytes included chlorsulfuron (Glean, Figure 3), chlorimuron ethyl (Classic, Figure 4), metsulfuron (Ally, Figure 5), sulfometuron-methyl (Oust, Figure 8), tribenuron methyl (Express, Figure 9), and thifensulfuron (Harmony, Figure 11). The herbicide bensulfuron methyl, (Londax, Figure 10), was also found to elute between Express and Classic using



Accent - Nicosulfuron

Figure 6. Structure of the sulfonylurea herbicide, Accent



Titus - Rimriduron

Figure 7. Structure of the sulfonylurea herbicide, Titus

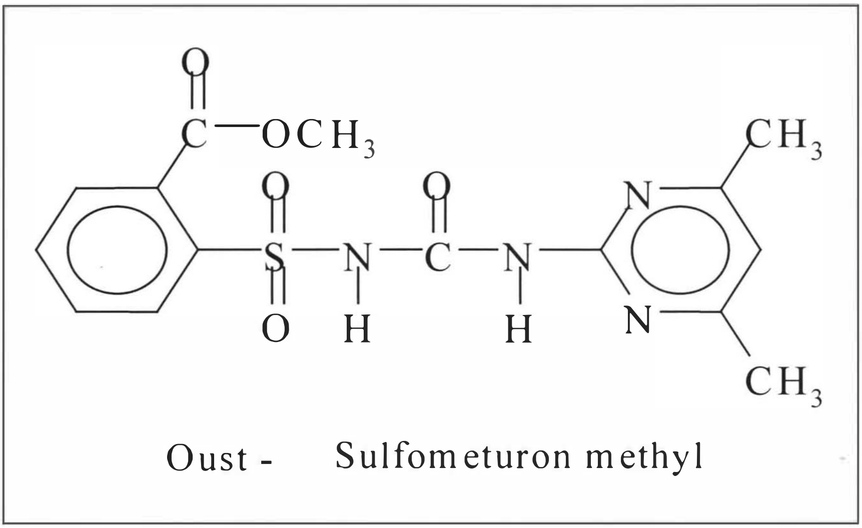


Figure 8. Structure of the sulfonylurea herbicide, Oust

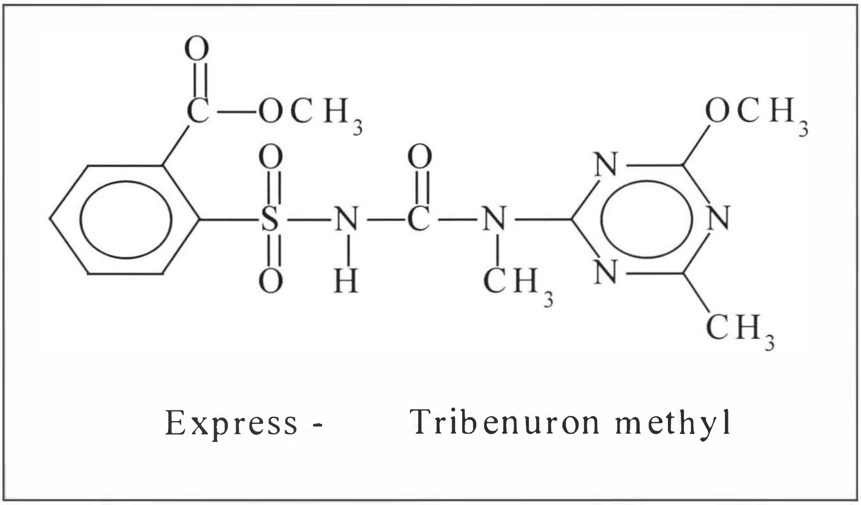


Figure 9. Structure of the sulfonylurea herbicide, Express

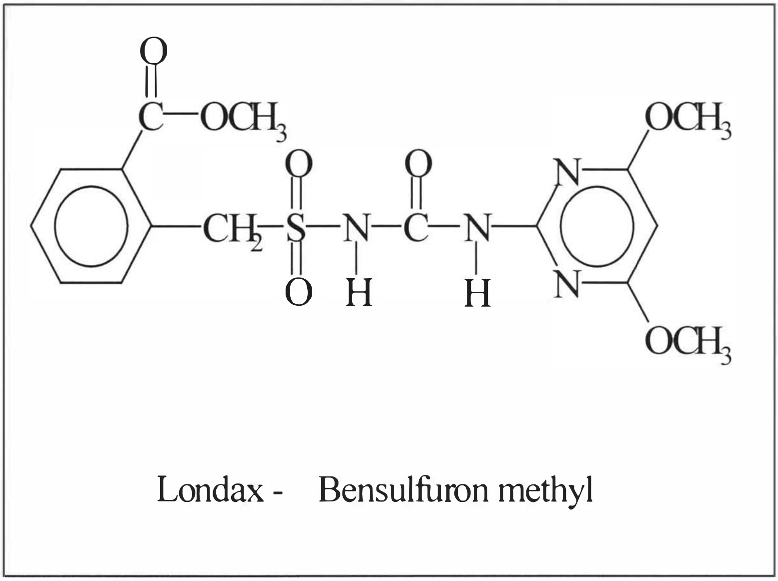


Figure 10. Structure of the sulfonylurea herbicide, Londax

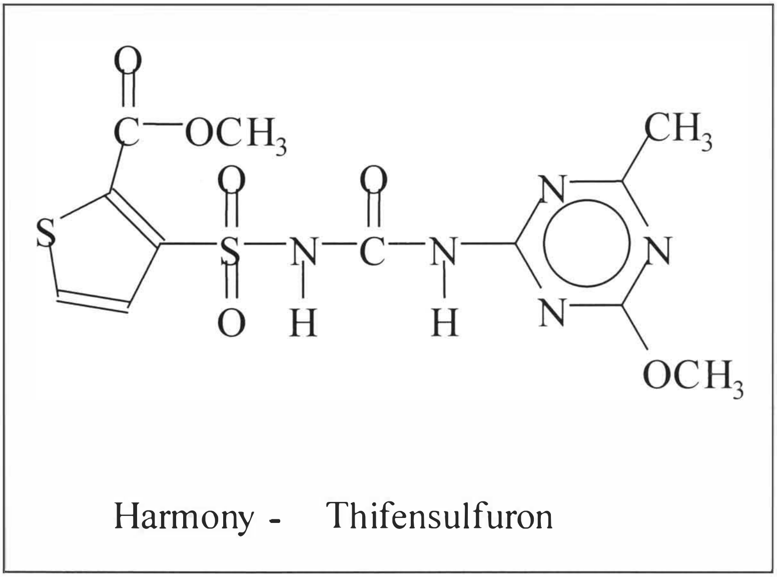


Figure 11. Structure of the sulfonylurea herbicide, Harmony

the same LC conditions. The HPLC system used an Alltech Spherisorb ODS column (the authors did not specify whether the column was ODS I, II, or III). A solution of 0.5 M ammonium acetate was added post-column as a carrier for the particle beam interface. (This will be discussed later) The mobile phase was kept isocratic at (30:70) acetonitrile/0.05 M formic acid for 15 minutes to separate Harmony, Ally, Oust, and Glean. The mobile phase was then ramped to 60% acetonitrile to elute Express and Classic.

The method was applied to the separation and detection of three sulfonylurea residues in wheat grain - chlorsulfuron (Glean), bensulfuron methyl (Londax), and thifensulfuron (Harmony). Ten gram samples were spiked with the herbicides at concentrations of 0.05, 0.20, and 0.50 ppm. Each sample was extracted twice with an acetonitrile/water mixture, vortexed, sonicated, vortexed a second time, and centrifuged. The three herbicides were detected without interferences with an average recovery of 90% for Harmony, 75% for Glean, and 95% for Londax .

Howard and Taylor [18] reported the supercritical fluid extraction of two sulfonylurea herbicides, sulfometuron methyl (Oust) and chlorsulfuron (Glean). Both herbicides were extracted from water onto a 47-mm diameter, solid phase extraction disk, of either C-8 or C-18 bonded silica particles suspended in a web of Teflon fibers. Then, using supercritical 2% methanol-modified CO_2, the herbicides were extracted from the solid phase extraction disks and analyzed by HPLC-UV at a 230-nm detection wavelength. Detection limits were found to be 0.30 ppm for each herbicide.

4. Herbicides and LC/PB/MS

The rapid growth in the use of herbicides began in the 1950s when only 11% of the corn crop and 5% of the cotton crop were treated with herbicides. In 1990, those figures had risen to 95% for corn, soybean, and cotton [19]. While many herbicide determinations involve packed or capillary column gas chromatography (GC) [20], those that are nonvolatile or thermally unstable cannot easily be analyzed by GC. High performance liquid chromatography (HPLC) can be applied to nonvolatile analytes. Applications of HPLC coupled to a mass spectrometer for the quantification and determination of pesticide residues are rapidly increasing.

Liquid chromatography-mass spectrometry, LC/MS, using a thermospray interface (TSP) has been used for the identification of several pesticides and their polar metabolites. Organophosphates, [21] p-nitrophenol, [22] hydroxytriazines, [23] aldicarb sulphoxide and sulphone [24] have all been determined using LC/TSP/MS.

LC/MS using a particle beam interface offers advantages over the thermospray interface, which produces ions in the presence of a relatively high vapor pressure caused by the solvent vapor. [25-28] The thermospray interface operates on the principle that if an ionic solution is nebulized, the ionic charges will not be evenly distributed, causing some droplets to carry a positive charge, while others carry a negative charge, causing dispersion. This dispersion is aided by the addition of an ammonium acetate buffer. [29] While the thermospray interface produces chemical ionization spectra useful for target analysis of known compounds, the particle beam allows for the use of electron impact (EI) ionization. This is useful since EI allows the use of computerized library searches of MS databases for identification of unknown compounds

The particle beam interface can also be used with compounds having a wide range of polarities. [30] [31] LC/PB/MS has been used for the determination of paraquat and diquat in water. [32] Eight chlorinated phenoxy acid herbicides and three ester herbicides were analyzed using both isocratic and gradient LC/PB/MS.[33] However, the sensitivities were not adequate for trace determinations. "Carryover" peaks were detected in some spectra, caused by previously eluted compounds. This was attributed to particles sticking in the interface or ion source from previous runs.

Enhanced ion abundances have been observed with the particle beam interface when ammonium acetate was added to the mobile phase. Beller et al. [34] found that the addition of ammonium acetate enhanced ion abundances of polar analytes in the particle beam by increasing the chromatographic efficiency of compounds and by enhancing the PB carrier process. The authors hypothesized the possible formation of relatively weak molecule-ammonium acetate ion complexes in the mobile phase and in the liquid phase of the spray. The complexes were thought to be held together by hydrogen bonding or weak dipole-dipole interactions. These complexes would be expected to reduce the vaporization of the analytes from the droplets of the spray thus reducing analyte loss in the momentum separator.

Mattina used phenylurea as a carrier for the analysis of three chlorophenylurea herbicides, diuron, linuron, and monuron. The phenylurea was thought to function as a carrier through the PB interface, improving the detection limits and the linearity of the calibration curve. An isocratic mobile phase consisting of acetonitrile-water (68:32) was used with the acetonitrile containing the phenylurea at a concentration of 2.90 ng/μL. The column used was a Waters (300 mm x 2.1 mm I.D.) stainless steel μBondapak10μm C-18 column, protected by a Supelco LC18 guard column. [35]

In a government-sponsored study, eight acid herbicides were examined in an EPA investigation using an LC/PB/MS with disappointing results. [36] Poor precision and detection limits were observed. Response curves over a range of 200 to 2000 ng were non-linear, even with the addition of ammonium acetate to the mobile phase to increase signal strength through the particle beam interface.

The ion degradation products at low concentrations observed were caused by thermal degradation of the sample. Low source temperatures partially moderated the problem but caused excessive peak tailing. Follow-up studies on phenoxy acid herbicides also reported thermal degradation problems. [37] It is exactly these problems encountered with the particle beam interface that makes it a suitable choice for use testing an ANN, since one of the advantages of ANNs is the ability to handle noisy and non-linear data. [38]

Successful determination of three chlorinated acid herbicides with a particle beam interface was accomplished using methane-moderated electron capture negative ion chemical ionization (NCI). [35] Single ion monitoring was used for improved sensitivity. Phenoxyacetic acid was added as a carrier to improve results. Linear response curves were obtained in the 8 to 60 ng range and detection limits in the hundreds of picogram range. NCI was also successfully used on three phenylurea herbicides. [35] Detection limits were in the nanogram range.

Although many methods exist for the analysis of water-soluble pesticide residues, all of them require extensive pretreatment to isolate the analyte from the matrix or the use of a detector that is selective and sensitive to only the analyte of interest. Most of the accepted methods are either labor intensive or require expensive, specialized equipment. No detection methods for sulfonylurea herbicides are given in the Official Methods of Analysis of the Association of Official Analytical Chemists.[39] However, the Pesticide Analytical Manual,[9] (PAM I) does include a "general" section on multi-class, multi-residue methods (MRMs). Method 302 is for non-fatty foods(<2% fat) with high water content (>75%), Method 303 is for non-fatty foods(<2% fat) with low water content (< 75%) and Method 304 is for fatty foods(>2% fat), but sulfonylurea herbicides are not mentioned in PAM I.

None of the current methods of analysis in either book make use of any of the advances in data analysis developed in the field of chemometrics to reduce the difficulty of sample preparation or cleanup.

Much time and effort are involved in screening foods for pesticide residues. Most of the general screening methods extract the analytes into an organic solvent, and thus miss the water-soluble pesticides, like sulfonylurea compounds. When asked what is the biggest problem in pesticide residue analysis, a state worker[40] identified sulfonylurea compounds, along with glyphosate, the active ingredient in Roundup™, another water-soluble pesticide, as causing the most trouble in general screening procedures.

An extraction procedure developed by DuPont in cooperation with the Minnesota Department of Agriculture has been described to extract the sulfonylurea herbicides from soil.[41] This procedure was used in modified form to extract four sulfonylurea herbicides from soil and use their mass spectra with the two neural networks developed in this research.

In a paper presented at an EPA workshop in April 1998, this method describes an approach to extract and quantitate sulfonylurea herbicides using electrospray LC/MS. Quantitation was accomplished using a calibration curve based on the abundance of the molecular ion. Average percent recovery values were listed both by DuPont and the Minnesota Department of Agriculture. Some of their results are listed in Table 3.[42]

5. Artificial Neural Networks

During the last two decades, many new techniques and algorithms have been developed by chemometricians in order to extract useful information from large quantities of data. In the area of nonlinear data analysis, artificial neural networks are said to be capable of modeling nonlinear relations. [43] If an exact solution to a problem cannot be derived mathematically, artificial neural networks may be capable of establishing a connection between the problem and solution. [44]

Artificial neural networks (ANNs) are mathematical tools that attempt to duplicate some of the functions of biological neurons. [45] ANNs are very good at handling nonlinear data or outliers in data. [4, 46] They are also useful when a

Table 3. Average Recoveries of Eight Sulfonylurea Herbicides in High Organic Media Soil. Recovery rates are determined at two levels of fortification, 0.5 ppb and 5 ppb. Two research groups are listed, Dupont and the Minnesota Department of Agriculture (MDA). [42]

Analyte	0.5 ppb Fortification		5 ppb Fortification	
	Average Percent Recovery		Average Percent Recovery	
	DuPont	MDA	DuPont	MDA
Accent	34		50	56
Harmony	77		74	83
Ally	78		76	88
Oust	90		77	87
Titus	50		47	
Glean	73		72	85
Express	34		36	25
Classic	76		76	148

problem cannot be described by a specific model or when there is noise associated with the data. [47]

In ANNs, subgroups of processing elements, neurons, arranged in layers, make independent computations and then pass the result to different subgroups in other layers. This process is repeated until a subgroup of processing elements determines the output. The first subgroup is known as the "input layer," and the last is known as the "output" layer. Any layers between the input and output layers are known as "hidden layers." A simple three-layer, feed-forward, back-propagation network is illustrated in Figure 12. Since the information travels forward from the input layer to the output layer, the ANN is known as "feed-forward." The term "back propagation" will be explained later. In Figure 12, each square represents a neuron and each line represents an interconnection between the neurons. The inputs labeled "bias" provide a constant input of one, but they also have their weights adjusted, as will also be discussed later. [43]

A simple example of the use of a neural network is as follows. If one is trying to determine the boiling point (BP) of a hydrocarbon, based on the number of carbon atoms (C), hydrogen atoms (H), double bonds (DB), and triple bonds (TB) it contained, the neural network would use an input matrix, X_{4xs}, consisting of s column vectors, where each column would contain the four numbers C, H, DB, and TB associated with a particular hydrocarbon. S would be the number of

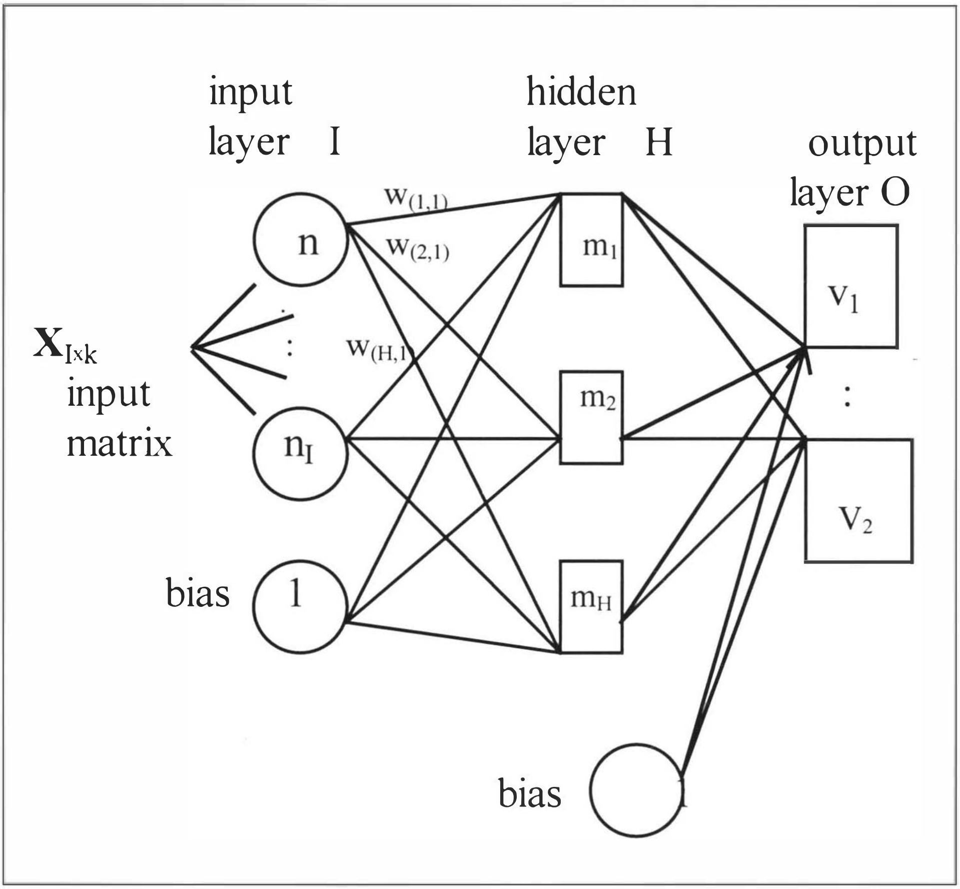


Figure 12. Feed-Forward, Back-Propagation Artificial Neural Network. The ANN in this figure consists of three layers, an input layer, a hidden layer, and an output layer.

hydrocarbons used in the study. The ANN would need to have four neurons in its input layer (one each for the C, H, DB, and TB values). The output layer would contain a single neuron representing the boiling point of the compound associated with the particular set of structural features.

The processing elements in ANNs are known by many different names, including artificial neurons, synthetic neurons, neurons, neurodes, cells, and neuromines. (Figure 13) [43, 48] In an artificial neuron, an input matrix, X_{IxK} , is applied for processing. This input matrix may be raw data, a set of parameters, a series of mass spectra, or the output from other neurons. [49] Some researchers suggest normalizing the data between negative one and one. [50] As can be seen in Figure 12, the input matrix, X_{IxK}, is shown coming into the input layer of the network. The units in the input layer do no calculations, they simply distribute the incoming data along the multiple connections to the hidden layers. Each input to a neuron in a hidden or output layer is multiplied by a weight factor, w, associated with the connection between the neuron receiving the input and the neuron providing the input. These weights are adjusted to make the ANN "learn" its training data. The input layer contains I neurons, the hidden layer contains H neurons, and the output layer contains O neurons. At an individual neuron, all of the weighted inputs are summed, and this sum becomes the argument upon which the activation function operates.

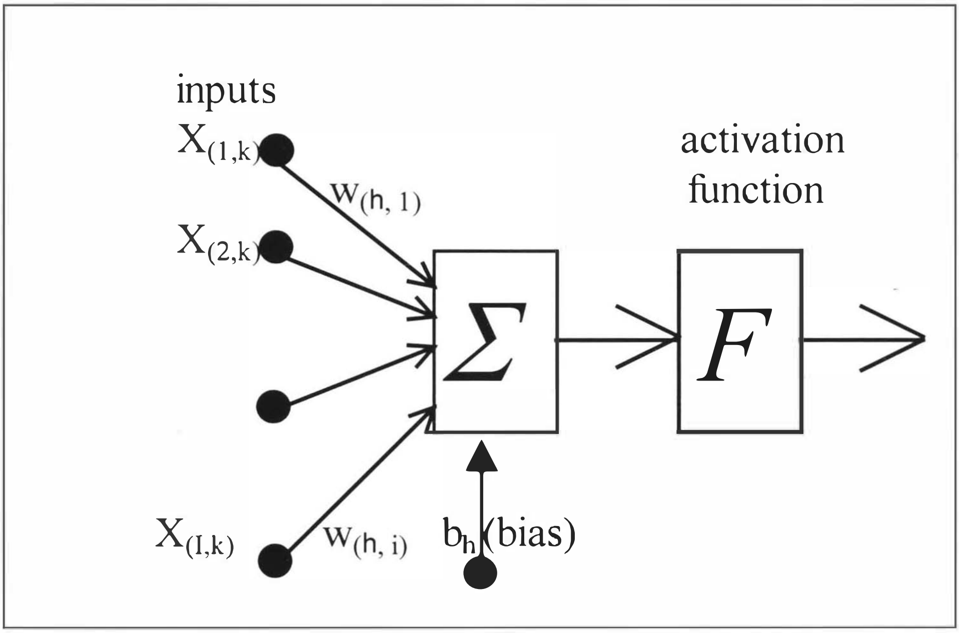


Figure 13 - The Artificial Neuron - A signal, X_{IxK}, at the input layer is connected to hidden neuron h and is multiplied by a weight w_{hxi}, where h is the neuron number and i is the input node number.

5. Activation Functions and Networks:

Activation functions are represented by the boxes in Figures 12 and 13 and are used to determine the final output of the artificial neuron. The activation function is sometimes referred to as a "transfer function". Many types of activation functions exist, but the most common ones are the linear function, the step function, the sigmoid function and the hyperbolic tangent. [51, 52]

The sigmoid function shown in Equation 1 and Figure 14 defines a

$$f(x) = 1/(1 + e^{-x}) \qquad \text{Eq. 1}$$

nonlinear gain for the artificial neuron. This function has the advantage of handling very small or very large values of x and returning an output between zero and one. [53] This function also has a well-defined derivative, which is necessary for the back-propagation training method. The hyperbolic tangent (Equation 2 and Figure 15), also has a well-defined derivative (Equation 3), and has the advantage of ranging between -1 and 1. The hyperbolic tangent proves useful when used with principal component analysis and will be discussed later.

$$\tanh = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad \text{Eq. 2}$$

$$\frac{d}{dx} \tanh x = \operatorname{sech}^2 x \qquad \text{Eq. 3}$$

To understand how a neuron in an ANN performs, consider Figure 16. In this example, a hyperbolic tangent transfer function is being used. There are four inputs and four weights. A bias term is also included. The four inputs are each multiplied by an appropriate weight, summed, and passed to the transfer function, the hyperbolic tangent. The output of the neuron is –0.501.

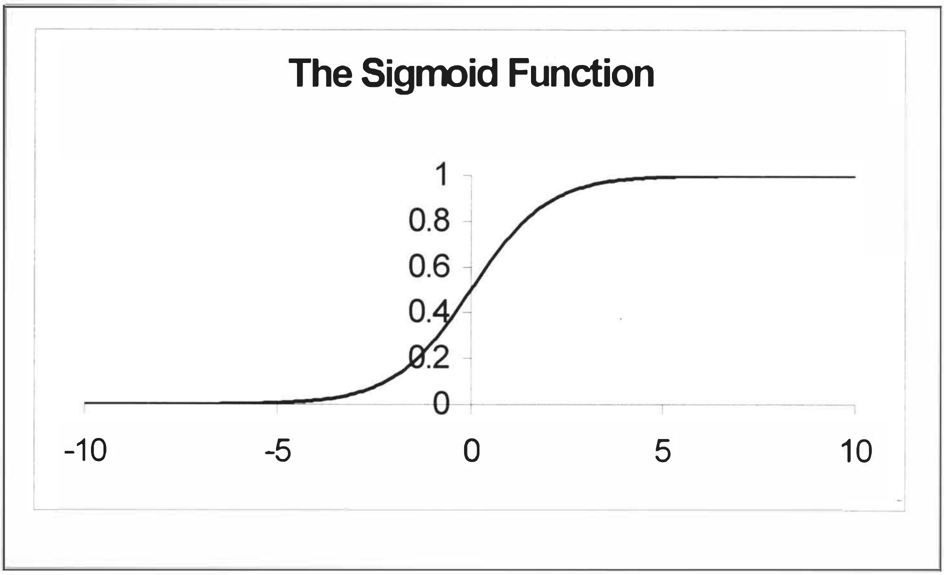


Figure 14. The Sigmoid Function.

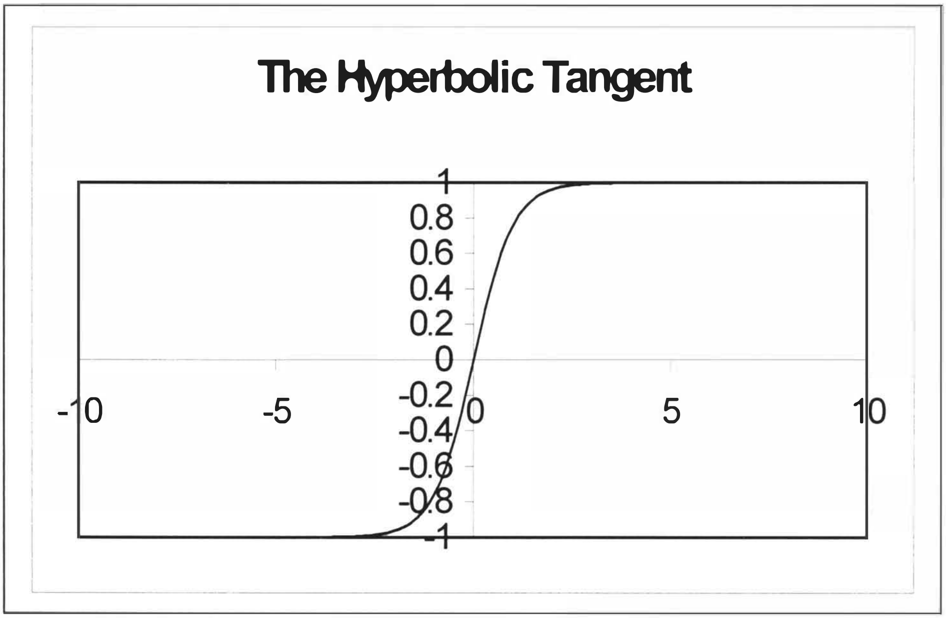


Figure 15. Hyperbolic Tangent.

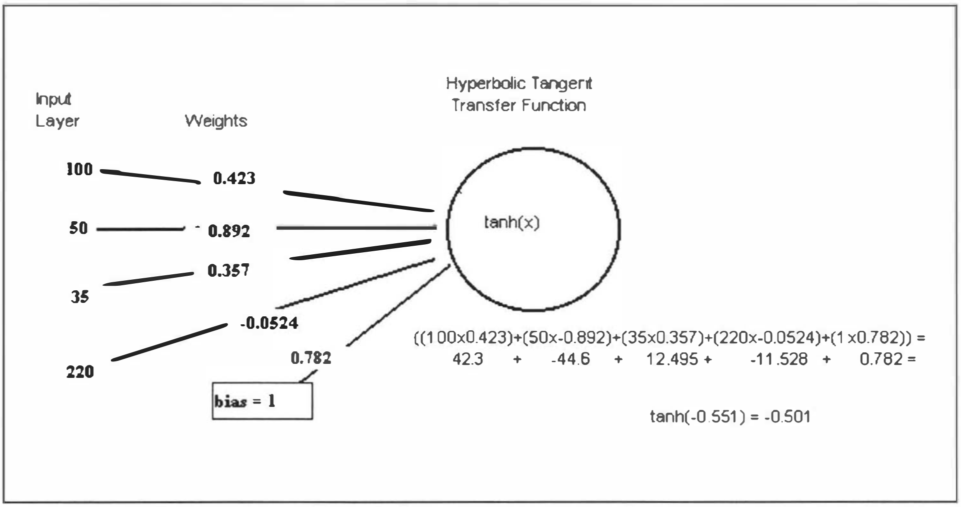


Figure 16. An Example of a Working Neuron.

7. Network Training:

Training an ANN involves adjusting the weights to minimize the difference between the output of the network and the desired output for a set of training data with known or desired output values.[54] This data is called the "training set." One common training method is known as the gradient descent method, and makes use of the derivative of the activation function. The most common error term used is the mean squared error (MSE), which is found by squaring the difference between the actual value and target value for each output neuron, and averaging across them all. Equation 4 represents the definition of this error term, where $t_{(v,s)}$ is the output target value for input vector $\mathbf{X}_{\mathbf{I}\times\mathbf{K}}$, and $o_{(v,s)}$ is the actual output value for input vector $\mathbf{X}_{1xK}$, and O is the number of output neurons. The goal of the training process is to minimize this average sum of squares error over all the training data.

A quantity called the error signal for sample s at output node v, δ_{vs}, is defined as the product of the first derivative of the activation function and the difference between the target value and the value actually produced by the network, $(t_{(v,s)} - o_{(v,s)})$, [55] as shown in Equation 5.

$$\text{MSE} = \frac{1}{O} \sum_{v=1}^{O} (t_{(v,s)} - o_{(v,s)})^2 \qquad \text{Eq. 4}$$

$$\delta_{vs} = f'(x)(\,t_{(v,s)} - o_{(v,s)})) \qquad \text{Eq. 5}$$

If the activation function for the output layer is linear, f(x) = x, then the first derivative is just f '(x) = 1 and the expression for the output layer error signal, δ_{vs}, becomes the difference of t_{vs} and o_{vs}, as shown in Equation 6.

$$\delta_{vs} = (t_{(v,s)} - o_{(v,s)})). \qquad \text{Eq. 6}$$

If the activation function for the output layer is the sigmoid function, (Equation 1, $f(x) = 1/(1 + e^{-x})$ the one used in this research), then the first derivative is calculated as follows. Using the quotient rule, the derivative becomes

$$f^{'}(x) = \frac{0\ -(\ -e^{-x})}{(\ 1 + e^{-x}\ )^{2}} \qquad \text{Eq. 7}$$

By adding and subtracting 1 from the numerator, the derivative becomes

$$f^{'}(x) = \frac{\mathbf{1}\ + e^{-x}\ \mathbf{- 1}}{(\ 1 + e^{-x}\ )^{2}} \qquad \text{Eq. 8}$$

Partitioning the numerator into two parts yields

$$f^{'}(x) = \frac{1\ + e^{-x}}{(\ 1 + e^{-x}\ )^{2}} - \frac{1}{(\ 1 + e^{-x}\ )^{2}} \qquad \text{Eq. 9}$$

Factoring out a $1/(\ 1 + e^{-x}\ )$ term, the expression becomes:

$$f^{'}(x) = \frac{\mathbf{1}}{\mathbf{1 + e^{-x}}} \left[\frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}} \right] \qquad \text{Eq. 10}$$

which equals

$$f^{'}(x) = f(x)\ [\ 1 - f(x)\ ] \qquad \text{Eq. 11}$$

Thus the expression of the output layer error signal becomes

$$\delta_{vs} = f(x)(1 - f(x))(\ t_{(v,s)} - o_{(v,s)}) \qquad \text{Eq. 12}$$

To train the network, this error signal must be propagated backward to adjust the necessary weights. There are two methods by which this can be achieved. The first method is known as *on-line training* or *single pattern training*. Here, the error is propagated backwards to adjust the weights after each training pattern is presented to the network. The second method is to accumulate the δ's for each neuron for the entire training set, adding them and then propagating back the error based on the total δ's. This is known as *batch training* or *epoch training*. The batch method is the default method of training used by Matlab and is used in this research. Equation 13 describes the function used to update the weights.

$$w_{oh}(\text{new}) = w_{oh}(\text{old}) + \eta\delta_{vs}\, o_{(v,s)} \qquad \text{Eq. 13}$$

where η is known as the learning coefficient, which is usually assigned a value between 0 and 1, and w_{oh}(new) is the weight of the interconnection between neuron h in the hidden layer and neuron o in output layer.

Sometimes the network can become caught in a local minimum, especially when the learning coefficient is set very small. In order to avoid this problem, Eberhart has suggested adding a "momentum" term to Equation 13, yielding Equation 14:

$$w_{oh}(\text{new}) = w_{oh}(\text{old}) + \eta\delta_{vs}\, o_{vs} + \alpha[\Delta\, w_{oh}(\text{old})] \qquad \text{Eq. 14}$$

where α is a momentum factor term, and Δw_{oh}(old) stands for the previous weight change. This previous weight change thus acts as a momentum factor that should help the ANN to avoid becoming caught in a local minimum.

As shown in Figure 12, bias neurons can also provide input to the hidden and output layer neurons. The bias neurons always have an output of one and provide a constant offset. The weights for the bias neurons are adjusted exactly like the other weights.

The error signal for hidden node h for input sample s is defined by Rumelhart and McClelland [56] and is shown in equation 15, where $f'(o_{hs})$ is

$$\delta_{hs} = f'(o_{hs})\sum_{v=1}^{O} \delta_{vs} w_{vh} \qquad \text{Eq. 15}$$

the derivative of the activation function, δ_{vs} is the error signal of the output layer neuron (Eq. 12), and w_{vh} is the weight of the interconnection between neuron h in the hidden layer and neuron v in the output layer. Equation 15 propagates the δ_{vs} error term produced in the output layer through the network, hence the name "back-propagation network". Since for the sigmoid function, the derivative is $f(x)(1 - f(x))$, the resulting error signal for the hidden layer neuron is

$$\delta_{hs} = f(o_{hs})(1 - f(o_{hs}))\sum_{v=1}^{O} \delta_{vs} w_{vh} \qquad \text{Eq. 16}$$

The new weights for the interconnections from the input layer to the hidden layer can be calculated from Equation 17, where subscript i takes on values from zero to I for each hidden neuron with I representing the number of input neurons. The I^{th} value is usually assigned to the bias neuron.

$$w_{hi}(new) = w_{hi}(old) + \eta\delta_{hs}o_{hs} + \alpha[\Delta w_{hi}(old)] \qquad \text{Eq. 17}$$

The back propagation method is implemented in the following manner. For each input vector in the training set, the error signal for each output neuron is calculated using Equation 12, and the error signal for each hidden neuron is calculated using Equation 16. The error signals are summed, and after all input vectors have been presented once, the weight adjustments are calculated using Equations 14 and 17. Because the method of updating is the "batch" mode, the δ's in Equations 14 and 17 are the grand totals for each neuron for the entire training set. The values of η and α usually need to be fine-tuned to successfully train the network. Calculations are done iteratively until the error becomes less than a previously specified value or the maximum number of iterations is exceeded. [57]

8. ANNs and Chemistry

Burns and Whitesides [58] have reviewed the use of the back-propagation model, also called the feed-forward neural network (FFN) in applications to chemistry. When applied to spectroscopic data, such as ultraviolet-visible (UV-VIS), mass spectrometry (MS), and near infrared (NIR), etc., ANNs can be used in four ways: pattern recognition, quantification, classification, and transformation into a related function [59]. In this research the ANN is used for pattern recognition.

In another application of ANNs, Smits et al. [60] investigated using "modular" neural network systems for the interpretation of infrared spectra. Instead of using one neural network to interpret the entire IR spectrum, the authors dedicated separate ANNs trained with specific functional groups, such as alcohols or carbonyls. After training, the modular system was compared to a large, "flat" neural network that covered the entire IR spectrum and an actual human expert in the interpretation of IR spectra.

The training of the network was found to greatly depend on the different functional groups present. Five networks were trained to answer three questions. "Does the spectrum represent a compound that contains one, none, or a combination of (1) an alcohol and/or a carbonyl group ('general groups')? (2) any of the specific alcohol groups? (3) any of the specific carbonyl groups?" The "specific" alcohols and carbonyls were listed separately. The five networks consisted of three modular types and two flat types. Results showed that the modular networks outperformed the flat networks. When compared to the human expert, the expert scored best on the first question, the networks scored best on the second question. The scores on the third question were comparable. The networks and the expert often chose the same answers.

Gemperline et al. [61] and Long et al. [62] have used artificial neural networks with back-propagation of errors for developing non-linear calibration models for spectroscopic assays. Figure 17 shows a schematic diagram of the artificial neural network used by these authors. The ANN consisted of three layers of neurons: an input layer, a hidden layer, and an output layer. The input signals, X_i's, were UV-VIS absorbance values, measured at I different wavelengths. The input neurons

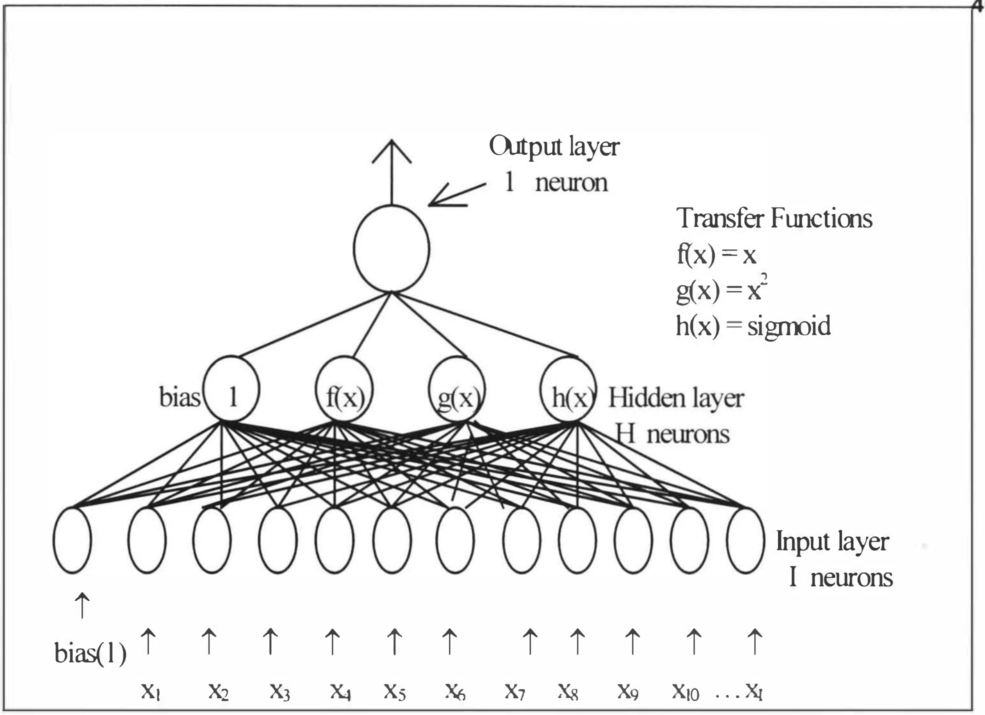


Figure 17. Schematic Diagram of an ANN

had linear outputs and just served to distribute the input signals to the hidden layer through weighted connections. The output layer consisted of one linear neuron. The hidden layer used several different activation functions, including a linear, quadratic, and sigmoid. The authors experimented with different network architectures, varying the number of neurons in the hidden layer and the activation functions. The network architecture that gave the minimum prediction error was retained. Overfitting was guarded against by rejecting models where the relative standard error of prediction (% SEP) for the j-th component was significantly greater than the relative standard error of calibration (% SEC$_j$). For ANNs, the %SEC$_j$ is defined as in Equation 18, where c_{mj} represents the mean

$$\% \text{ SEC}_j = \frac{1}{k} \left(\sum_{i=1}^{n} (\hat{C}_{ij} - c_{ij})^2 \ ) / c_{mj} \right)^{1/2} \text{x}100\% \qquad \text{Eq. 18}$$

concentration of component j: $\hat{C}_{ij}$ represents the estimated concentration of the j-th component for the i-th standard, c_{ij} represents the observed response, and k is the number of calibration standards. The relative standard error of prediction for the j-th component, % SEP$_j$, is defined as in Equation 19, where p represents the

$$\% \text{ SEP}_j = \frac{1}{p} \left(\sum_{i=1}^{p} (\hat{C}_{ij} - c_{ij})^2 \ ) / c_{mj} \right)^{1/2} \text{x } 100\% \qquad \text{Eq. 19}$$

number of unknowns. The neural networks were compared to principal components regression in two trials. Spectra were acquired for 100 lots of hard red spring wheat using wavelengths from 1000 nm to 2600 nm at 1.6 nm intervals. Every eighth data point was used as a sample input, giving 126 data points. The actual protein content of the wheat was determined by the Kjeldahl procedure, and used to train the network. Spectra were also acquired for an aqueous solution of a proprietary pharmaceutical in benzyl alcohol in the range of 270 to 290 nm in increments of 2 nm. The data were known to contain a nonlinear response due to the interaction of the high molecular weight pharmaceutical with the benzyl alcohol. Results showed the neural network performed slightly better than principal components regression (PCR) with the nonlinear data, but PCR outperformed the neural network when the response was linear.

Gemperline et al. [61] and Long et al. [62], also studied the effect of using singular value decomposition (Equation 20) on their calibration spectra matrix, $\mathbf{X}_{\mathbf{IxK}}$, to obtain orthogonal transformations, giving n principal components. $\mathbf{X}_{\mathbf{IxK}}$, was a matrix of K calibration spectra measured at I wavelengths. The n columns of **U** were used as the new transformed orthogonal variables.

$$\mathbf{X}_{\mathbf{IxK}} = \mathbf{U} \bullet \mathbf{S} \bullet \mathbf{V}^{\mathrm{T}} \qquad \text{Eq. 20}$$

For unknown samples, the orthogonal variables were computed using Equation 21, where $\mathbf{X}_{\mathbf{unk}}$ was the unknown sample's measured spectrum, $\mathbf{u}_{\mathbf{unk}}$ was the unknown sample's vector of new orthogonal variables, and **S** and **V** were determined from the SVD of Equation 20.

$$\mathbf{u}_{\mathbf{unk}} = \mathbf{X}_{\mathbf{unk}} \bullet \mathbf{V} \bullet \mathbf{S}^{-1}\, \mathbf{u}_{\mathbf{unk}} \qquad \text{Eq. 21}$$

The authors reported that before obtaining the orthogonal input variables, training the network required from 10,000 to 150,000 iterations per component. When the orthogonal inputs were used, training the network required only 500 to 10,000 iterations.

Eghbaldar et al. used ANNs to identify mass spectra data with a very elaborate input layer. They used 180 intensities of m/z ratios between m/z 40 and m/z 220, five "indicators" depicting general aspects of the spectrum, fifteen "indicators" specific to particular families of compounds, and fifty "autocorrelation sums" which were used to identify homologous series of compounds. All together, they had a total of 456 input values. Their output consisted of logical responses corresponding to three classes: present, absent, or not-classified for 17 structural features, using the numerals 1 to indicate present, and 0 to indicate absent. Results were generally considered successful, but varied with the type of structural feature being determined. The authors concluded that very significant improvements in performance could be achieved with more research in the areas of input data and structural component selection. [63]

9. Factor Analysis

Factor analysis is a multivariate technique used to represent an n-dimensional data matrix in a smaller number of dimensions. [64] This is accomplished by the use of orthogonal factor spaces and transformations that can yield chemically recognizable factors. Two vectors **x** = $(x_1, x_2, \ldots, x_n)$ and **y** =$(y_1, y_2, \ldots , y_n)$ are orthogonal if their inner or "dot" product $(x_1y_1 + x_2y_2 + \ldots + x_ny_n)$ equals zero. Factor analysis usually involves the following five steps: (1)

preparation, (2) reproduction, (3) transformation, (4) combination, and finally (5) prediction. [65]

In the preparation step, the data are selected and possibly pretreated mathematically. Data are first arranged into a matrix. Data should be capable of being modeled as a linear sum of product terms, such as using spectral intensities (Beer's law) in spectroscopy, or abundances in mass spectrometry. Data pretreatment is used when there is a need to standardize the data before factor analysis. Standardization is applied when the columns or rows of the matrix involve measurements made with different units. Data can be mean centered and normalized to unit variance. Logarithms of the data can be used if the logarithms, rather than the raw data itself, are linearly additive.

Reproduction involves two steps, obtaining the abstract factors and determining the number of factors present in the data. In order to calculate the abstract factors, a mathematical procedure known as eigenanalysis, or principal component analysis, is usually carried out. [66] The most common method of eigenanalysis is known as singular value decomposition (SVD). In SVD, the original data matrix, $\mathbf{D}_{mxn}$, is broken down or "decomposed" into the product of three matrices, where each column of **U** is an abstract orthonormal eigenvector

$$\mathbf{D}_{\mathbf{mxn}} = \mathbf{U}_{\mathbf{mxn}}\mathbf{S}_{\mathbf{nxn}}\mathbf{V}^{\mathbf{T}}{}_{\mathbf{nxn}} \qquad \text{Eq. 22}$$

that spans the row space. [67] In this research, the row space can be thought of as an abstract version of the abundances of the individual m/z ratios. Each column of **V** is an abstract orthonormal eigenvector that spans the column space. In this research, the column space can be thought of as an abstract version of the individual m/z ratios of the mass spectra. Vectors are orthonormal if they are orthogonal and have a norm (length) of one. The length ||**x**|| of vector **x** is defined as $\|\mathbf{x}\| = (\mathbf{x}\bullet\mathbf{x})^{1/2} = (x_1^2 + x_2^2 + \ldots + x_n^2)^{1/2}$. Vectors with length one are known as "unit vectors."

$\mathbf{S}_{nxn}$ is a diagonal matrix whose elements are the square roots of the eigenvalues of the row and column eigenvectors. Eigenvalues and eigenvectors are also known as characteristic values and characteristic vectors, respectively. Each eigenvalue represents a part of the total variation present in the data and measures the relative importance of its associated eigenvector.

The data matrix $\mathbf{D}_{mxn}$ can be pre- or post- multiplied by its transpose, $\mathbf{D}^T$ before SVD is performed. The product, $\mathbf{DD}^T$ or $\mathbf{D}^T\mathbf{D}$, is called the correlation matrix if the data matrix it has been normalized, and it is called the covariance matrix if the data matrix has not been normalized. Both matrices $\mathbf{DD}^T{}_{nxn}$ or $\mathbf{D}^T\mathbf{D}_{mxm}$ are square symmetric, which guarantees that all the eigenvalues of Equation 22 are real.

The number of eigenvalues and their associated eigenvectors is usually greater than the number of physically meaningful factors in the data. This is caused by experimental error and/or noise encountered in generating the data. It is most common to eliminate the least important eigenvalues/eigenvectors to reduce the number of factors with which one works.

Several methods are available for selecting the appropriate number of factors. These include methods based on experimental error, such as residual standard deviation (RSD) and root mean square error (RMS), methods determined empirically, such as factor indicator function (IND) and the imbedded error function (IE), and methods based on statistics, such as the "prediction sum of squares" (PRESS) or the reduced eigenvalue indicator function (REV) of Malinowski. [66]

Zhang et al. [50] have developed a hidden node pruning algorithm (HNPA) for feed-forward networks based on SVD. Although the paper contained ambiguous matrix dimensional notation, it did present a method to improve the generalization ability of an ANN. This issue will be addressed later in this section. Using a three-layer network, the authors began training their network with an excess of hidden nodes. After 15,000 to 20,000 iterations, a Matlab version of SVD was performed on the output of the hidden layer, **H**, which produced

$$\mathbf{H}_{\mathbf{sxh}} = \mathbf{U}_{\mathbf{sxk}}\mathbf{S}_{\mathbf{sxh}}\mathbf{V}^{\mathbf{T}}{}_{\mathbf{hxh}} \qquad \text{Eq. 23}$$

Equation 23, where s was the sample number and h was the hidden node number. The number of significant singular values, r, was determined by using a variance ratio, Equation 24.

$$s_{r+1}^{2} / s_{r}^{2} < 0.01 \qquad \text{Eq. 24}$$

The network was then reduced to contain only "r" hidden nodes, and the weights and bias between the input layer and hidden layer were adjusted to produce a hidden layer output $\mathbf{H}^{(2)}$ equal to the first r column of **U.**

$$\mathbf{H}^{(2)} = [\ \mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_r\ ] \qquad \text{Eq. 25}$$

An activation function of $\mathbf{f(x) = 2/(1 +} \boldsymbol{e}^{\mathbf{-x}}\mathbf{) - 1}$ was used to produce an output of the hidden layer in the range of (-1, 1), similar to the values of **U**, produced in the SVD of **H**. Once the network was pruned, training was allowed to continue using a standard back-propagation algorithm. The resulting trained network was then first tested with nonlinear, simulated data and then with nonlinear fluorescence data for a three component system. Compared to the non-pruned network, the Zhang's pruned network achieved better prediction ability and enhanced generalization ability.

CHAPTER III

EXPERIMENTAL

Mass spectra data were obtained using a Hewlett-Packard 5988a MS. A particle beam supplied by CSS Company connected a Hewlett-Packard 1090 LC to the MS. The MS was equipped with a quadrupole mass analyzer and used electron impact as the method of ionization, with an ion source set at 70eV. A high-energy dynode set at 6.0 kV was attached to an electron multiplier detector.

1. Particle Beam Interface

A particle beam interface transfers an analyte from an LC to a MS by three basic processes: aerosol formation, desolvation and momentum separation [25], as shown in Figure 18. The aerosol formation process begins in the nebulizer. The LC effluent is introduced in a fused-silica capillary tube where it is co-axially mixed with helium that shears the liquid eluting from the tube into droplets forming the aerosol. By changing the position of the capillary tube, the shape of the spray and the size of the aerosol droplets that enter the desolvation chamber are altered. Evaporation of the aerosol takes place in the desolvation chamber

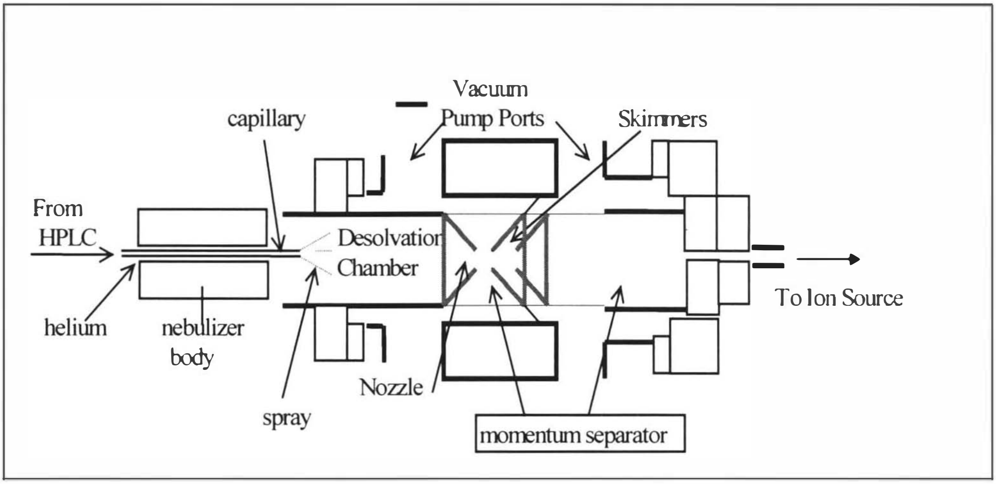


Figure 18. The Particle Beam Interface. The interface consists of a nebulizer, a desolvation chamber, and a momentum separator.

under slight vacuum (0.25 to 0.5 atm) and elevated temperature (50-80°C). As the mobile phase evaporates, the sample molecules combine to form particles.

The momentum separator consists of a nozzle placed at the end of the desolvation chamber and a two-stage skimmer with corresponding vacuum pump. The two-stage momentum separator separates most of the solvent from the particles formed in the desolvation chamber. The solvent and sample particle mixture leave the desolvation chamber and enter the momentum separator at the nozzle where the low-mass and low-momentum solvent vapor particles are removed by the first-stage vacuum pump. The high mass, high momentum sample particles are not removed and are passed through a small opening in the first skimmer. The process is repeated once again in the second-stage vacuum pump and skimmer, leaving a beam of de-solvated analyte particles to enter the mass spectrometer ion source.

2. 2^3 Factorial Experiment

In order to determine which factors, and interactions between factors, were important in determining the output of the LC/PB/MS instrument, a 2^3 factorial design experiment was performed. In a 2^3-factorial design experiment, three variables are investigated at two different levels, usually known as "high" and "low." For quantitative variables, the "high" and "low" terms have their usual numerical meanings. For qualitative variables, the terms can refer to the presence or absence of a variable (e.g., with or without a catalyst), or one of two types of conditions (ex. with or without stirring). [68]

In a 2^3-factorial design experiment, there are 2 x 2 x 2 = 8 possible combinations of factors and levels. Thus, eight experiments are run using all possible combinations of these factors and levels. For the research described in this paper, the desolvation temperature, the helium pressure, and the LC flow rate were investigated. The experiment determined the effect of these factors on the peak area produced by an analyte in the mass spectrometer.

The two levels for each factor were as follows: desolvation temperatures of $50^{O}C$ (low) and $75^{O}C$ (high), helium pressures of 60 psi (low) and 68 psi (high), and LC flow rates of 0.4 mL/min (low) and 0.6 mL/min (high). The concept of coded variables was used, with a plus sign (+) denoting that the factor is at a high level, and a minus sign (-) denoting that the factor is at a low level. The results and statistical analysis from this 2^3 factorial experiment are shown in Table 6 of the Results Chapter.

3. Preparation of Standards for LC Analysis

Stock standards at a concentration of 400 µg/mL were prepared by dissolving 10 mg of each individual herbicide in 25.0 mL of acetonitrile. From the individual stock standards, 100 µg/mL secondary standards of each herbicide were prepared to determine the retention times on the LC. A mobile phase of (75:25) acetonitrile/water, both containing 0.05% acetic acid was used with a flow rate of 0.4 mL/min, unless otherwise noted. The LC was equipped with a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 µm particle size column. Retention times of the individual herbicides were determined (see Table 9, page 76), and based on these retention times, four were chosen to be in a test mixture used to evaluate the ANN's ability to identify the herbicides from their mass spectra. The mixture consisted of four herbicides that had resolutions between 0.87 and 1.4. These resolutions were expected to provide a challenge to the ANN in identifying the individual herbicides from which the MS was taken.

4. Extraction Procedure

Samples were extracted from soil both individually and as a test mixture. A flow chart of the extraction procedure is presented in Figure 19. [42] Soil was obtained by digging in a plot of land near the entrance to Oliver Hall at Virginia Commonwealth University. A 100 mL soil sample was washed with 300 mL of deionized water, decanted, and dried in an oven at 105°C until dry. Five-gram samples of this soil were placed in five 250-mL centrifuge bottles (VWR HDPE wide mouth bottle with cap) and fortified with 1.0 mL of 400 μg/mL sulfonylurea standards (Accent, Londax, Oust, Classic and a mixture containing all four.) A blank five-gram soil sample containing no added herbicide was also processed as a control. After letting air dry for 15 minutes, 100 mL of a (10/90) acetone /0.1M aqueous ammonium carbonate mixture were added to the bottles and the bottles were shaken until the soil was thoroughly wet. The samples were then shaken by hand for 15 minutes and filtered through a Büchner funnel.

Another 100 mL of the acetone/ammonium carbonate mixture were added to each sample, and the bottles were again shaken and filtered. The pH of the filtrate was adjusted to 3.5 with 20% phosphoric acid using a pH meter. The pH

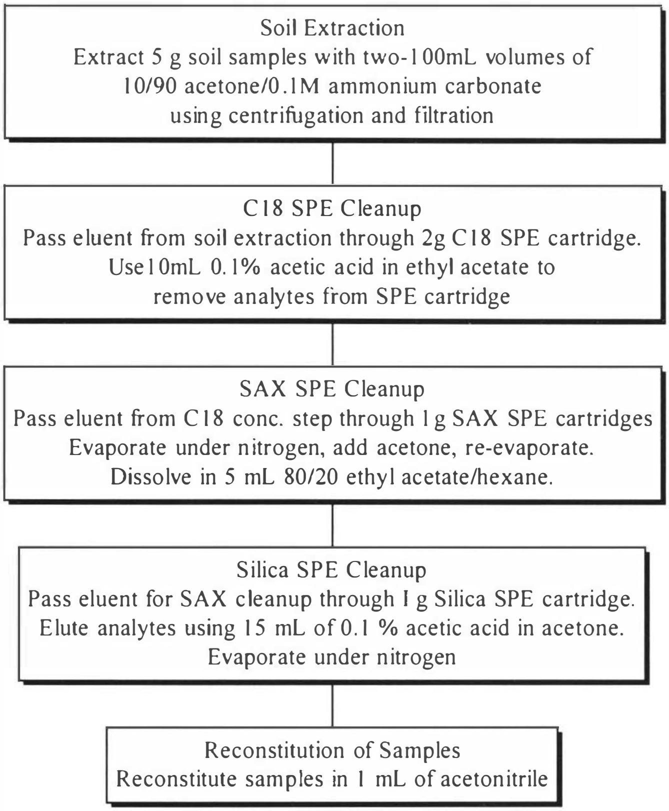

Soil Extraction
Extract 5 g soil samples with two-100mL volumes of
10/90 acetone/0.1M ammonium carbonate
using centrifugation and filtration
C18 SPE Cleanup
Pass eluent from soil extraction through 2g C18 SPE cartridge.
Use10mL 0.1% acetic acid in ethyl acetate to
remove analytes from SPE cartridge
SAX SPE Cleanup
Pass eluent from C18 conc. step through 1g SAX SPE cartridges
Evaporate under nitrogen, add acetone, re-evaporate.
Dissolve in 5 mL 80/20 ethyl acetate/hexane.
Silica SPE Cleanup
Pass eluent for SAX cleanup through 1 g Silica SPE cartridge.
Elute analytes using 15 mL of 0.1 % acetic acid in acetone.
Evaporate under nitrogen
Reconstitution of Samples
Reconstitute samples in 1 mL of acetonitrile

Figure 19. Extraction Procedure Flowchart. The flow chart diagrams the extraction through the use of three different solid-phase extraction cartridges, a C-18 SPE, a SAX-SPE, and a Si-SPE.

meter was initially calibrated at pH's of 4.0 and 7.0. The filtrate of each sample was passed through a 2-gram C-18 SPE cartridge (Varian Mega Bond Elut, 2g, 12mL). The samples were extracted from the C-18 SPE cartridge with 10 mL of 0.1% acetic acid in ethyl acetate and captured in 15 mL conical test tubes (VWR disposable polypropylene centrifuge tube, 15 mL). A long-stemmed pipette was used to remove approximately one mL of water from the bottom of each test tube, since the water had separated into a separate layer. The water layer was discarded and mass spectra were taken of the organic layer at this time.

The eluent from the C-18 cleanup was next passed through a 1-gram SAX SPE cartridge (VWR SAX column type SPE device, anion exchange, 1g, 12 mL) and evaporated under nitrogen. One mL of acetone was added to the dried sample and re-evaporated to remove any residual water. The sample was then dissolved in five mL of 80/20(v/v) ethyl acetate/hexane and mass spectra were again taken. Finally, the samples were passed through a 1-gram silica SPE cartridge (VWR Bakerbond Silica Gel (SiOH) SPE column, 1 g, 6 mL). The analytes were extracted from the Si-SPE cartridges with 15 mL of 0.1% acetic acid in acetone. The samples were evaporated to dryness under nitrogen and reconstituted with 1 mL of acetonitrile, where mass spectra were taken one final time.

5. Hewlett-Packard-Probability Based Matching (HP-PBM) System

All mass spectra were analyzed by the neural network developed in this research and by the Hewlett-Packard-Probability Based Matching (HP-PBM) system. The HP-PBM algorithm operates in the following manner. To each compound in its library, the algorithm analyses the mass spectrum and assigns significance to each peak based on two parameters, U, the uniqueness and A, the abundance. The uniqueness measures how likely a particular m/z ratio occurs in a spectrum. Values range from 0 to 12, with the most common m/z ratios being assigned a 0. The abundance is assigned based on the relative abundance of a m/z peak in a spectrum. Values range from –3 to 5. When a compound is added to library, the PBM algorithm uses U and A to identify the most significant peaks in the spectrum. These significant peaks are then used to generate a condensed reference spectrum that the PBM algorithm uses in its search routine. When an unknown spectrum is analyzed, it is pre-filtered and each of its peaks is assigned significance. The PBM algorithm then uses a reverse search technique in which it verifies that peaks present in the reference spectrum are present in the unknown spectrum. The algorithm generally stores between 15 to 26 peaks in a condensed reference spectrum.

6. Determination of the Neural Network Input Layer

In order to determine the number of input nodes to be used in the neural network, it was first necessary to determine how many individual m/z ratios would be necessary to accurately identify and distinguish the eight sulfonylurea herbicides. To accomplish this, 400 μg/mL samples of each herbicide dissolved in acetonitrile were injected into the LC/MS with an injection loop volume of 20 µL and a mobile phase of 100% acetonitrile. The high concentration of organic solvent was used to ensure good mass spectra. Using 100% acetonitrile and no water in the mobile phase maximizes the throughput from the particle beam because water interferes with the nebulization process.

For each of the eight sulfonylurea herbicides used in this research, a mass spectrum (See Appendix II) was generated after background subtraction using the HP Chemstation™ software. From these mass spectra, mass/charge (m/z) ratios that were characteristic for each herbicide were determined. Using the "Data Analysis" module in the HP Chemstation™ software and extracting individual ion chromatograms with m/z ratios ranging from 100 to 250 accomplished this. An extracted ion chromatogram reveals the contribution of a particular mass (m/z ratio) to the total ion chromatogram.

The resulting 47 m/z ratios are shown in Table 4. The abundances of these 47 selected m/z ratios were the values passed to the 47 nodes of the input layer for the neural network developed in this research. Appendix II contains the mass spectra and identification of some fragment ions of the sulfonylurea herbicides. Most of the fragment ions that were identified were formed by cleavage along the sulfonylurea bridge α to one of the nitrogen atoms. Another group of ions formed by this same cleavage, but also lost one methoxy-group on the pyrimidine or triazine group.

Table 4. Selected m/z Ratios of the Sulfonylurea Herbicides

Accent	Ally	Classic	Express	Glean	Harmony	Londax	Oust
123	120	104	124	108	126	119	105
154	136	120	125	110	136	123	106
180	166	121	153	111	140	149	120
181	184	155	154	113	151	180	121
	199	156	199	127	157	212	123
	210	157	210	128	165		149
		158	241	136	166		184
		159		140	190		199
		184		166	205		
		185		175	221		
		186		177			
		187					
		212					
		234					

7. The Hidden Node Pruning Algorithm

The three-layer feed-forward neural network diagramed in Figure 12 can be mathematically formulated as follows:

$$\mathbf{Y}_{\mathbf{o \times k}} = \mathrm{F}_2[\mathbf{w}_{\mathbf{o \times h}}\{\mathrm{F}_1(\mathbf{w}_{\mathbf{h \times i}}\mathbf{X}_{\mathbf{i \times k}} + \mathbf{b1})\} + \mathbf{b2}] \qquad \text{Eq. 26}$$

where $\mathbf{Y}_{\mathbf{o \times k}}$ is the output matrix that is produced by the network, with O rows, one for each node in the output layer, and K columns, one of each sample presented in the input matrix; $\mathbf{X}_{\mathbf{i \times k}}$ is the input matrix, with I rows, one for each data point in the sample (47 in this research), and K columns, one of each sample presented to the network. Both 8 samples, representing the eight herbicides, X_{47x8} and 40 samples, where each herbicide was represented 5 times, X_{47x40}, were used; F_1 and F_2 are the activation (transfer) functions for the hidden and output nodes, respectively; $\mathbf{w}_{\mathbf{hxi}}$ is the weight matrix connecting the input layer, I, to the hidden layer, H; $\mathbf{w}_{\mathbf{oxh}}$ is the weight matrix connecting the hidden layer, H, to the output layer, O; and **b1** and **b2** are the biases for the hidden node and output nodes, respectively.

Specifically, the result produced at each node in the output layer of the network described in this research can be described by Equation 27.

$$\mathbf{y}_{\mathbf{j}} = \mathrm{sigmoid}\left(\sum_{h=1}^{20} \mathbf{w2}_{\mathbf{jh}} * \tanh\left[\sum_{i=1}^{47} \mathbf{w1}_{\mathbf{hi}} * \mathbf{x}_{\mathbf{i}} + \mathbf{b1}_{\mathbf{h}}\right] + \mathbf{b2}_{\mathbf{j}}\right) \qquad \text{Eq. 27}$$

The inner summation from 1 to 47 represents the 47 abundances, $\mathbf{x}_{\mathbf{i}}$, from the mass spectrum of each herbicide. These abundances are multiplied by a weight connecting input layer I to hidden layer H, $\mathbf{w1}_{\mathbf{hi}}$. To this product, the bias term, $\mathbf{b1_h}$**,** is added. This value is then acted upon by the hyperbolic tangent function, tanh. The resulting value is then multiplied by a weight connecting the hidden layer to the output layer, $\mathbf{w2_{jh}}$**.** A second bias term is added, $\mathbf{b2_j}$ and the result is passed to the sigmoid function. The output of this function is the output of the output node.

A hidden-node pruning algorithm was developed based on the ideas presented in the Zhang [50] paper. However, as mentioned in the literature review section, several inconsistencies in the matrix dimensions exist in this paper. For example, the dimensions of the hidden layer are given as (KxH), where K is the number of input samples and H is the number of hidden nodes. However, the actual dimensions of the hidden layer as produced by Matlab (which was used by Zhang) are (HxK).

This paper also differed from most other papers in the author's use of subscripts on the weights. It is a very common practice in neural network papers to reverse the order of the subscripts from the input and hidden layers. That is, a weight connecting a node in the input layer, I, to a node in the hidden layer, H, is usually designated as "w_{hi}". However, in the Zhang paper, this same weight is designated "$w_{ih.}$" In fact, throughout the entire paper, every matrix multiplication is backward from the standard notation used in Matlab.

The algorithm developed in this work was constructed with functions in the Matlab Neural Network Toolbox. A matrix, **X**, consisting of the 47 m/z ratios for each of the eight sulfonylurea herbicides (or 40 when using five versions of each herbicide) was used as the input to a three-layer feed-forward neural network. The network was initially designed with excessive hidden nodes, and then allowed to run for 1500 iterations. This allowed the network to develop weights for the input to hidden layers and the hidden to output layers. The 1500 iteration mark was chosen because experiments allowing the iterations to continue until the target goal was achieved with excess hidden nodes usually took between 9000 and 16000 iterations.

Training the network was accomplished in the following manner. The input matrix, $\mathbf{X}_{47\mathrm{x}8}$ or $\mathbf{X}_{47\mathrm{x}40}$, was initially pre-multiplied by weight matrix, $\mathbf{w1}_{20\mathrm{x}47}$, with dimensions 20x47, where 20 was an arbitrarily chosen number of excessive hidden nodes. These weights were initially randomly generated. To the product of these two matrices, a bias term, $\mathbf{b1}_{20\mathrm{x}1}$, was added. This bias, a column matrix, was added to each column in the product matrix. This sum was then passed to the first activation function, F_1, the hyperbolic tangent function. The output of this function, which is also the output of the hidden layer, $\mathbf{H}_{20\mathrm{x}8}$ **or** $\mathbf{H}_{20\mathrm{x}40}$, is a matrix with dimensions 20x8 or 20x40. This matrix was then pre-multiplied by the second weight matrix $\mathbf{w2}_{8\mathrm{x}20}$, forming a matrix with dimensions 8x8 or 8x40. To this matrix a second bias term, $\mathbf{b2}_{8x1}$ was added, again to each column of the matrix. The new matrix formed was then passed to the second activation function, the sigmoid function. This produced the output matrix, $\mathbf{Y}_{8x8}$ or $\mathbf{Y}_{8x40}$. The production of the output matrix completes one iteration or epoch. Determining if the output goal was achieved was accomplished as follows. The output matrix produced after one iteration was compared to a target matrix consisting of the values 0.9000 and 0.1000. The "0.9000" corresponds to the herbicide being present, and the "0.1000" corresponds to the herbicide being absent, as shown in Table 5 for the $\mathbf{Y}_{8x8}$ version. Each column of Table 5 is associated with one sulfonylurea herbicide. Using the values 0.9 and 0.1 instead of 1 and 0 avoids the problem of having the output transfer function approach its limits during training.[62] Training the network with values 1.0 and 0, instead of 0.9 and 0.1 resulted in the network being over-trained, less generalized, and less able to identify the herbicides.

After one iteration through the network, the output matrix, $\mathbf{Y}_{8x8}$or $\mathbf{Y}_{8x40}$, and the target matrix were compared, and a sum-of-squares error term was computed. This error term was compared to an error goal, 0.3 for the 8-training-sample network and 1.5 for the 40-training-sample network. If the error was greater than the goal, the weights were adjusted using the equations described in the previous section, and another iteration was performed. The iterations continued until the

Table 5 - Target Matrix Used for Training

Accent	Ally	Classic	Express	Glean	Harmony	Londax	Oust
0.9000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
0.1000	**0.9000**	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
0.1000	0.1000	**0.9000**	0.1000	0.1000	0.1000	0.1000	0.1000
0.1000	0.1000	0.1000	**0.9000**	0.1000	0.1000	0.1000	0.1000
0.1000	0.1000	0.1000	0.1000	**0.9000**	0.1000	0.1000	0.1000
0.1000	0.1000	0.1000	0.1000	0.1000	**0.9000**	0.1000	0.1000
0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	**0.9000**	0.1000
0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	**0.9000**

error goal was met or the maximum number of iterations was exceeded.

Since the output of the hidden layer was produced with an excessive number of hidden nodes, its contents are expected to be linearly dependent and not of full rank. The "rank" of a matrix is the maximum number of linearly independent columns, or equivalently, rows, present in the matrix. If the network could produce a hidden layer output matrix that was of full rank, then it could be assumed that the number of hidden nodes in the hidden layer was optimum for the given inputs and outputs. In order to have the hidden layer produce such a full rank matrix, a hidden node pruning algorithm was performed in the following manner.

The output of the hidden layer can be represented by Equation 28.

$$\mathbf{H}_{\mathbf{HxK}} = \mathrm{F}_1(\mathbf{w1}_{(\mathbf{hxi})}\mathbf{X}_{(\mathbf{ixk})} + \mathbf{b1}_{(\mathbf{hx1})}) \qquad \text{Eq. 28}$$

Or, using the dimensions associated with the 8-training-sample network,

$$\mathbf{H}_{\mathbf{20x8}} = \mathrm{F}_1(\mathbf{w1}_{(\mathbf{20x47})}\mathbf{X}_{(\mathbf{47x8})} + \mathbf{b1}_{(\mathbf{20x1})}) \qquad \text{Eq. 29}$$

This hidden layer output matrix has the dimensions 20 x 8, where the 20 comes from the number of nodes in the hidden layer, and the 8 comes from the number of samples (columns) in the input matrix. The 47 and 8 in the input matrix, $\mathbf{X}_{\mathbf{47x8}}$ come from the 47 mass to charge ratios, m/z, in the mass spectra of the 8 sample herbicides. Applying the singular value decomposition algorithm to this hidden layer output matrix produces the following results:

$$(\mathbf{H}_{20x8}) = \mathbf{U}_{20x8}\bullet\mathbf{S}_{8x8}\bullet\mathbf{V}^{T}_{8x8} \qquad \text{Eq. 30}$$

where **U** and **V** are both orthogonal, and **S** contains the singular values, as discussed previously.

The number of dominant singular values in **S,** n, is indicative of the true rank of **H.** If one were to force the hidden layer of the network to output an orthogonal matrix of dimension (n x 8**),** using only n neurons ($n < 8$) in the hidden layer, then this smaller network would be expected to be more efficient, and be more likely to train without over-fitting the data. Since $\mathbf{V}^{T}$ is an orthogonal matrix, it would be convenient to be able to force the network to produce, as its hidden layer output, a matrix consisting of the first n rows of $\mathbf{V}^{T}$**.** As can be seen in Equation 30, a matrix consisting of the first n rows of $\mathbf{V}^{T}$ has dimensions (nx8). These dimensions fit into the matrix algebra of the algorithm.

In order to determine the estimated rank, n, of the hidden layer, $\mathbf{H}_{20x8}$, the matrices **S** and **H** from Equation 30 are passed to the Matlab program "hidnode" (see Appendix I). This program performs two rank estimation procedures. A percent variance test, where the number of factors kept is based on the ratio of the sum of n eigenvalues to the total of all eigenvalues, and a statistical F-test (program vlftest, Appendix I) are used to estimate the rank of H [69]. The algorithm uses the maximum of these two values as the estimated rank of H. The rank of **H,** symbolized by 'n', is then used to replace $\mathbf{H}_{(20x8)}$ with a matrix consisting of the first n rows of $\mathbf{V}^{T}{}_{(8x8)}$. Another rank estimation procedure was originally investigated involving the Malinowski Indicator function, [64] but was dropped from the algorithm because it overestimated the rank of H when using the 40-training sample network.

Based on the work of Zhang, [50] it was recommended that the weights and biases be reset after the number of hidden nodes was reduced. Equations 31 through 36 accomplish this task, but no improvement in performance was observed in the reduced hidden node network whether the weights were reset with these equations or randomly set by Matlab using the standard back propagation algorithm.

In order to have the hidden layer produce an output equivalent to the first n rows of $\mathbf{V}^{T}{}_{8x8}$, the weights **w1** and bias **b1** need to be reset in the following manner. The hidden layer output matrix was produce by Equation 31:

$$\mathbf{H}_{\mathbf{hxk}} = F_1(\mathbf{w}_{(\mathbf{hxi})}\mathbf{X}_{(\mathbf{ixk})} + \mathbf{b1}) \qquad \text{Eq.31}$$

Substituting in the $\mathbf{V}^{T}{}_{\mathbf{nx8}}$ matrix and the hyperbolic tangent for F_1 yields:

$$\mathbf{V}^{T}{}_{\mathbf{nx8}} = \tanh(\mathbf{w}_{(\mathbf{hxi})}\mathbf{X}_{(\mathbf{ixk})} + \mathbf{b1}) \qquad \text{Eq.32}$$

Applying the inverse hyperbolic tangent, $\tanh^{-1}$, to both sides of Equation 32 gives:

$$\mathbf{tanh}^{-1}(\mathbf{V}^{T}{}_{(\mathbf{nx8})}) = \mathbf{w}_{(\mathbf{hxi})}\mathbf{X}_{(\mathbf{ixk})} + \mathbf{b1} \qquad \text{Eq.33}$$

Initially, the value of the bias was subtracted from $\tanh^{-1}(V^T_{(nx8)})$ by adjusting the size of **b1** to match the dimensions of $\mathbf{V^T_{(nx8)}}$, before subtracting column-wise to yield Equation 34.

$$\mathbf{\tanh^{-1}(V^T_{(nx8)}) - b1 = w_{(hxi)}X_{(ixk)}} \qquad \text{Eq. 34}$$

However, this technique resulted in weights with imaginary values, and so was rejected. Finally, the "pseudoinverse" of the $\mathbf{X_{(ixk)}}$ matrix ($\mathbf{[X_{(ixk)}{}^T \bullet X_{(ixk)}]^{-1} X_{(ixk)}{}^T}$) was right multiplied on both sides, giving:

$$\mathbf{(\tanh^{-1}(V^T_{(nx8)}))\bullet ([X^T_{(ixk)} \bullet X_{(ixk)}]^{-1} X^T_{(ixk)}) = w_{(hxi)}} \qquad \text{Eq.35}$$

Thus the new weight matrix from the input layer to the hidden layer becomes a (nx47) matrix.

The weight matrix from the hidden layer to the output layer, $\mathbf{w_{(oxh)}}$ also needed to be changed from an (o x h) to an (o x n) matrix. This could be accomplished by the following equation:

$$\mathbf{W2_{(o\,x\,n)} = W2_{(oxh)}\bullet H_{(hxk)} * [V^T{}_{(nxk)}\bullet V_{(nxk)}]^{-1} V^T{}_{(nxk)}} \qquad \text{Eq. 36}$$

Once the two weights and bias had been resized, the network was allowed to train until the error goal was met. However, the network trained just as efficiently letting Matlab randomly choose the weights for the reduced sized hidden node network. After this observation, these equations were dropped from the hidden-node pruning algorithm.

CHAPTER IV

RESULTS AND DISCUSSION

1. 2^3 Factorial Experiment

The results of the 2^3 factorial experiment for optimization of the particle beam are shown in Table 6. For each of the eight experiments run, the conditions of LC flow rate, helium pressure, and desolvation temperature are shown. The peak areas of the MS total ion chromatogram obtained for the compound Accent (100 μg/mL acetonitrile) are listed for trials one and two. Using a statistical t-test with an α value of 0.05, any effect listed with a value equal to or greater than 55 is significant, that is, changing that variable will have an effect on the peak area. The term "main effect" is used to signify the effect an independent variable (temperature, helium pressure, LC flow rate) has on a dependent variable (peak area) without interactions of any other independent variable. The two main effects that are significant are the helium pressure and the LC flow rate. However, the interactions of desolvation temperature with both flow rate and helium pressure are also seen to be significant. According to Box et al., [70] the main effect of a variable should be considered individually only if there is no evidence that the

Table 6. 2^3 Factorial Design Experiment

	LC Flow Rate	**Helium Pressure**	**Desol. Temp**	**Trial 1 Peak Area**	**Trial 2 Peak Area**	**Peak Area Average**
	mL/min	Psi	°C			
Expt 1	0.40 (-)	60 (-)	50 (-)	516	441	479
Expt 2	0.60 (+)	60 (-)	50 (-)	427	451	439
Expt 3	0.40 (-)	68 (+)	50 (-)	503	429	466
Expt 4	0.60 (+)	68 (+)	50 (-)	445	432	439
Expt 5	0.40 (-)	60 (-)	80 (+)	322	309	315
Expt 6	0.60 (+)	60 (-)	80 (+)	442	401	421
Expt 7	0.40 (-)	68 (+)	80 (+)	515	367	441
Expt 8	0.60 (+)	68 (+)	80 (+)	616	624	620
Main Effects					(All data/1000)	
Flow Rate = **55**				Variance s^2 = 2,241		
He Press = **78**				$Var_{(effect)} = (4/N)*S^2 = 560$		
Temp = -6				Std. $Error_{(effect)} = \sqrt{560} = 23.7$		
Two-Factor Interactions				Null hypothesis, H_o effect = 0		
FlowRatexHe Press = 21				Alternative hypothesis, H_a: effect $\neq$ 0		
FlowRatexTemp = **88**				$T_{(\alpha = 0.05)} = 2.306$		
HePress x Temp = **84**				Dof = 8		
Three-Factor Interactions				$\mu \pm t_{x\ s(effect)}$		
FlowRate x He Press x Temp = 15				$\mu \pm 2.306 \times 23.7 = \mu \pm$ **55**		

variable interacts with other variables. When interactions between factors are present, the interacting variables should be considered jointly. If a main effect of a variable is not significant, then that variable is assumed to have no influence on the outcome of the dependent variable and can be ignored.

In looking at the peak areas for the eight experiments, it can be seen that the best performance of the system in terms of peak area is at high flow rate, high helium pressure and high temperature (Expt. 8). Consequently, experiments were initially done with an LC flow rate of 0.6 mL/min, a helium pressure of 68 psi, and a desolvation temperature at 75°C. However, the flow rate was reduced to 0.4 mL/min in subsequent experiments to increase resolution between peaks on the LC chromatograms, even though this did reduce the peak areas.

2. Determination of the Neural Network Training Matrix

Mass spectra of the herbicides were prepared for analysis by the neural network by using the macro "Wrkdata" (see Appendix I), which was written with the Microsoft Excel macro-language. However, before data could be entered into the "Wrkdata" spreadsheet, it was first necessary to pre-process it by the HP Chemstation™ software. From the total ion chromatogram of each herbicide, a particular mass spectrum was selected in the "Data Analysis" module of Chemstation™. The background was subtracted, and under the "Spectrum" pull down menu, the "Tabulate" option was chosen. This produced a window listing the m/z ratios and abundances characterizing that particular mass spectrum. Using the on screen "Copy" button, the data were copied to the Microsoft Windows' clipboard, and pasted (control+V) into a blank worksheet in Microsoft Excel. Once in the new worksheet, the data were parsed into columns so that the individual data points could be processed. All non-numeric information was also deleted. The data were then manipulated into two columns (m/z ratio and abundance), and sorted in ascending order by m/z ratio. A graph of these two columns at this point would reproduce the mass spectrum. At this point, the rows between m/z ratios 100 to 250 were copied and pasted into the first two columns of "Wrkdata." "Wrkdata" then determined the average abundance for each whole number m/z ratio for each of the 47 m/z ratios that were used as inputs to the neural network.

These 47 abundances were saved to the computer's hard drive in a "Text (Tab delimited)" file format, where the file could be read into Matlab. Once in Matlab, the data could be treated as a single-variable, (47x1) column matrix. The data for all eight herbicides were then combined into one (47x8) matrix using each herbicide once, and one (47x40) matrix, where five different mass spectra were obtained for each herbicide. Both of these matrices were used to train the neural network.

3. Training the Network

Once the training matrices were created in Matlab, a neural network was constructed using the Matlab Neural Network Toolbox version 2.0 [71] consisting of 47 nodes in the input layer, 20 nodes in the hidden layer, and eight nodes in the output layer. The 20 hidden nodes were considered to be in excess of the optimum number. The hidden layer used a hyperbolic tangent transfer function, and the output layer used a sigmoid transfer function. The input training matrix, (47x8) or (47x40), was pre-processed with a normalizing function (see function Normal, Appendix I), which normalized each column of values between 0.1 and 0.9. The normalized training matrix is shown in Appendix III.

4. Determination of the Sum-Squared Error Goal

Choosing the correct value for the sum-squared error (SSE) goal is critical to avoid over-training and to prevent modeling the noise present in a set of data. Values for the SSE goal ranging from 0.1 to 0.4 were used to train a series of neural networks using the 8-training-sample training set. After each network was trained, the standard error of calibration, %SEC, (Equation 18), and standard error of prediction %SEP (Equation 19) were calculated. In order to make the equations apply to the type of data produced by the neural network, the values produced in the output layer (0.1 to 0.9) were used in place of concentrations (See Appendix IV). The %SEP and %SEC obtained for the different SSE goals are listed in Table 7. Trying to train the network with a SSE goal of less than 0.1 required so much time (>24 hours) that the training was stopped before the error goal was attained.

In order to determine the optimum SSE goal for the network, five trials were initially run using each of the SSE goals of 0.1, 0.2, 0.3, and 0.4, for a total of 20 runs. A statistical f-test was used to determine that the variances for the %SEC and %SEP were significantly different for each value of the SSE. This led to the "t-test: Two-Sample Assuming Unequal Variance" being used in Excel to determine if the %SEP was significantly higher than the %SEC. A SSE goal of 0.3 was chosen, since this was the minimum values at which the %SEP was not significantly greater than the %SEC (See Appendix VI and Table 7.) Five trials were also run using a SSE goal of 0.25, but the %SEP was significantly greater than the %SEC at this value also. Thus the optimum SSE goal for the neural network was fixed at 0.3 for the remainder of this work. This error goal was scaled up to 1.5 for the 40-training-sample network.

Once the SSE goal was chosen, the network was again trained with the training matrix. Starting with the excessive number of hidden nodes (20), the network re-sized itself after 1,500 iterations to contain 6 hidden nodes for the 8-

Table 7. Sum-Squared Error Goal, %SEP and %SEC

Sum-Squared Error	%SEP	%SEC
0.4	22.6	17.7
0.3	17.7	15.3
0.25	16.0 *	14.0
0.2	16.7 *	12.5
0.1	12.3*	8.85

* indicates that the %SEP is significantly greater than the %SEC

training sample network, and 7 hidden nodes for the 40-training-sample network. After re-sizing, the network was allowed to train by the regular, back-propagation method. The training took 2502 iterations to complete for the 8-training-sample network and 3928 iterations for the 40-training-sample network. Selected values of the training record after re-sizing from 20 to 6 hidden nodes are shown in Table 8 and a graph of the sum-squared error training record is shown in Figure 20.

After training, the network was ready to receive data from the mass spectrometer in order to identify the presence of one or more of the eight herbicides present in a sample, as a validation test.

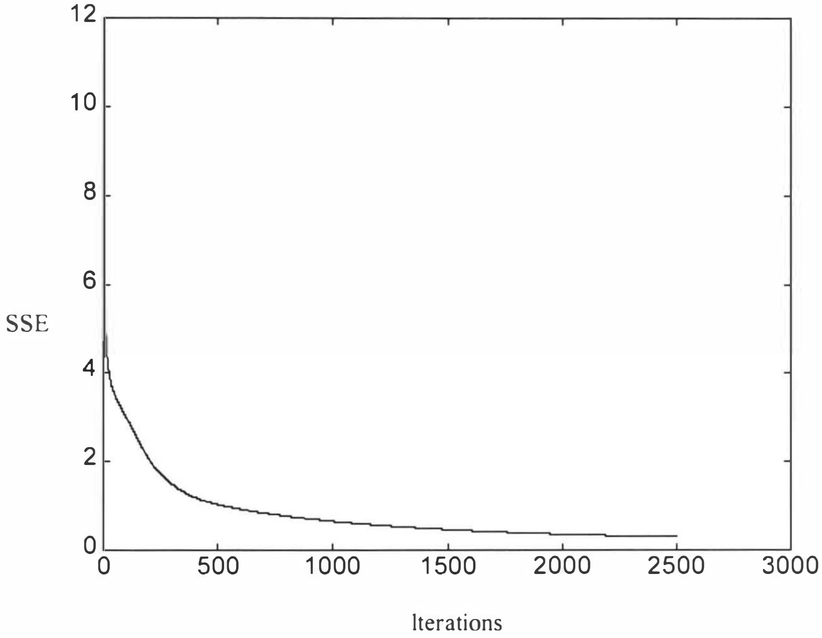


Figure 20. Plot of the Sum-Squared Error during Training. For a feed-forward, back-propagation ANN with 47 input neurons, 6 hidden neurons, and 8 output neurons

5. Determination of the Components of the Test Mixture

Setting the HP 5988a MS to obtain data in "scan mode", samples (400μg/mL acetonitrile) of each herbicide were injected into the HP1090 using a 20 μL sample loop. The LC was equipped with a Phenomenex LUNA C-18(2),

Table 8. Selected Values of the Training Record. For ANN with 47 input nodes, 6 hidden nodes, and 8 output nodes. SSE of 0.3.

Record Number	Sum-Squared Error
1	10.7
10	4.91
100	3.01
500	1.03
1000	0.655
1500	0.468
2000	0.361
2500	0.300
2502	0.2999

150 x 4.60 mm, 5-µm particle size column. The flow rate was set to 0.4 mL/min. to assure good peak separation. The mobile phase was (75:25) acetonitrile/water, both containing 0.05% (v/v) acetic acid.

Retention times and peak widths are shown in Table 9. LC chromatograms for all eight herbicides are shown in Figures 21-28. Based on their LC retention

Table 9. Retention Times and Peak Widths of Sulfonylurea Herbicides

Herbicide	Retention Time (minutes)	Peak Width (minutes)
Accent	4.2	0.75
Ally	4.7	0.63
Classic	6.4	0.94
Glean	5.0	0.70
Express	5.7	0.88
Harmony	4.6	0.75
Londax	5.3	0.93
Oust	4.8	0.63

times, four herbicides were selected to test the ability of the neural network to identify the herbicides present in a mixture in soil. Accent, Oust, Londax, and Classic were chosen for this mixture. As can be seen in Figure 29, a simulated chromatogram of this mixture, which was created by superimposing the four

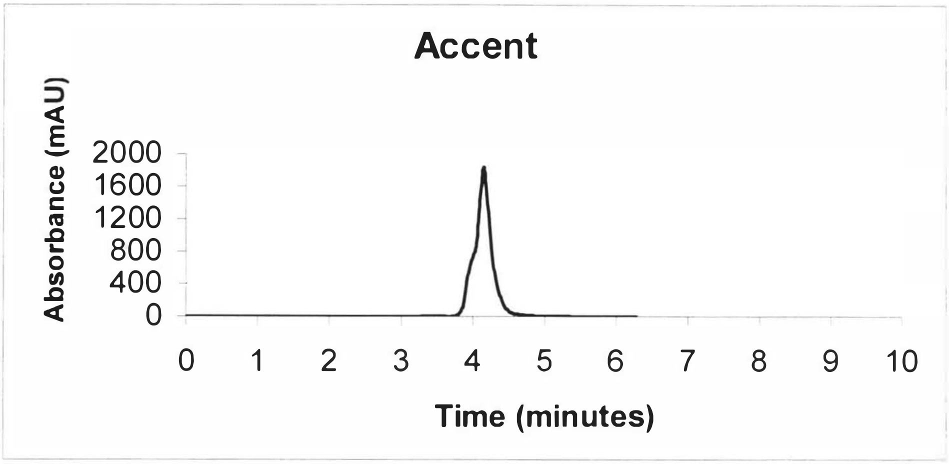


Figure 21. Liquid Chromatogram of Accent The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 µm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

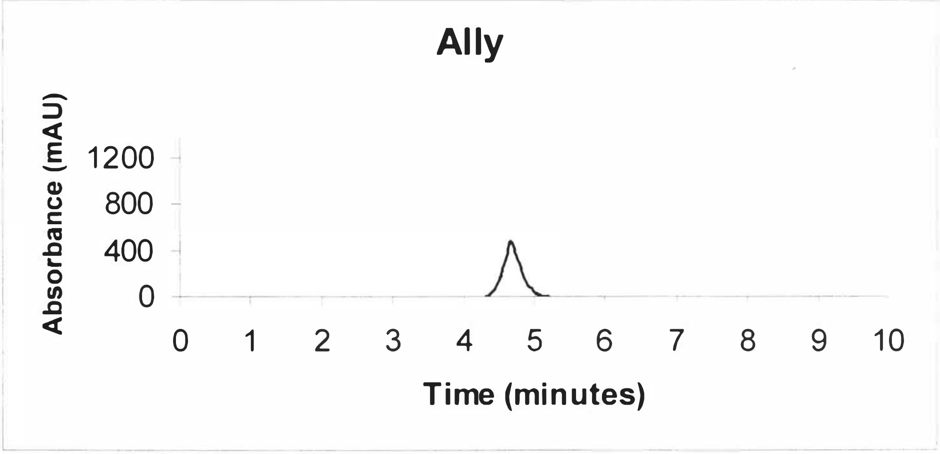


Figure 22. Liquid Chromatogram of Ally. The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 µm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

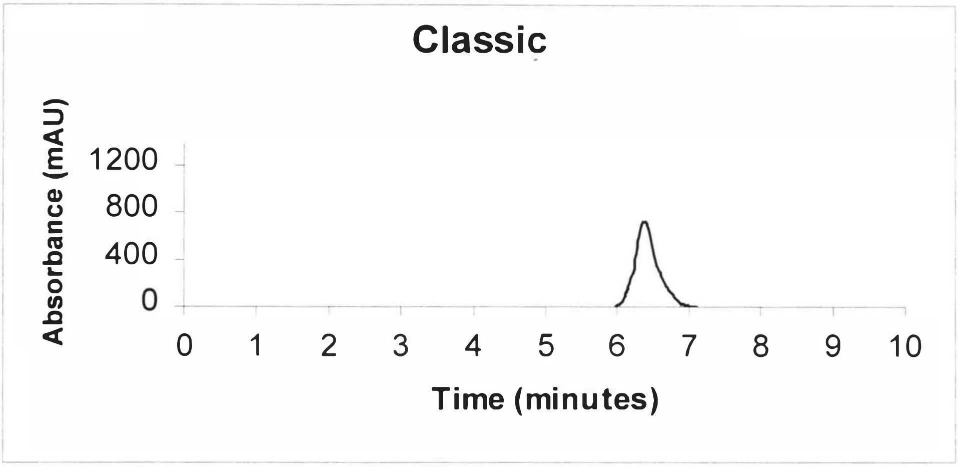


Figure 23. Liquid Chromatogram of Classic. The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 μm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

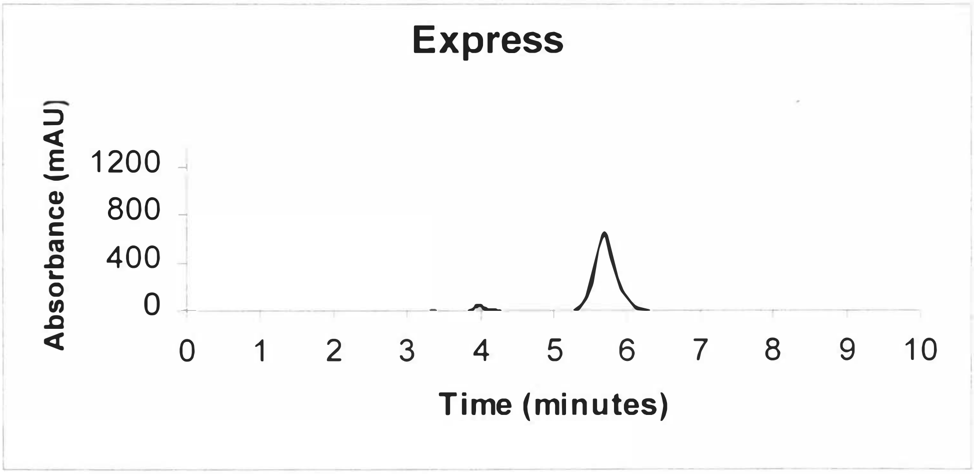


Figure 24. Liquid Chromatogram of Express. The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 μm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

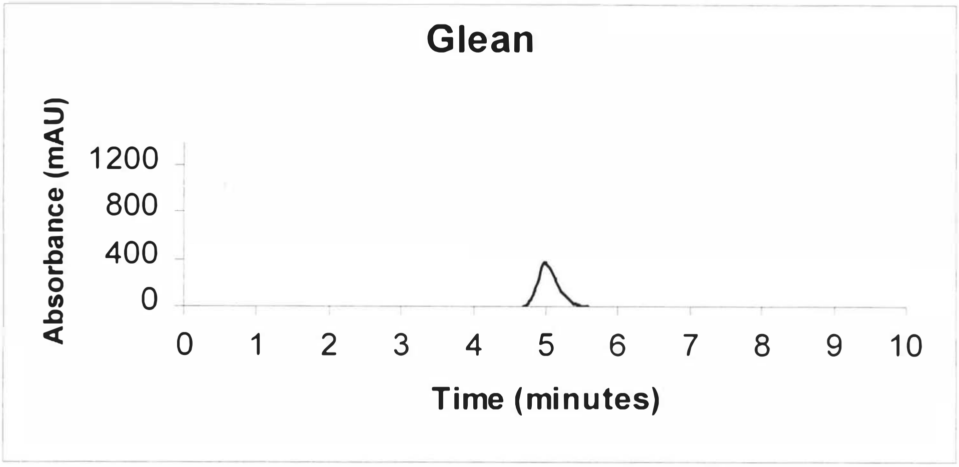


Figure 25. Liquid Chromatogram of Glean The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 µm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

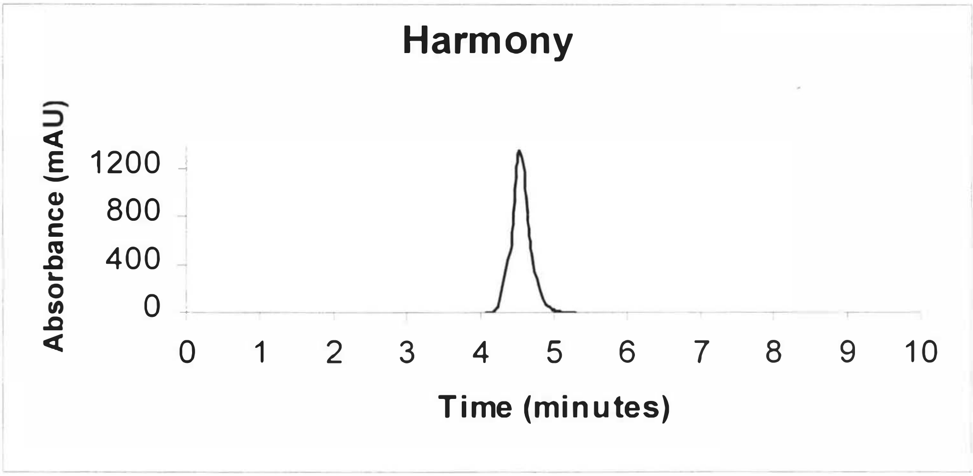


Figure 26. Liquid Chromatogram of Harmony. The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 µm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

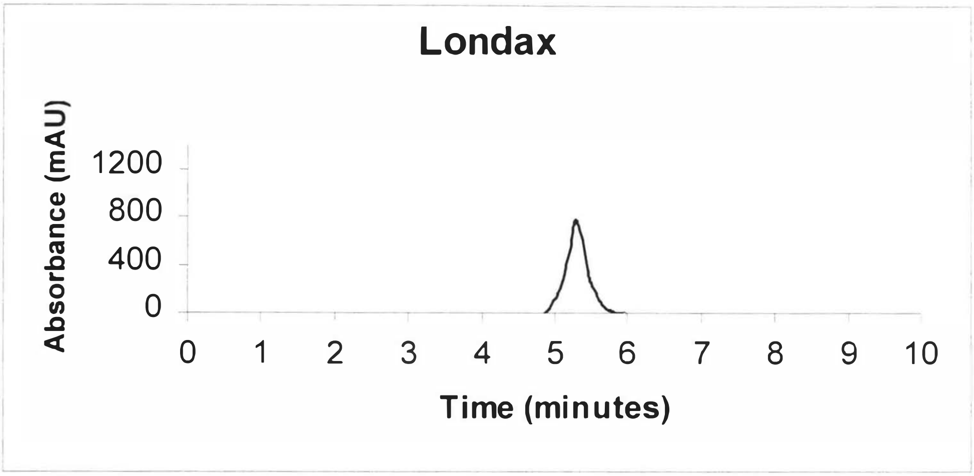


Figure 27. Liquid Chromatogram of Londax. The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 μm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

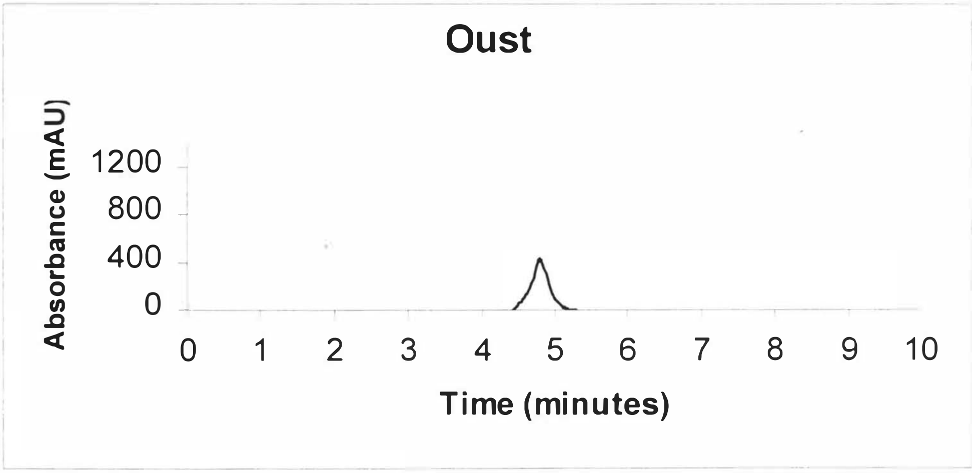


Figure 28. Liquid Chromatogram of Oust. The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 μm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

Table 10. Resolution of Selected Sulfonylurea Herbicides

Peak 1 – Peak 2	Resolution
Accent – Oust	0.87
Oust – Londax	1.4
Londax – Classic	1.2

chromatograms onto one graph, these compounds were expected to have moderate separation, with resolution at least above one-half. The actual chromatogram appears in Figure 30. The resolutions were calculated using the standard resolution formula given in Equation 37 and are listed in Table 10.

$$R = 2(tr_b – tr_a) / (w_a + w_b) \qquad \text{Eq. 37}$$

In Equation 37, tr_b and tr_a are the retention times for peaks b and a respectively, and w_a and w_b are the baseline widths of peaks a and b.

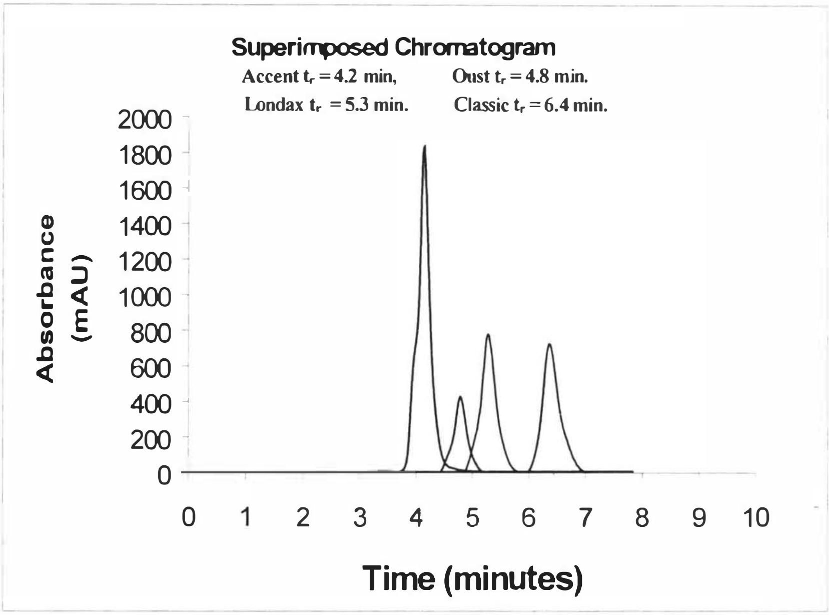


Figure 29. Superimposed Chromatograms for Accent, Oust, Londax, and Classic

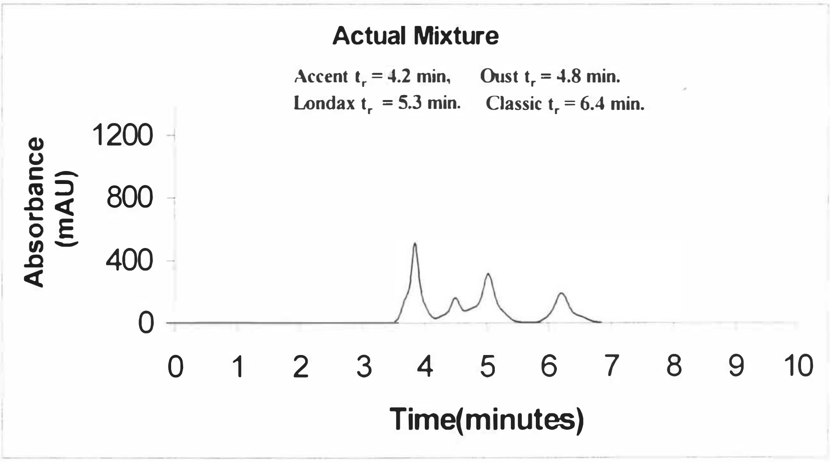


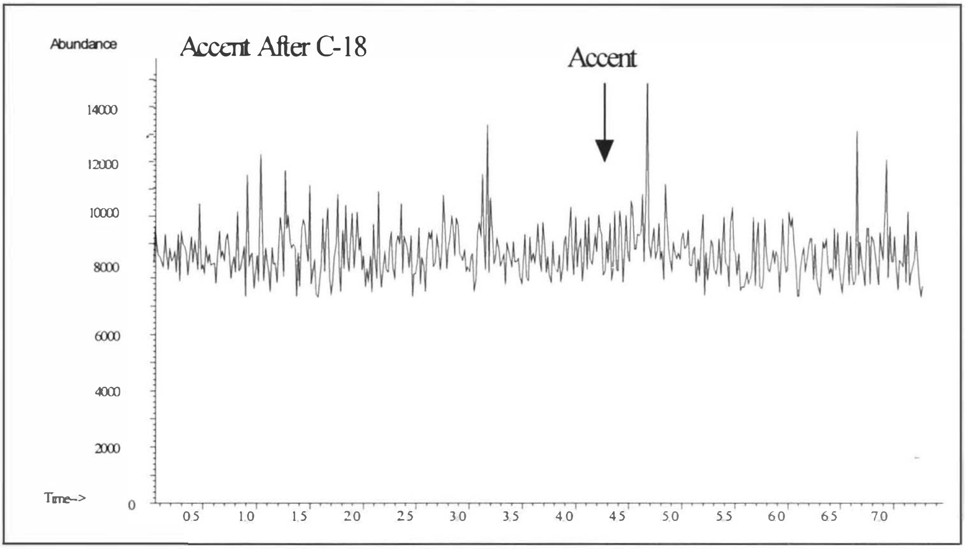
Figure 30. Actual Chromatogram of a Mixture of Accent, Oust , Londax, and Classic. The flow rate was 0.4 mL/min, the mobile phase was (75:25) ACN/H2O, both containing 0.05% HAc. The column was a Phenomenex LUNA C-18(2), 150 x 4.60 mm, 5 μm particle size column. UV detection was at 254 nm. Concentration was 100ug/mL.

6. Analysis of Extracted Mixture

The mixture was prepared by accurately pipeting one mL of each of the 400 μg/mL solutions of each herbicide into a single vial. The concentration of the herbicide was thus diluted to 100 μg/mL each, which accounts for the difference in the absorbance between the superimposed and actual LC chromatograms (Figures 29 and 30.)

The analysis was carried out in the following manner. The four individual herbicides and the mixture were mixed with individual soil samples as described in the Experimental Section. Following that procedure, the first attempt at analysis came after the C-18 extraction. The total ion chromatograms of the individual herbicides and the mixture are shown in Figures 31 and 32. A two-minute window on either side of the expected MS retention time for each herbicide was used to create an average MS of each particular herbicide for analysis. At this point, neither the Hewlett-Packard-Probability Based Matching (HP-PBM) system nor either of the neural networks could identify the presence of any of the herbicides in the individual samples or the mixture.

The next attempt at analysis came after the SAX extraction. The total ion chromatogram of the mixture is shown in Figure 33. Since the individual herbicide peaks were not readily apparent, the retention times from previous runs were used to locate the mass spectrum for each herbicide. Analyzing for Accent,



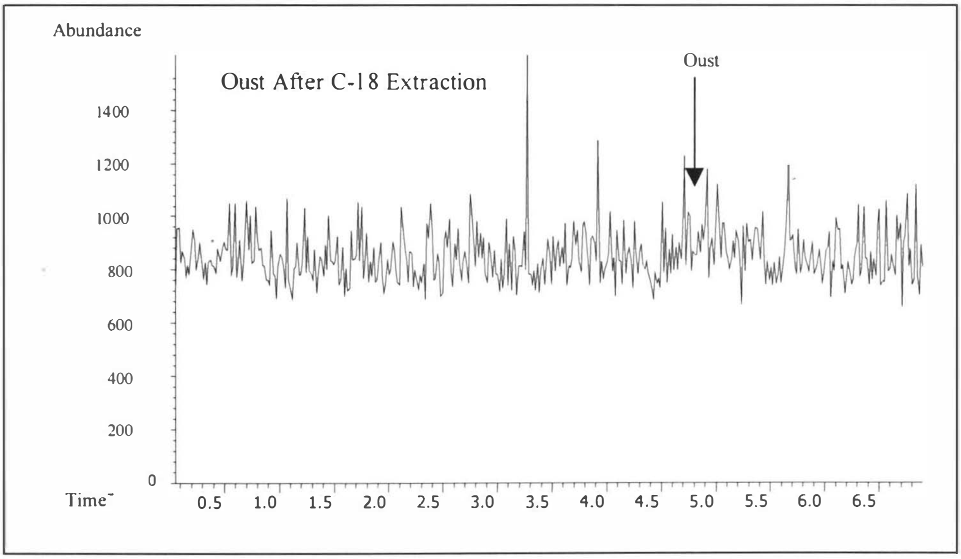


Figure 31. TIC of Accent and Oust After C-18 Extraction. Arrows indicate the expected retention times of the herbicides in the liquid chromatograms.

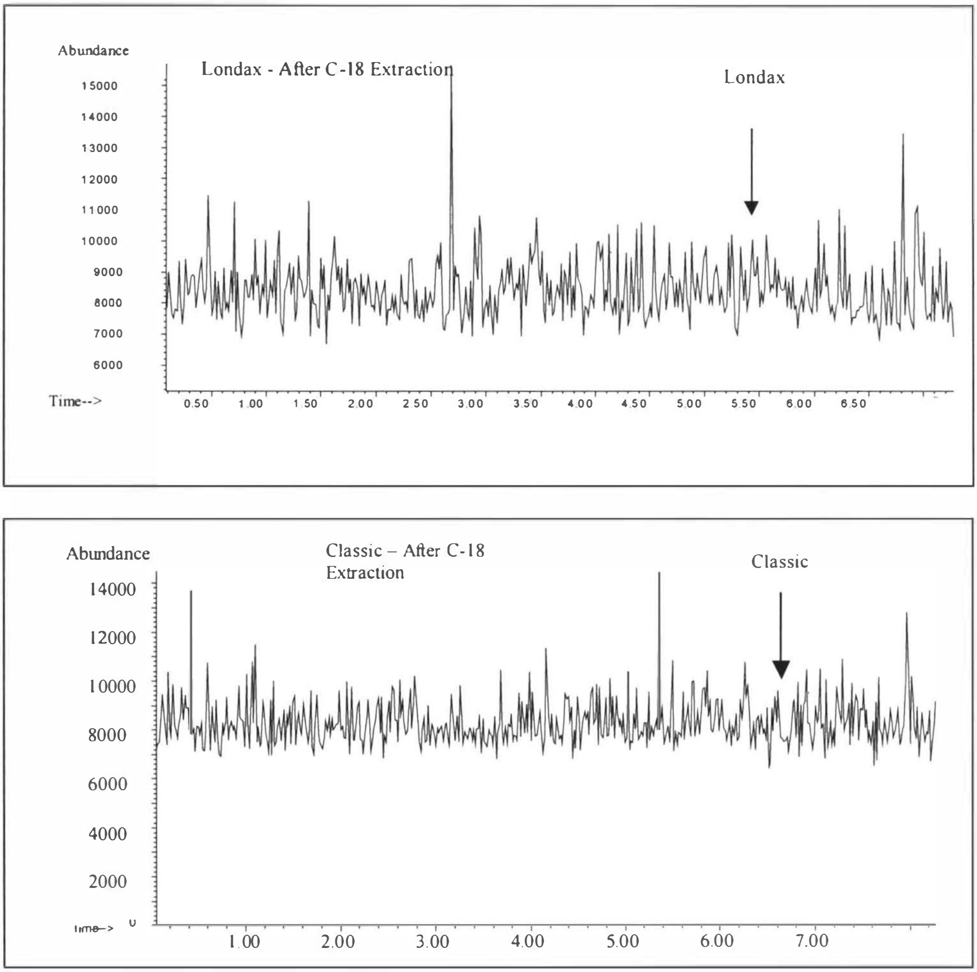


Figure 32. TIC of Londax and Classic After C-18 Extraction. Arrows indicate the expected retention times of the herbicides in the liquid chromatograms.

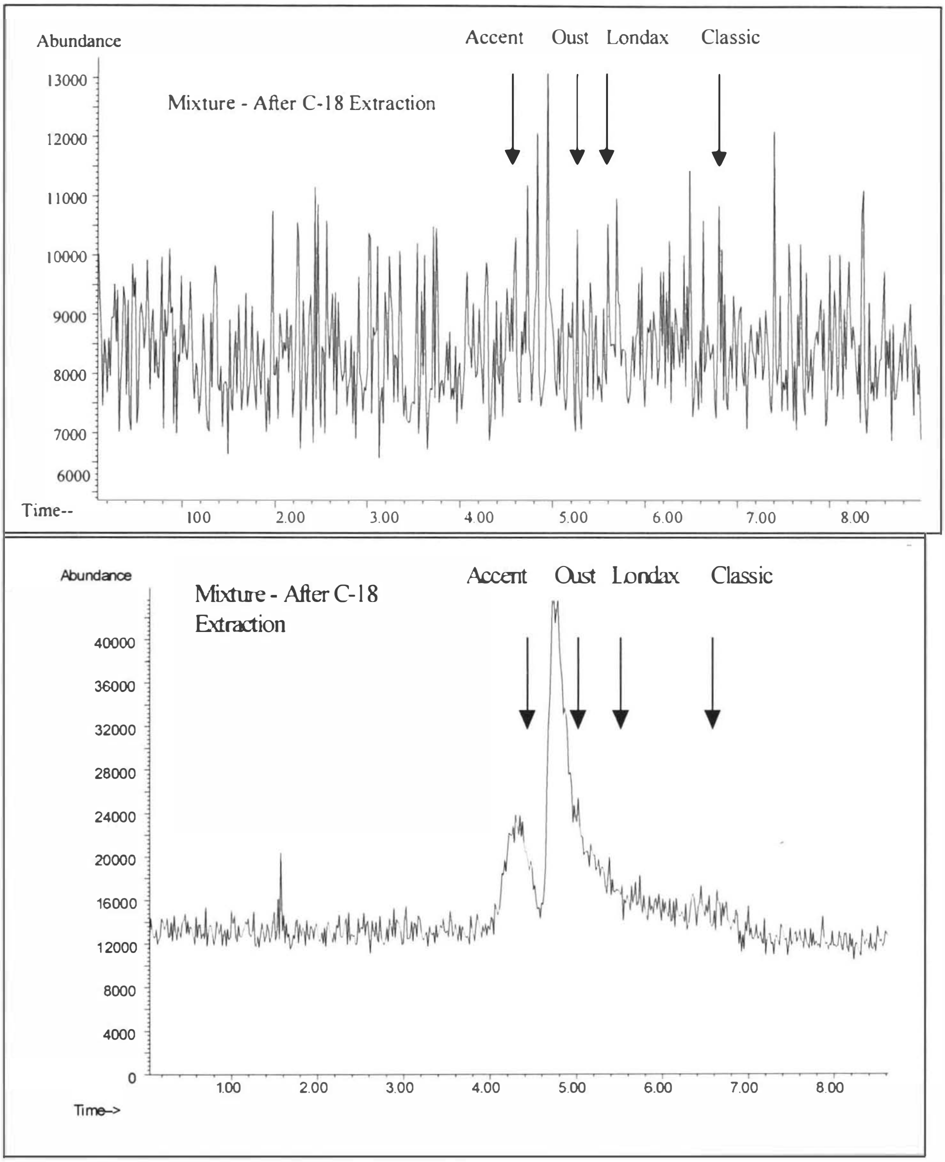


Figure 33. TICs of the Mixture After the C-18 and SAX Extractions. Arrows indicate the expected retention times of the herbicides in the liquid chromatograms.

Londax, Oust, and Classic after the SAX extraction, the following results were obtained: both the 8-training-sample neural network and the HP-PBM system identified Accent, Oust, and Classic, while the 40-training-sample neural network identified Accent, Oust and Londax. The HP-PBM system failed to identify the presence of Londax in the mass spectrum, as is shown in Figures 34-37. The results for the individual herbicides after the SAX extraction are shown in Figures 40- 47

As seen in Figure 40, the TIC of the extracted Accent after the SAX extraction shows two detectable peaks. The first peak is Accent. The second peak is a matrix peak. The HP-PBM system correctly identified the mass spectrum of the Accent peak. When the trained neural networks processed the data, both networks were also able to identify the peak as Accent as shown in Figure 41.

Overall, both neural networks were able to identify Accent, Oust, Londax, and Classic. The HP-PBM system was also able to identify Accent, Londax, and Classic, but misidentified Oust as Classic. However, it only ranked the quality of the match at 4%, which was not very good. These results are shown in Figures 42 through 47. The blanks were also analyzed. The TICs are shown in Figures 38 and 39, however, no detection of any herbicides was made in either blank.

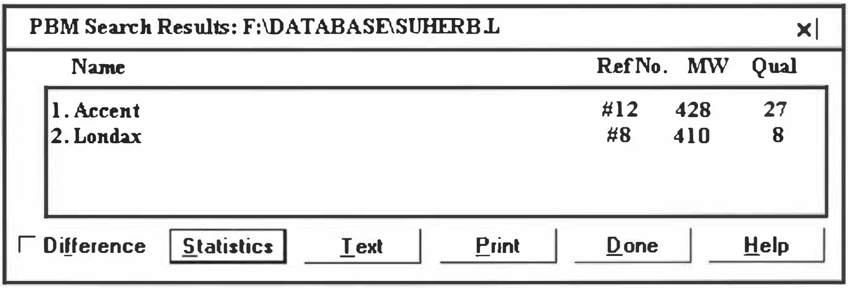
Neural Network Output Layer (40-training-sample ANN)

0.7682	**→**	**Accent**
0.0319	→	Ally
0.0087	→	Classic
0.0073	→	Express
0.0056	→	Glean
0.0004	→	Harmony
0.1476	→	Londax
0.0008	→	Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.6097	**→**	**Accent**
0.0850	→	Ally
0.0000	→	Classic
0.0000	→	Express
0.0457	→	Glean
0.1212	→	Harmony
0.1388	→	Londax
0.0000	→	Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Accent	#12	428	27
2. Londax	#8	410	8

Figure 34. Neural Network and HP-PBM Results for Accent in Mixture After SAX Extraction.

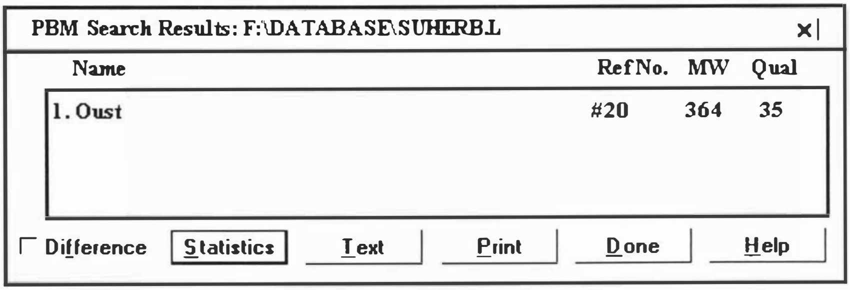
Neural Network Output Layer (40-training-sample ANN)

0.0094→ Accent
0.0037→ Ally
0.0163→ Classic
0.0053→ Express
0.0119→ Glean
0.0058→ Harmony
0.3865→ Londax
0.4574→ Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0006→ Accent
0.1985→ Ally
0.0895→ Classic
0.6858→ Express
0.0061→ Glean
0.1208→ Harmony
0.5577 → Londax
0.9857 → Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Oust	#20	364	35

Figure 35. Neural Network and HP-PBM Results for Oust in Mixture After SAX Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0167 → Accent

0.0270 → Ally

0.0430 → Classic

0.0024→ Express

0.1536 → Glean

0.0050 →Harmony

0.7792 → Londax

0.1536 → Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0000 → Accent

0.0246→ Ally

0.0806→ Classic

0.6328 →Express

0.1630 →Glean

0.1315 →Harmony

0.9109→ Londax

0.4520 → Oust

HP-PBM Result – "No hits found for this spectrum"

Figure 36. Neural Network and HP-PBM Results for Londax in Mixture After SAX Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0007 → Accent

0.0464 → Ally

0.2307 → Classic

0.0040 → Express

0.0022 → Glean

0.0004 → Harmony

0.0004 → Londax

0.0828 → Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0414 → Accent

0.0081→ Ally

0.9085→ Classic

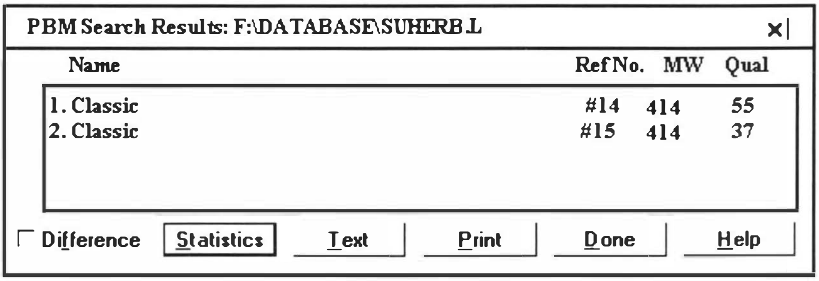
0.0575 → Express

0.0736 →Glean

0.0250 →Harmony

0.0007→ Londax

0.0711 → Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Classic	#14	414	55
2. Classic	#15	414	37

Figure 37. Neural Network and HP-PBM Results for Classic in Mixture After SAX Extraction.

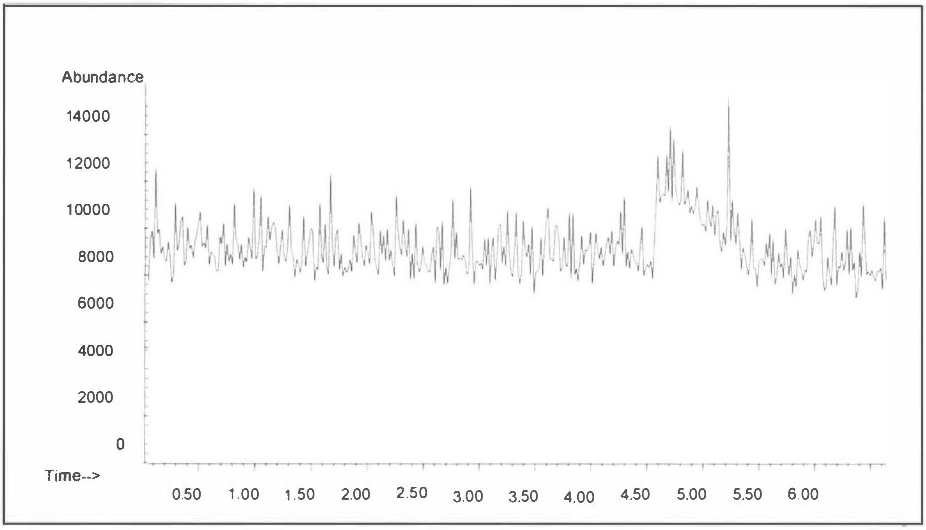


Figure 38. TIC of Blank after SAX Extraction.

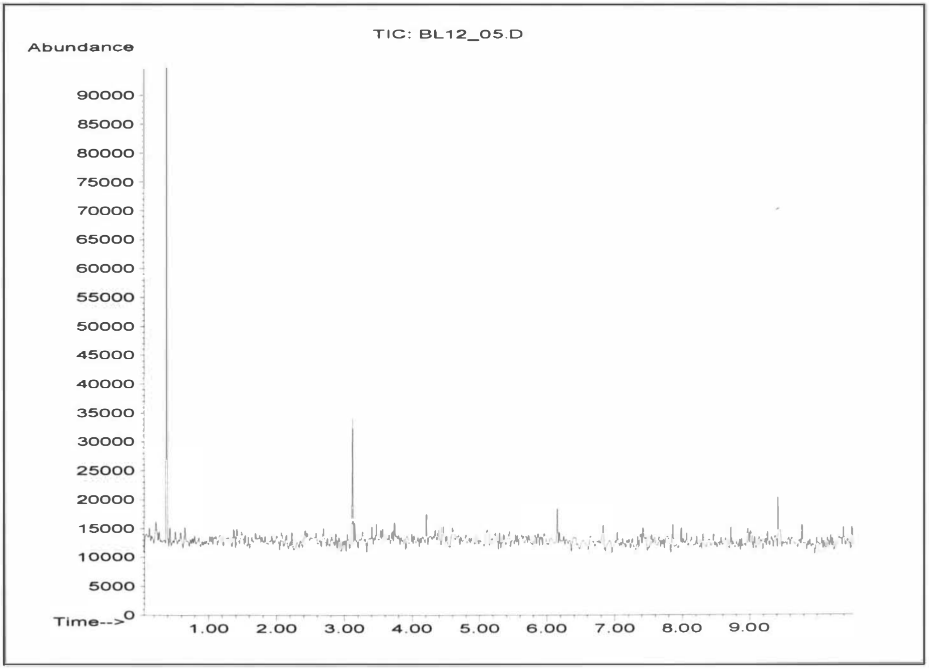
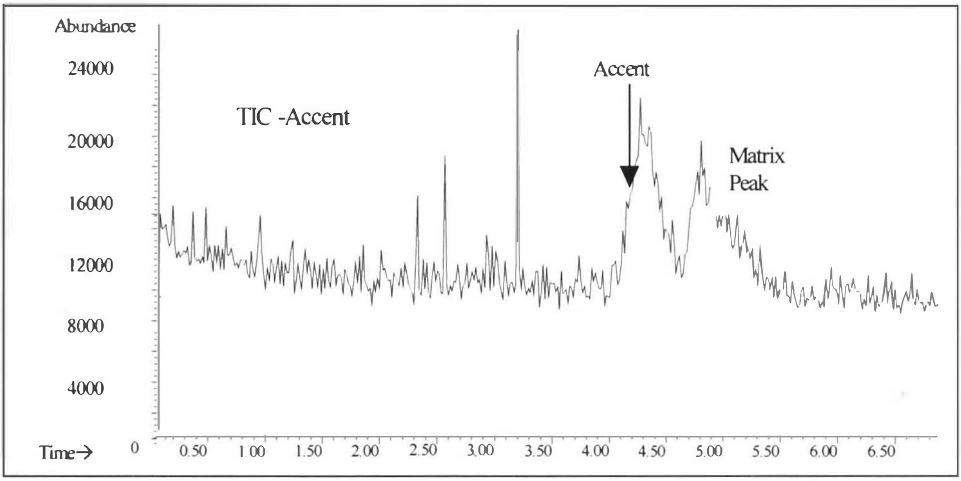


Figure 39. TIC of Blank after Si Extraction.

The final test of the ability of the neural network to identify the herbicides came after the Silica-SPE step. The total ion chromatogram of the mixture is shown in Figure 48. Accent was identifiable by both neural networks and the HP-PBM system, as shown in Figure 51. The 8-training-sample network was also able to identify Classic. The HP-PBS System identified only Accent, as is shown in Figures 52 to 54. In terms of the individual herbicides extracted from soil, the 8-training-sample network was able to identify Accent, Classic, and Oust. The 40-training-sample network was able to identify Accent and Oust. The HP-PBM system was able to identify only Accent, as shown in Figures 53 through 60. Tables 11 and 12 are summaries of these results.

Comparison of the extraction results with those listed in Table 3 reveal some unexpected results. There appears to be very little correlation between the extraction efficiency obtained by Dupont and the Minnesota Department of Agriculture (MDA) using electrospray MS and the efficiency of a particle beam.



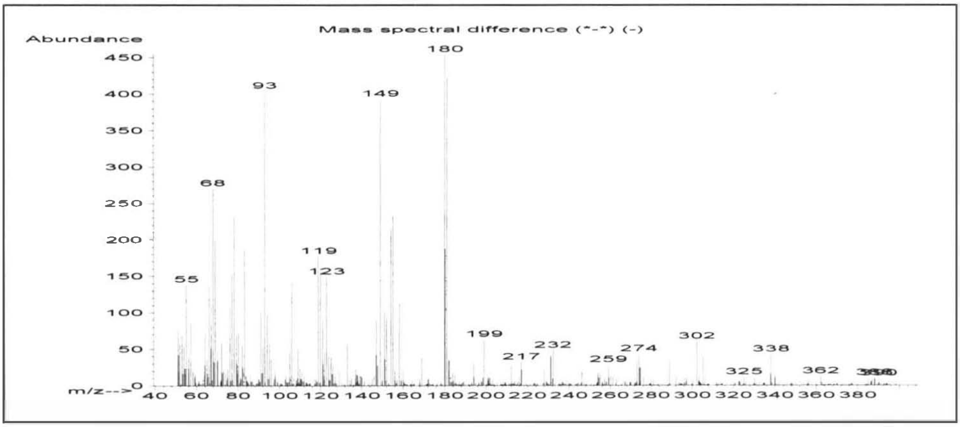


Figure 40. TIC and MS of Accent after SAX Extraction.

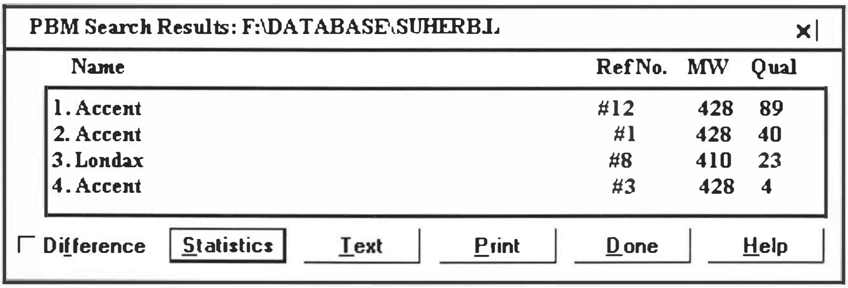
Neural Network Output Layer (40-training-sample ANN)

0.6570→	**Accent**
0.0032 →	Ally
0.0102→	Classic
0.0100→	Express
0.0104→	Glean
0.0005→	Harmony
0.5878→	Londax
0.0106→	Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.5327 →	**Accent**
0.0688→	Ally
0.0569→	Classic
0.0793→	Express
0.0323 →	Glean
0.0135 →	Harmony
0.1771 →	Londax
0.2581→	Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Accent	#12	428	89
2. Accent	#1	428	40
3. Londax	#8	410	23
4. Accent	#3	428	4

☐ Difference

Figure 41. Neural Network and HP-PBM Results for Accent After SAX Extraction.

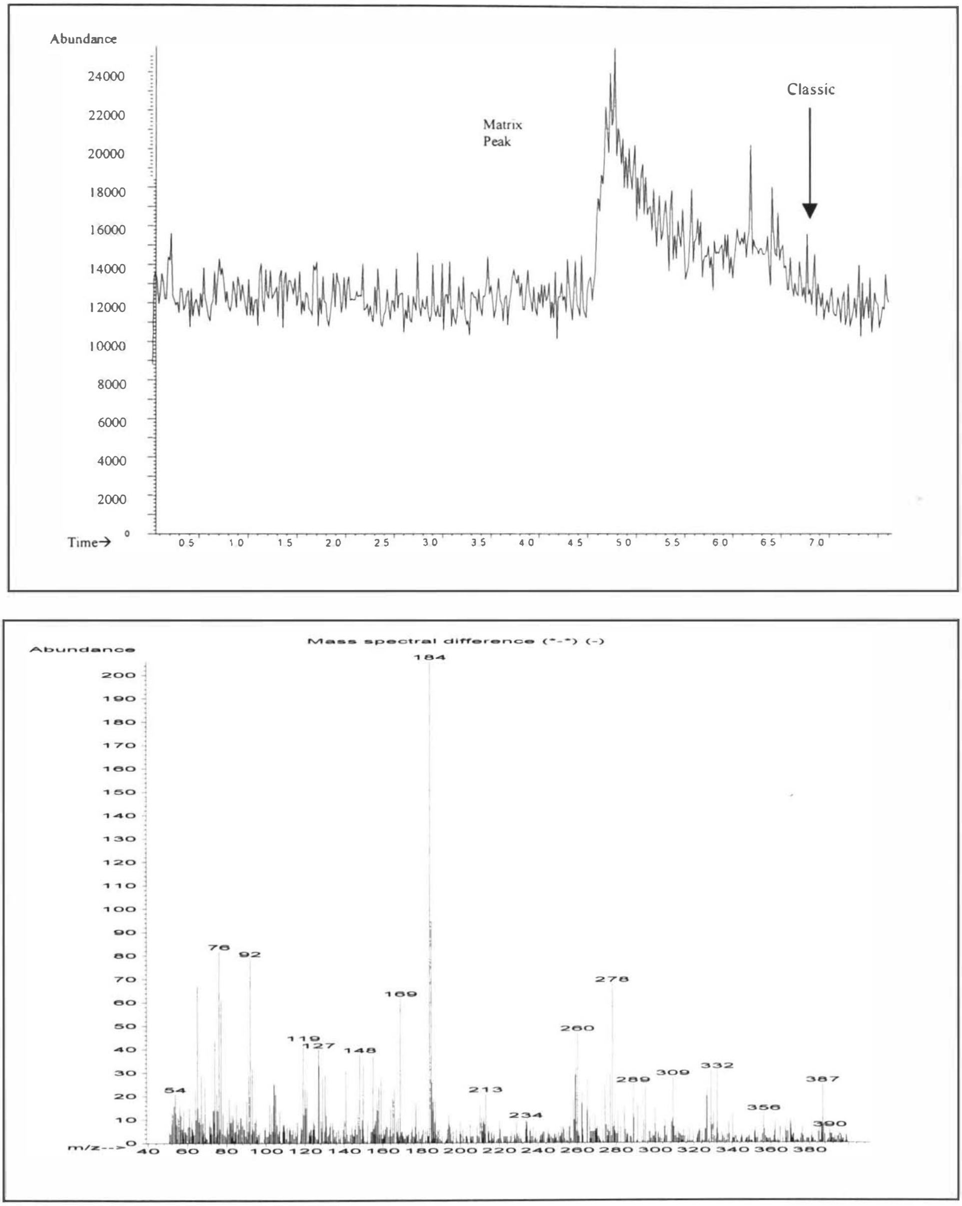


Figure 42. TIC and MS of Classic after SAX Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0007→ Accent

0.0464→ Ally

0.2307→ Classic

0.0040→ Express

0.0022→ Glean

0.0004→ Harmony

0.0004→ Londax

0.0828→ Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0290 → Accent

0.0103 → Ally

0.6458 → Classic

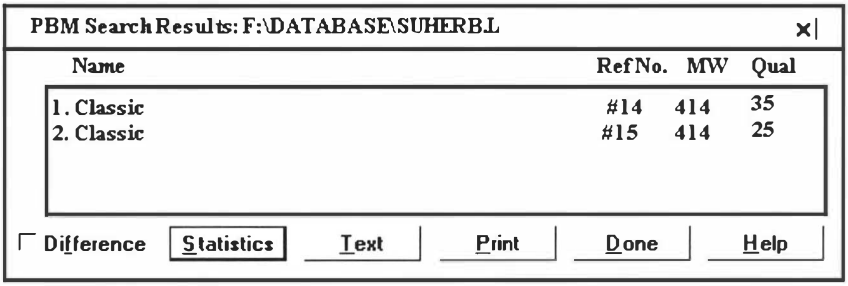
0.0591 → Express

0.1859 → Glean

0.1749 → Harmony

0.0243 → Londax

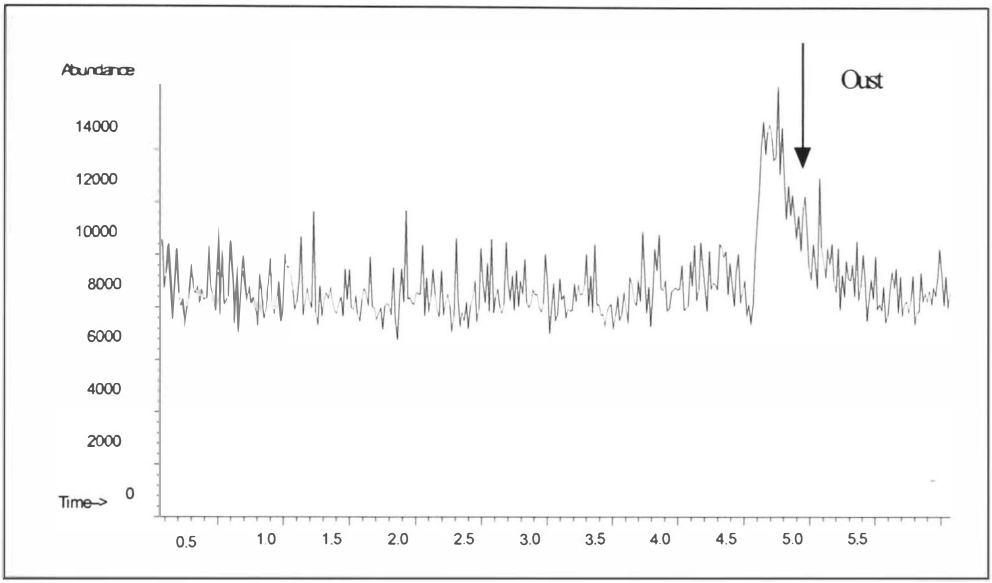
0.0405 → Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Classic	#14	414	35
2. Classic	#15	414	25

Figure 43. Neural Network and HP-PBM Results for Classic After SAX Extraction.



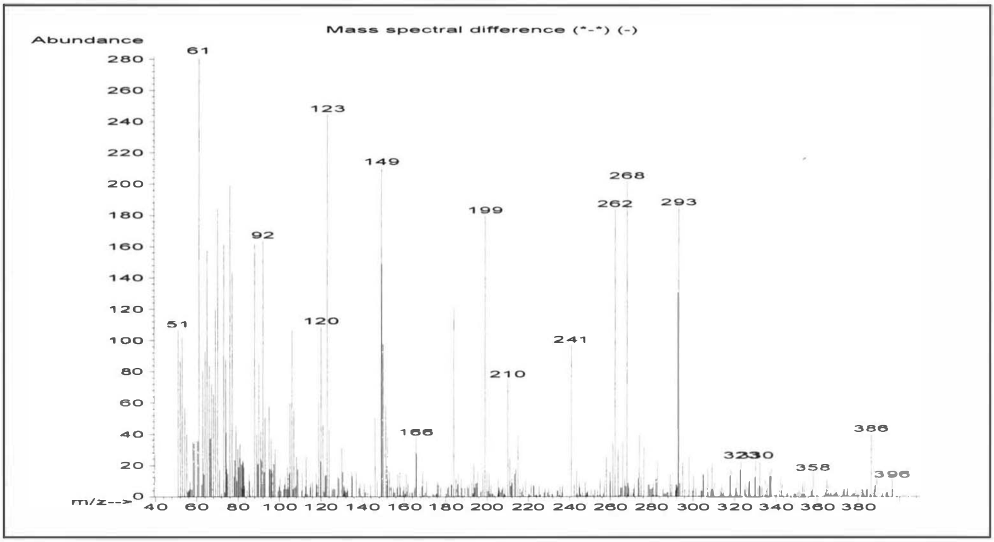


Figure 44. TIC and MS of Oust after SAX Extraction.

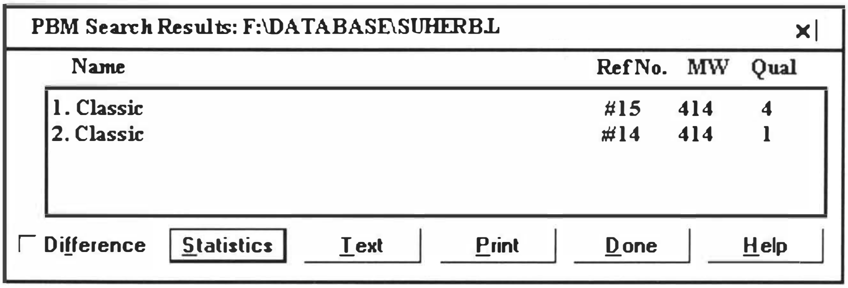
Neural Network Output Layer (40-training-sample ANN)

0.0009→	Accent
0.0027→	Ally
0.0091→	Classic
0.0091→	Express
0.0188→	Glean
0.0561→	Harmony
0.4351→	Londax
0.7920 →	**Oust**

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.1960 →	Accent
0.0614 →	Ally
0.1857 →	Classic
0.0028 →	Express
0.1882 →	Glean
0.0094 →	Harmony
0.0173 →	Londax
0.8223 →	**Oust**



PBM Search Results: F:\DATABASE\SUHERBL

Name	Ref No.	MW	Qual
1. Classic	#15	414	4
2. Classic	#14	414	1

Figure 45. Neural Network and HP-PBM Results for Oust After SAX Extraction.

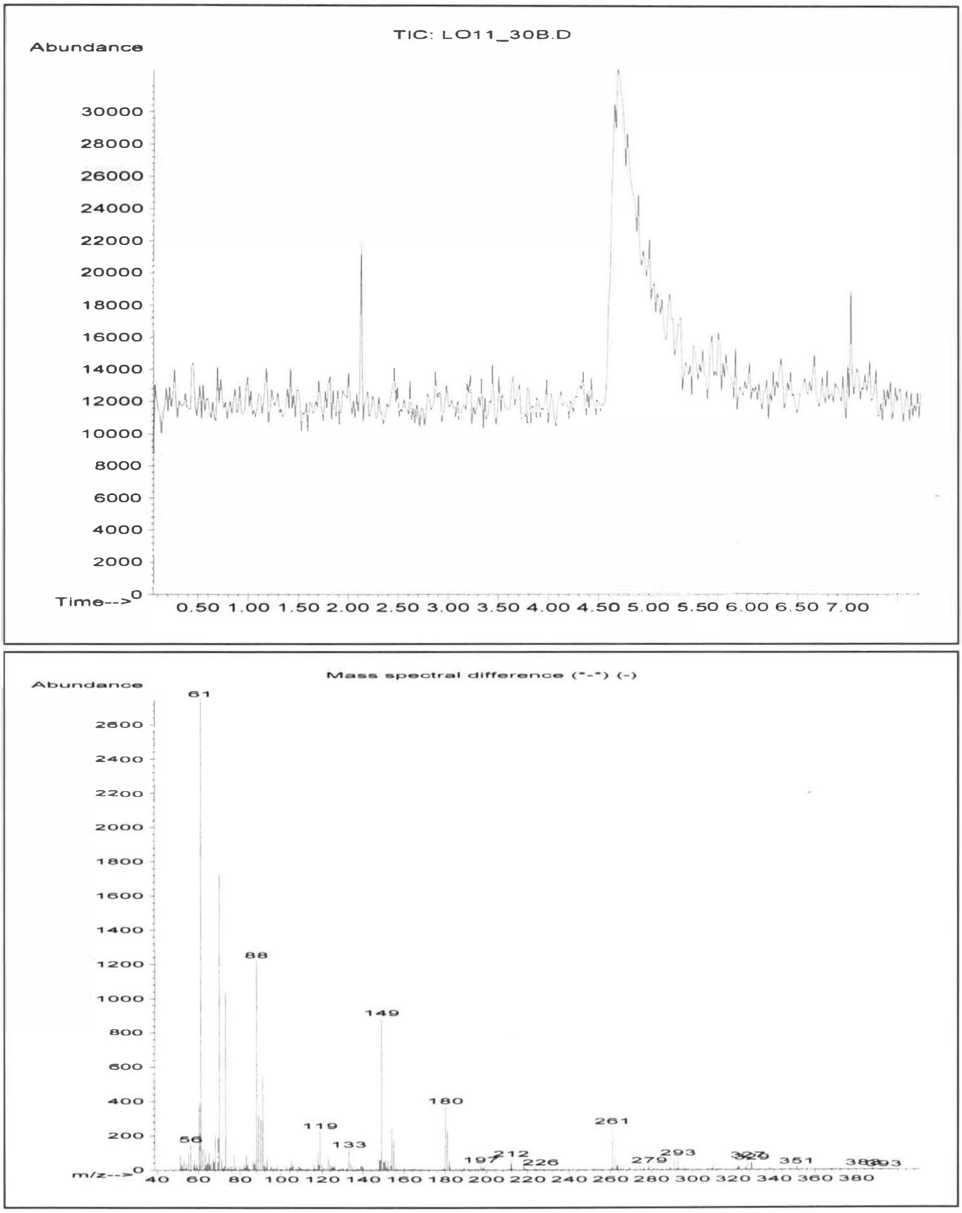


Figure 46. TIC and MS of Londax after SAX Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0144→ Accent

0.0062→ Ally

0.0061→ Classic

0.0025→ Express

0.0075→ Glean

0.0207 →Harmony

0.9446 Londax

0.1009→ Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.3394 → Accent

0.0580 → Ally

0.0675 → Classic

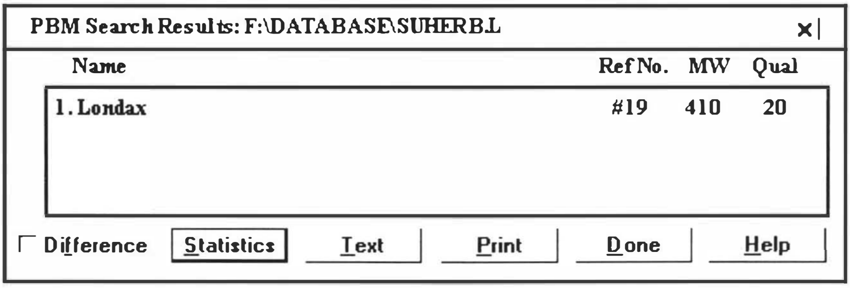
0.1961 → Express

0.0427 → Glean

0.0695 → Harmony

0.6068 → Londax

0.1142 → Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Londax	#19	410	20

Figure 47. Neural Network and HP-PBM Results for Londax After SAX Extraction.

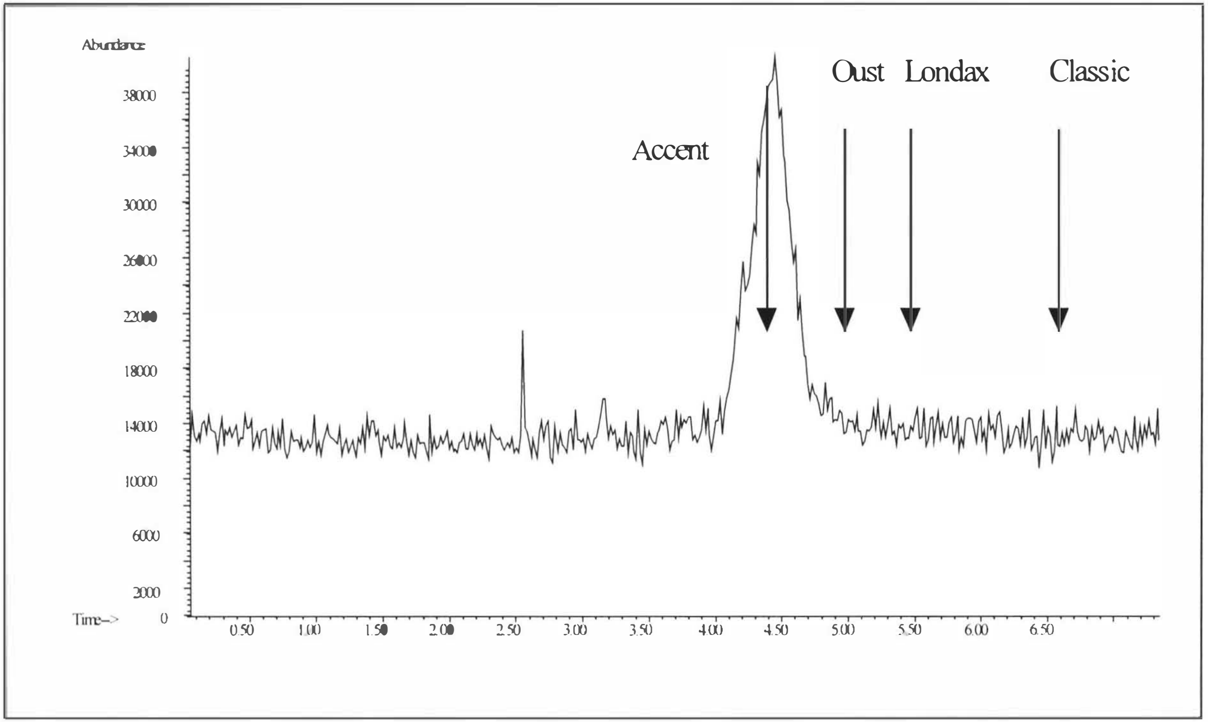


Figure 48. TIC of the Mixture After the Silica Extraction.

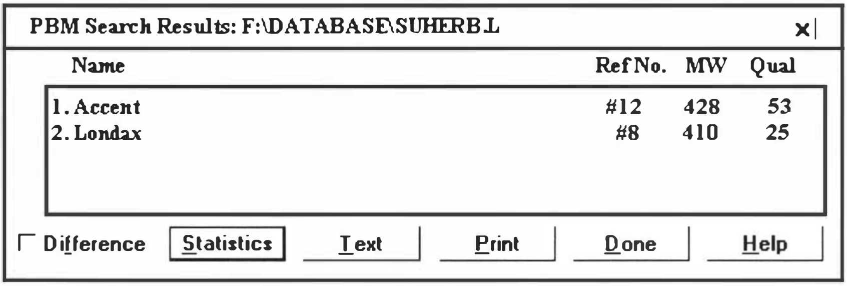
Neural Network Output Layer (40-training-sample ANN)

0.9429→	**Accent**
0.0069→	Ally
0.0068 →	Classic
0.0273→	Express
0.0114→	Glean
0.0001→	Harmony
0.0104→	Londax
0.0007→	Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.6732	**→ Accent**
0.0527	→ Ally
0.0406	→ Classic
0.1133	→ Express
0.0218	→ Glean
0.0092	→ Harmony
0.1326	→Londax
0.1398	→ Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Accent	#12	428	53
2. Londax	#8	410	25

Figure 49. Neural Network and HP-PBM Results for Accent in Mixture After Silica Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.1067→ Accent
0.1904→ Ally
0.1656 → Classic
0.9772→ Express
0.1149→ Glean
0.0084 →Harmony
0.0002→ Londax
0.0909→ Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0442→ Accent
0.0199→ Ally
0.5747 → Classic
0.0117 → Express
0.2001→ Glean
0.0441 →Harmony
0.0041→ Londax
0.2050→ Oust

HP-PBM Result – “No hits found for this spectrum”

Figure 50. Neural Network and HP-PBM Results for Classic in Mixture After Silica Extraction.

Neural Network Output Layer (40-training-sample ANN)

- 0.0028→ Accent
- 0.5432→ Ally
- 0.1681→ Classic
- 0.1014 → Express
- 0.0100→ Glean
- 0.0060→ Harmony
- **0.0003 Londax**
- 0.3978→ Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0575→ Accent

0.0093→ Ally

0.6024 → Classic

0.0007 → Express

0.3674→ Glean

0.0069 →Harmony

0.0001→ Londax

0.3493→ Oust

HP-PBM Result – "No hits found for this spectrum"

Figure 51. Neural Network and HP-PBM Results for Londax in Mixture After Silica Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0109→ Accent

0.0070→ Ally

0.0222 → Classic

0.0205 →Express

0.0043 → Glean

0.0008 →Harmony

0.0005 → Londax

0.0118 → Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0530→ Accent

0.0298→ Ally

0.2376→ Classic

0.0110 → Express

0.2808 →Glean

0.1018→ Harmony

0.0380→ Londax

0.1909→ Oust

HP-PBM Result – "No hits found for this spectrum"

Figure 52. Neural Network and HP-PBM Results for Oust in Mixture After Silica Extraction.

Three of the four herbicides used in the mixture were used in the Dupont/MDA study. Accent had average percent recoveries of 50% by Dupont and 56% by the MDA. The average percent recovery of Oust was 77% by Dupont and 87% by the MDA. Classic had an average recovery of 76% by Dupont and 148% by the MDA! It appears as if some type of contamination occurred at the MDA with Classic. The high percent recoveries are due in some part to the use of selected ion monitoring (SIM) with the mass spectrometer. In this research using the particle beam, Accent proved to be the easiest herbicide to identify, while Classic and Oust were moderately difficult to identify. This is in contrast with the electrospray results. Perhaps the ability of the electrospray interface to find the molecular ion, coupled with the use of the SIM technique accounts for these different results.

In comparison to other research in the area of using ANNs in the identification of mass spectrometry, several facts may be noted. Development has generally fallen along two lines. Some ANNs have been designed to identify specific species, such as trimethylsilyl derivatives of certain organic acids [72], binary mixtures of microbial samples using pyrolysis MS [73], and steroids-like compounds. [74] The work developed in this research falls into this first category, since the training was accomplished using only sulfonylurea herbicides.

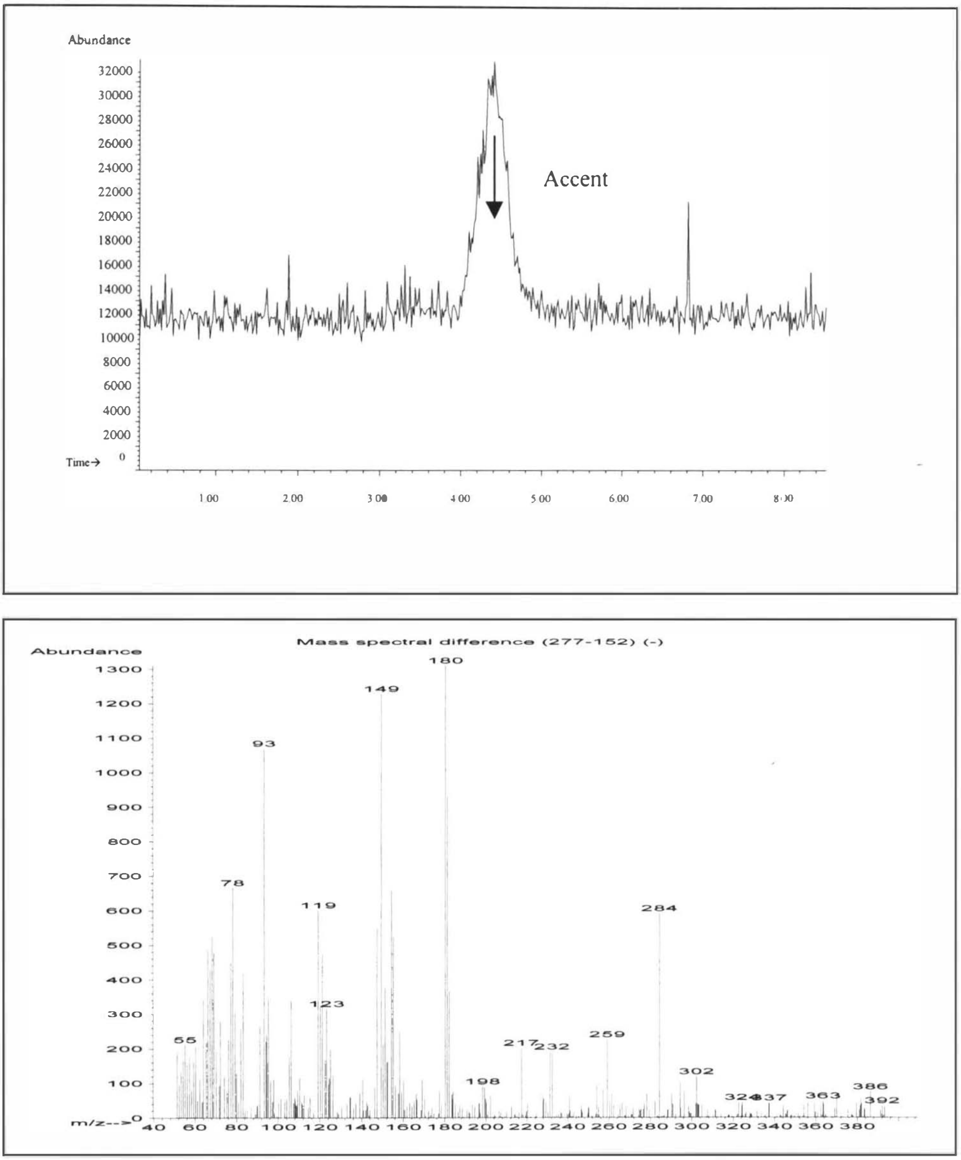


Figure 53. TIC and MS of Accent after Silica Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.9215→ Accent

0.0343 → Ally

0.0147 → Classic

0.0123 → Express

0.0077 → Glean

0.0001 → Harmony

0.0062 → Londax

0.0005 → Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.4527 → Accent

0.0625 → Ally

0.0324 → Classic

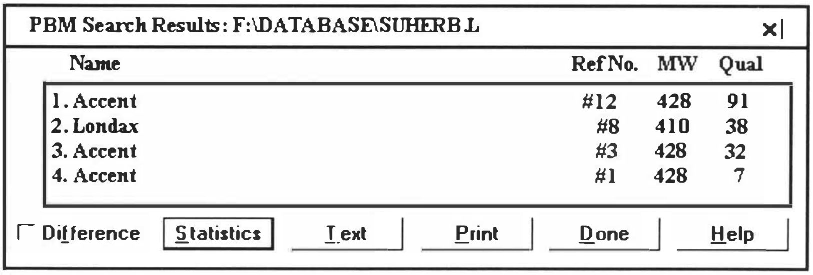
0.0506 → Express

0.0536 → Glean

0.0221 → Harmony

0.1278 → Londax

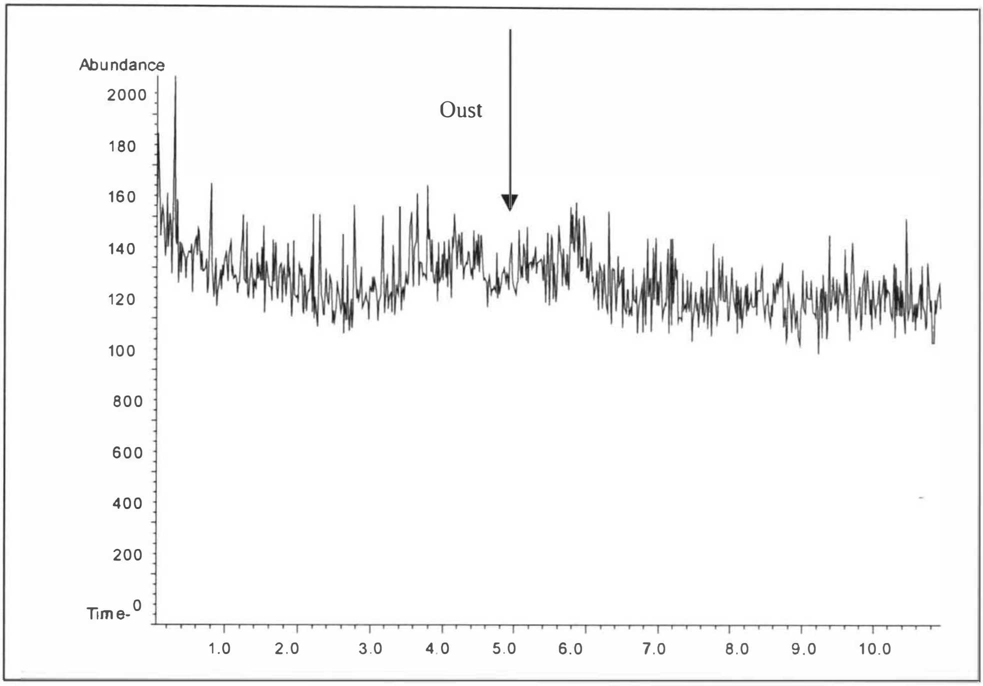
0.1372 → Oust



PBM Search Results: F:\DATABASE\SUHERB.L

Name	Ref No.	MW	Qual
1. Accent	#12	428	91
2. Londax	#8	410	38
3. Accent	#3	428	32
4. Accent	#1	428	7

Figure 54. Neural Network and HP-PBM Results for Accent After Silica Extraction.



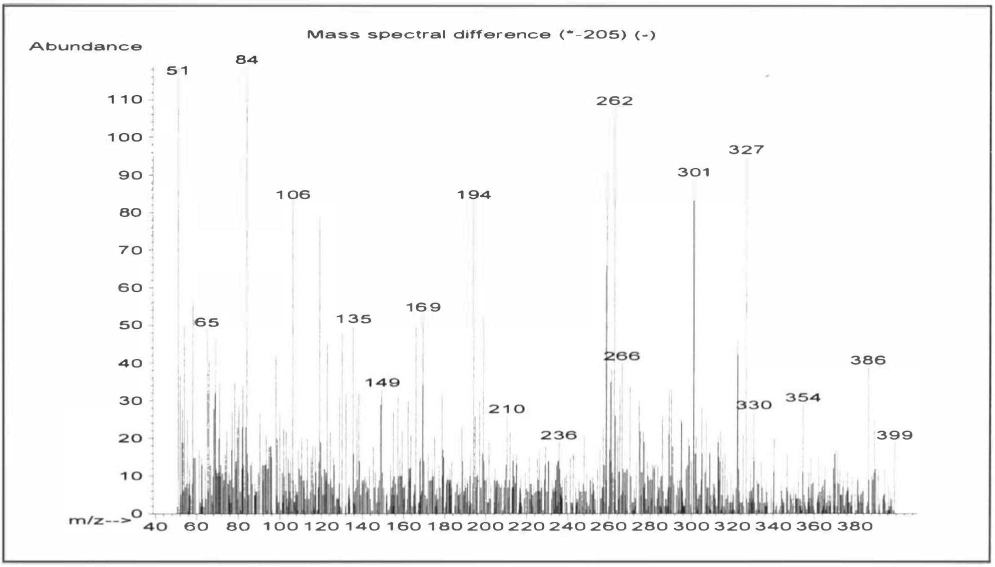


Figure 55. TIC and MS of Oust after Silica Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0048→ Accent

0.0779→ Ally

0.0189→ Classic

0.0909 → Express

0.0961→ Glean

0.0405→ Harmony

0.0020→ Londax

0.9239→ Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0967→ Accent

0.0259→ Ally

0.4669→ Classic

0.0007 → Express

0.2768 →Glean

0.0043→ Harmony

0.0002→ Londax

0.7183→ Oust

HP-PBM Result – "No hits found for this spectrum"

Figure 56. Neural Network and HP-PBM Results for Oust After Silica Extraction.

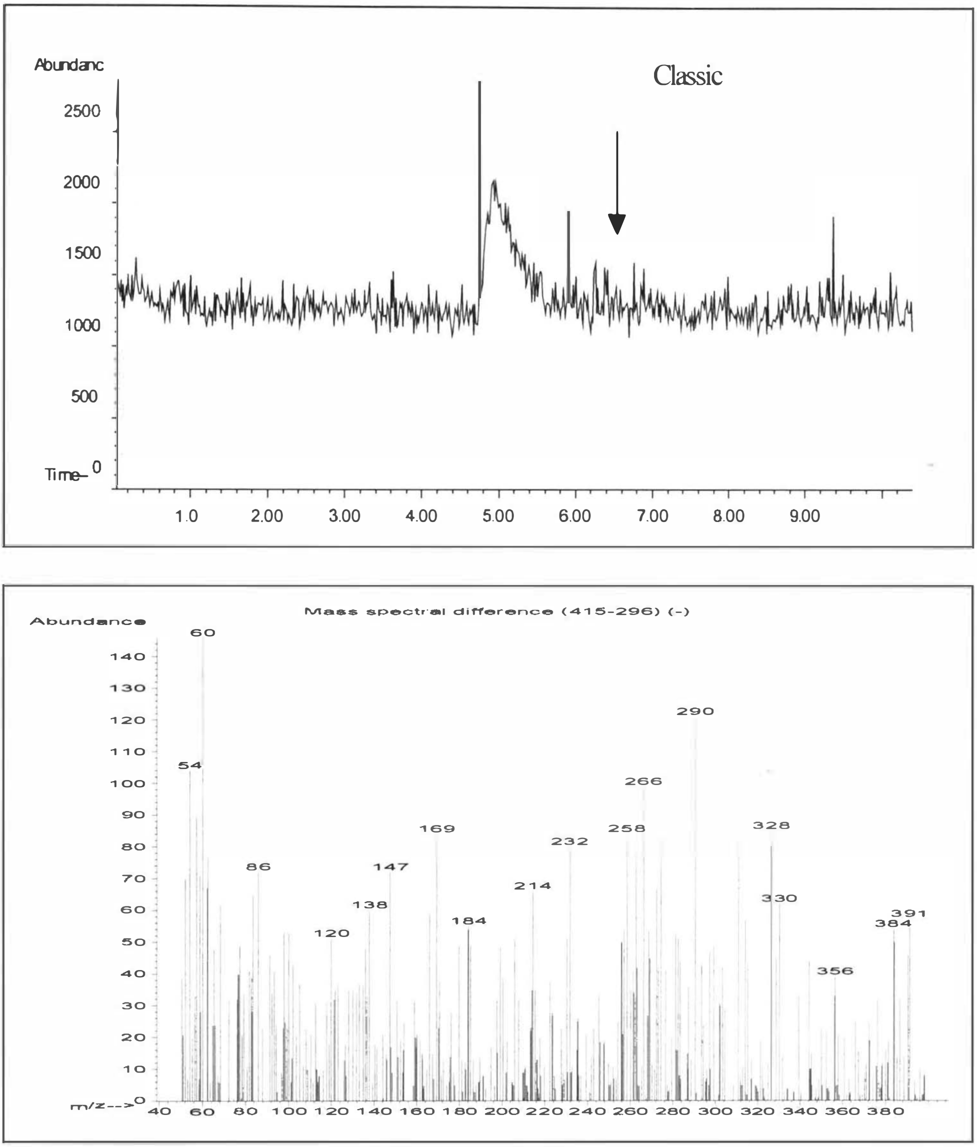


Figure 57. TIC and MS of Classic after Silica Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0069 → Accent

0.5602 → Ally

0.0052 → Classic

0.0027 → Express

0.5408 →Glean

0.2800 → Harmony

0.0015 → Londax

0.0004 → Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0496→ Accent

0.0052→ Ally

0.9124 →Classic

0.0254→ Express

0.0888→ Glean

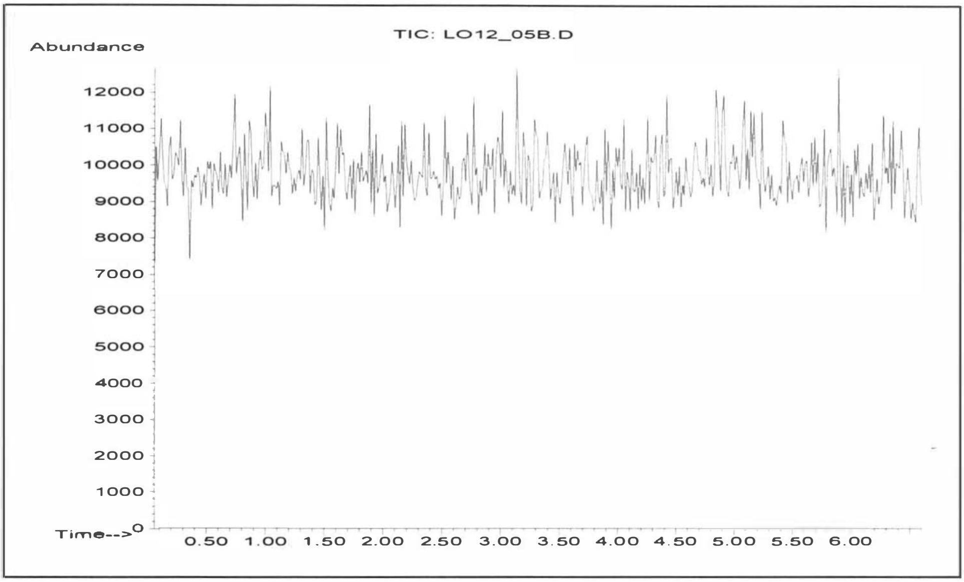
0.0122 →Harmony

0.0001→ Londax

0.0656→ Oust

HP-PBM Result – "No hits found for this spectrum"

Figure 58. Neural Network and HP-PBM Results for Classic After Silica Extraction.



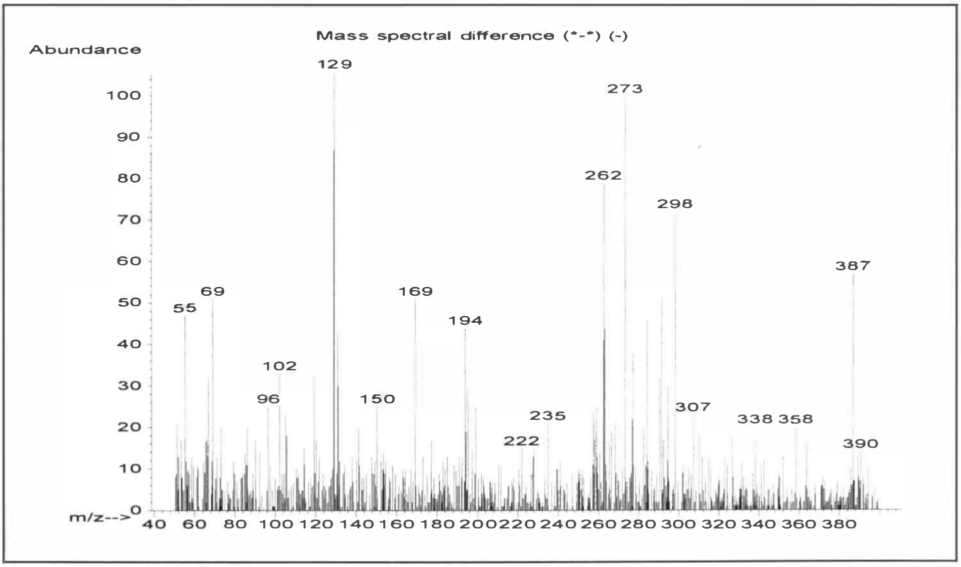


Figure 59. TIC and MS of Londax after Silica Extraction.

Neural Network Output Layer (40-training-sample ANN)

0.0838→ Accent
0.6152→ Ally
0.1368→ Classic
0.9850 → Express
0.1476→ Glean
0.0384→ Harmony
0.0003 → Londax
0.2581→ Oust

Higher values indicate the presence of the herbicide, lower values indicate the absence of the herbicide.

Neural Network Output Layer (8-training-sample ANN)

0.0599→ Accent
0.0046→ Ally
0.9418→ Classic
0.0258 → Express
0.0882→ Glean
0.0103→ Harmony
0.0003→ Londax
0.1599→ Oust

HP-PBM Result – "No hits found for this spectrum"

Figure 60. Neural Network and HP-PBM Results for Londax After Silica Extraction.

Table 11. Summary of Herbicides Identified by the Neural Networks and HP-PBM System for Individual Herbicides.

	Neural Network		HP-PBM
	8-training-sample	40-training-sample	
After C-18 Extraction	none	none	none
After SAX Extraction	Accent, Londax, Classic, Oust	Accent Londax Classic Oust	Accent, Londax, Classic
After Silica Extraction	Accent, Oust Classic	Accent, Oust	Accent

Table 12. Summary of Results of Neural Network and HP-PBM System for Mixture of Herbicides.

	Neural Network		HP-PBM
	8-training-sample	40-training-sample	
After C-18 Extraction	none	none	none
After SAX Extraction	Accent, Oust, Classic	Accent, Oust, Londax	Accent, Oust, Classic
After Silica Extraction	Accent Classic	Accent	Accent

The hidden node pruning algorithm however falls into the second category of ANN training, those developed to be applied to any class of compound. Three authors have developed ANNs that have general applicability. Penchev et al. studied the efficiency of ANNs by testing different architectures and different ways of normalizing the mass spectral data. [75] The results were rated with a Global Quality (GQ) indicator:

$$GQ = \frac{(IPc + IAc)}{nt} \qquad \text{Eq. 38}$$

where IPc is the number of species that were correctly indicated as present, IAc is the number of species that were correctly indicated as absent, and nt is the total number of species, the authors recorded scores ranging from 79.6% to 89.%, depending on the type of species studied. Using the same GQ equation, the 8-training-sample ANN in this research scored a 12/16 = 75.0%. The 40-training-sample ANN scored a 10/16 = 62.5% and the HP-PBM system scored a 8/16 = 50.0%. While these score are definitely lower, the data generated by the particle beam interface was not optimal, and definitely had a negative effect on the score.

In one paper mentioned previously, Eghbaldar [63] et al. developed a methodology to optimize a neural network and also utilized the GQ equation. Their scores ranged from 84.4% to 93.2% which are slightly better than the work done by Penchev. The authors also investigated the optimal size training set.

Starting with a training set with 500 examples, they increased the number by 100 until they determined that the optimal size was 2100 examples, a value much larger than the eight training samples used in this work.

Finally, Curry et al. also developed a neural network capable of detecting different functional groups from mass spectra. The MSnet involves a hierarchical system of several neural networks, each dedicated to different subnetworks were stacked one on another. His results were similar in quality to Eghbaldar's, but the authors did not give details about their choice of design or their methods of training.

CHAPTER 5

CONCLUSIONS

The work discussed in this dissertation covers the use of a feed-forward, back propagation ANN to analyze the less than optimal data produced by a liquid chromatograph/particle beam/mass spectrometer. The chemical system studied was a group of water soluble, heat labile sulfonylurea herbicides. A hidden node pruning algorithm was developed which used the significant eigenvalues of the matrix produced by the hidden layer of an ANN with excessive hidden nodes to determine the true rank of the hidden layer matrix.

The logic behind the hidden-node pruning algorithm is based on the fact that the number of eigenvalues and their associated eigenvectors are usually greater than the number of physically meaningful factors in a data set. Therefore, it is generally helpful to eliminate the least important eigenvalues/eigenvectors and regenerate a more compact data matrix containing only the significant eigenvectors. Two methods were used for selecting the number of significant eigenvalues. They were a variance ratio test, and a statistical F-test. The maximum value of these two tests was chosen as the true number of significant eigenvalues, and the hidden layer was resized to contain that number of nodes.

After re-sizing, the new network was then retrained using the standard back propagation method.

Determining the sum-squared error goal was accomplished by determining a standard error of calibration (SEC) and a standard error of prediction (SEP) for ANNs trained with sum-squared error goals ranging from 0.1 to 0.4. A statistical f-test was used to find the smallest sum-squared error goal where the SEP was not significantly higher than the SEC. The sum-squared error goal of 0.3 was chosen as the final value.

Because of the ability of using electron ionization, a library of MS spectra was developed for use by the ANN and the Hewlett Packard Probability Based Matching (HP-PBM) library searching system. Making a library of the mass spectra of the sulfonylurea herbicides, both the trained ANN and the HP-PBM system attempted to analyze data generated from an extraction procedure from soil of a mixture of four sulfonylurea herbicides

Using data from the LC/PB/MS, eight sulfonylurea herbicides were used as test compounds to determine the neural network ability for pattern recognition. The first task was the determination of the size of the training matrix. It was necessary to determine the number of mass/charge ratios needed to identify all eight herbicides. This was accomplished by examining the individual mass spectra and observing those m/z ratios which were characteristic to each herbicide. These were then combined into one array and contained 47 individual m/z ratios.

Next, two neural networks were developed. One used only one mass spectrum from each herbicide as a training set. This resulted in a training matrix with dimensions 47x8. The other used five mass spectra from each herbicide as a training set, resulting in a training set with dimensions 47x40. Once the network was trained, the mass spectra of the soil extract were used to identify the herbicides. Both individual herbicides and a four-herbicide mixture were extracted from soil and analyzed at three different steps in the process.

After the first extraction process, neither the neural networks nor the HP-PBM system could identify any of the herbicides present. This was due to the fact that the solution was very dilute and adulterated with many impurities from the soil.

After the second extraction, both neural networks were able to identify, Accent, Londax, Oust, and Classic from the herbicides that were extracted individually from the soil, while the HP-PBM system identified Accent, Londax, and Classic. The results of the analysis of the herbicides extracted from the mixture showed that Accent, Oust, and Classic were identified by both the 8-training-sample network and the HP_PBM system, while the 40-training-sample network identified Accent, Oust, and Londax.

Recovery was worse after the third extraction. This can in part be explained by the inefficient throughput of the particle beam. The HP-PBM system identified only one herbicide, with Accent as its one success. The best results were obtained by the 8-training-sample network, in which three of the four herbicides that were extracted individually from the soil were identified, missing only Londax. Of the herbicides in the mixture, the 8-training-sample network identified Accent and Classic. The 40-training-sample network identified Accent and Oust for the individual extraction, but only Accent in the mixture.

The results illustrate several points. The most surprising is that the network trained with more training samples did not outperform the network trained with fewer training samples. Perhaps an increase in samples from one to five per herbicide is not enough to increase the ability of the ANN to identify a compound, but if trained with even greater numbers an improvement could be noticed. Even at the cost of greater training time, it would be advantageous to increase the size of the training set if an improvement could be realized. As computers become more powerful, the training requirements of large training sets will not present barriers in terms of time and memory requirements.

The most useful result is the fact that neural network can be an aid to the analyst in the identification of mass spectra. While not providing flawless identification, neural networks can provide possibilities, especially when choosing from a limited sample set.

Further research in this area should be directed toward identifying the properties of the individual herbicides that could account for the varying levels of success in being identified by the ANN or the HP-PBM system. Also, other chemical systems than just the sulfonylurea herbicides should be investigated. Of course, the most obvious extension would be to use the neural network with mass spectra generated with an electrospray MS.

References

(1) Rodriquez, M.; Orescan, D. B. *Anal. Chem.* **1998,** *70*, 2710-2717.

(2) Beyer, E. M.; Duffy, M. J.; Hay, J. V.; Schlueter, D. D. *Herbicides: Chemistry Degradation and Mode of Action*; Marcel Dekker, Inc.: New York, NY, 1988.

(3) Sherma, J. *Anal. Chem.* **1995,** *67*, 1R-20R.

(4) Rao, V. B.; Rao, H. V. *C++Neural Networks and Fuzzy Logic*; Henry Holt and Company, Inc.: New York, 1993.

(5) Ciesielski, S.; Loomis, D. P.; Mims, S. R.; Auer, A. *Am. J. Pub. Health* **1994,** *84*, 446.

(6) Forastiere, F.; Quercia, A.; Miceli, M.; Settimi, L.; Terenzoni, B.; Rapiti, E.; Faustini, A.; Borgia, P.; Cavariani, C. A. *Scandinavian Journal of Work Environment & Health* **1993,** *19*, 382.

(7) Foulke, J. E. In *FDA Consumer,* 1993.

(8) Council(NRC), N. R. *Pesticides In The Diet of Infants and Children* , National Academy Press: Washington, DC, 1993.

(9) McMahon, B. M.; Hardin, N. F. ; ., Ed.; USDA, 1994.

(10) Fletcher, J. S., Pfleeger, T.G., Ratsch, H.C., and Hayes, R. *Env. Tox. and Chem.* **1996,** *15*, 1189-1196.

(11) Fletcher, J. S., Pfleeger, T.G., and Ratsch, H.C *Environ. Sci. Technol.* **1993,** *27*, 2250-2252.

(12) McNally, M. E.; Wheeler, J. R. *J. Chromatogr.* **1988,** *435*, 63-71.

(13) Prince, J. L.; Guinivan, R. A. *J. Agric. Food Chem.* **1988,** *36*.

(14) Nilve, G.; Stebbins, R. *Chromatographia* **1991,** *32*.

(15) Kelley, M. M.; Zahnow, E. W.; Petersen, W. C.; Toy, S. T. *J. Agric. Food Chem.* **1985,** *33*, 962-965.

(16) Shalaby, L. M.; Bramble, F. Q.; Lee, P. W. *J. Agric. Food Chem.* **1992,** *40*, 513-517.

(17) Shalaby, L. M.; George, S. W. *Liquid Chromatography/Mass Spectrometry, Applications in Agriculture, Pharmaceutical, and Environmental Chemistry*; American Chemical Society: Washington, D.C., 1990.

(18) Howard, A. L.; Taylor, J. *J. Chromatogr. Sci.* **1992,** *30*.

(19) Pimentel, D.; Lehman, H. *The Pesticide Question*; Chapman and Hall: New-York, NY, 1993.

(20) Heftman, E. *Chromatography, 5th edition Part B: Applications*; Elsevier Science Publishers,: The Netherlands, 1992.

(21) Durand, G.; Sanchez Baeza, F.; Messeguer, A.; Barcelo, D. *Biomed. Environ. Mass Spectrom.* **1991,** *20*.

(22) Farran, A.; DePablo, J.; Barcelo, D. *J. Chromatogr.* **1991,** *562*, 163.

(23) Durand, G.; Barcelo, D.; Albaiges, J.; Mansour, M. *Chromatographia* **1990,** *29*, 120.

(24) Cairns, T.; Siegmund, E. G.; Stamp, J. J. *Rapid Commun. Mass Spectrom.* **1987,** *1*, 89.

(25) Creaser, C. S.; Stygall, J. W. *Analyst* **1993,** *118*, 1447.

(26) Behymer, T. D.; Bellar, T. A.; Budde, W. L. *Anal. Chem.* **1990,** *62*, 1686-1690.

(27) Ligon, W. V.; Dorn, S. B. *Anal. Chem.* **1990,** *62*, 2573-2580.

(28) Jones, T. L.; Betowski, L. D.; Lesnik, B.; Chiang, T. C.; Teberg, J. E. *Environ. Sci. & Tech.* **1991,** *25*, 1880-1884.

(29) Toth, J. P. ; Univ. of Florida: online Internet:, http://www.ifas.ufl.edu/~ipto/ifasms.htm, 1999.

(30) Cappiello, A.; Famiglini, G. *Anal. Chem.* **1995,** *67*, 412-419.

(31) Stanley, S. M. R.; Wilhelmi, B. S.; Rodgers, J. P.; Guthrie, A. *Biological Mass Spectrometry* **1994,** *23*, 483-491.

(32) Rosen, R. T., Khambhampati, I., Roinestad, K.S., Fukuda, E.K., Hartman, T.G., Lech, J., Hiserodt, R.D., Lippincott, R.L., Chicago, Illinois, May 29 - June 3, 1994; 880.

(33) Kim, I. S., Sasinos, F.I., Stephens, R.D., Wang, J., Brown, M.A. *Anal. Chem.* **1991,** *63*, 819-823.

(34) Bellar, T. A.; Behymer, T. D.; Budde, W. *J. Am. Soc. Mass Spectrom.* **1990,** *1*.

(35) Mattina, M. J. I. *J. Chromatogr.* **,** *549*, 237.

(36) Pace, C. M., Miller, D.A., Roby, M.R. ; U.S. Environmental Protection Agency, Environmental Systems, Testing Laboratory,: Las Vegas, NV, 1990.

(37) Betowski, L. D., Pace, C.M. Roby, M.R. *J. Am. Soc. Mass Spectrom.* **1992,** *3*, 823.

(38) Li, Z.; Cheng, Z.; Xu, L.; Li, T. *Anal. Chem.* **1993,** *65*, 393-396.

(39) Helrich, K. E. *Official Methods of Analysis of the Association of Official Analytical Chemists,* 15 ed.; Association of Official Analytical Chemists: Arlington, VA, 1990.

(40) Martin, M. ; Virginia Consolidated Laboratories, 1995.

(41) Adam, W.; Bronstein, I.; Edwards, B.; Engel, T.; Reinhardt, D.; Schneider, F. W.; Trofimov, A. V.; Vasil'ev, R. F. *J. Am. Chem. Soc.* **1996,** *118*, 10400-10407.

(42) Fu, L. *Neural Networks in Computer Intelligence*; McGraw-Hill, Inc.: New York, 1994.

(43) Burns, J. A.; Whitesides, G. M. *Chem. Reviews* **1993,** *93*, 2583-2601.

(44) Wythoff, B. J. *Chemo. Intell. Lab. Systems* **1993,** *18*, 115-155.

(45) *Neural Computing Architectures*; The MIT Press: Cambridge, Massachusetts, 1989.

(46) Mukesh, D. *J. Chem. Ed.* **1996,** *73*, 431-433.

(47) *Parallel Algorithms for Machine Intelligence and Vision*; Springer-Verlag: New York, 1990.

(48) Despagne, F.; Massart, D. *Chemo. Intell. Lab. Systems* **1998,** *40*, 145-163.

(49) Zhang, L., Jiang, J., Liu, P., Liang, Y., and Yu,R. *Anal. Chim. Acta* **1997,** *344*, 29-39.

(50) Walczak, B.; Massart, D. L. *Anal. Chim. Acta* **1996,** *331*, 177-185.

(51) Chen, S.; Cowan, C. F. N.; Grant, P. M. *IEEE Transactions on Neural Networks* **1991,** *2*, 302-309.

(52) Harrington, P. *Anal. Chem.* **1993,** *65*, 2167-2168.

(53) Cross, S. S.; Harrison, R. F.; Kennedy, R. L. *Lancet* **1995,** *346*, 1075-1079.

(54) Eberhart, R. C.; Dobbins, R. W. *Neural Networks PC Tools*; Academic Press, Inc.: San Diego, CA, 1990.

(55) Rumelhart, D. E.; McClelland, J. L. *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol. 1: Foundations.*; MIT Press,: Cambridge, MA, 1986.

(56) Wang, J.; Jiang, J.; Yu, R. *Chemo. Intell. Lab. Systems* **1996,** *34*, 109-115.

(57) Burns, J. A.; Whitesides, G. M. *Chem. Reviews* **1993, 93**, 2583-2601.

(58) Cirovic, D. *Trends in Anal. Chem.* **1997,** *16*, 148-155.

(59) Smits, J. R. M.; Schoenmakers, P.; Stehmann, A.; Sijstermans, F.; Kateman, G. *Chemo. Intell. Lab. Systems* **1993,** *18*, 27-39.

(60) Gemperline, P. J. Long, R.L.; and Gregoriou, V.G. *Anal. Chem.* **1991,** *63*, 2313-2322.

(61) Long, J. R. Gregoriou, V.G.; Gemperline, P.J. *Anal. Chem.* **1990,** *62*, 1791-1797.

(62) Eghbaldar, A.; Forrest, T. P.; Cabrol-Bass, D. *Anal. Chimi. Acta* **1998,** *359*, 283-301.

(63) Massart, D. L.; Vandeginste, B. G. M.; Deming, S. N.; Michotte, Y.; Kaufman, L. *Chemometrics: a textbook*; Elsevier Science Publishers B V: Amsterdam, 1988.

(64) Malinowski, E. R. *Factor Analysis in Chemistry*, Second ed.; John Wiley & Sons, Inc.: New York, 1991.

(65) Kramer, R. *Chemometrics Toolbox User's Guide*; The MathWorks, Inc.: Natick, Massachusetts, 1993.

(66) Peel, C.; Willis, M. J.; Tham, M. T. *Application of Neural Networks to Modelling and Control*, 1st. ed.; Chapman & Hill: London, 1993.

(67) Gonzalez, A. G. *Anal. Chim. Acta* **1998,** *360*, 227-241.

(68) Goodnough, D. B.; Marek, L. J.; Orescan, D. B.; Babicki, W.; Powley, C. R.; Duffy, M., J. , St. Paul, Minnesota, April 6-10 1998; E.I. DuPont de Nemours and Company DuPont Agriculture Products Global Technology Division; 62.

(69) Faber, K.; Kowalski, B. *Anal. Chim. Acta* **1997,** *337*, 57-71.

(70) Box, G. E. P.; Hunter, W. G.; Hunter, S. J. *Statistics for Experimenters*; John Wiley & Sons: New York, 1978.

(71) Demuth, H.; Mark, B. *Neural Network Toolbox for Use with Matlab*; The MathWorks, Inc.: Natic , Mass, 1994.

(72) Eghbaldar, A.; Forrest, T. P.; Cabrol-Bass, D.; Cambon, A.; Guigonis, J. M. *J. Chem. Info. Comp. Sci.* **1996,** *36*, 637-643.

(73) Goodacre, R.; Neal, M. J.; Kell, D. B. *Anal. Chem.* **1994,** *66*, 1070-1085.

(74) Gmehling, J. In *Computers in Chemistry*; Springer-Verlag: Berlin, 1991.

(75) Penchev, P.; Argirov, O.; Andreev, G. *Analytical Laboratory* **1994,** *3*, 23.

(76) Hewlett-Packard Co.: Palo Alto, California, 1993.

APPENDICES

APPENDIX I.

COMPUTER PROGRAMS

nnjoe.m

This program is written as a Matlab m-file for Matlab version 4.0 and it directs the creation of the neural network. It asks for the number of hidden neurons initially desired in the network, the type of transfer function in the hidden layer, the frequency of progress displays, the maximum number of epochs, the error goal, the learning rate, and the momentum constant.

It then re-seeds the random number generator and calls the Matlab function jptrnbp.m, which returns values for the weights and biases developed for the network after training for 1500 epochs. It also returns the matrix "hidlay", which is the matrix produce by the hidden layer with an excessive number of hidden nodes.

It next passes "hidlay" to the Matlab function "hidnode.m" which determines the number of significant eigenvalues in "hidlay." It then resizes the weights and bias and calls the Matlab function "trainbp.m", which is the standard Matlab function to perform back-propagation training.

Finally it calls the Matlab function "secsep.m" which calculates the standard error of calibration and the standard error of prediction.

```
%nnjoe  FF ANN to find sulfonylurea herbicides

% Joseph M. Pompano
clf;
figure(gcf)
% setfsize(400,150);
echo on
clc
rand('seed',sum(100*clock))
%
=================================================================
%   nnjoe
%
=================================================================

%   INITFF   - Initializes feed-forward networks up to 3 layers.
%   TRAINBPX - Trains a feed-forward network with faster back propagation.
%   trainlm  - Trains a feed-forward network with Levenberg-Marquardt
%   SIMUFF   - Simulates feed-forward networks up to 3 layers.
%   SULFONYLUREA SYSTEM IDENTIFICATION:
%   Using the above functions a feed-forward network is trained
%   to recognize sulfonylurea herbicides.
Pause
 % Strike any key to continue...
clc
%   DEFINING THE INPUT DATA
%   =======================

pause % Strike any key to define the network...
clc
%   DEFINING THE NETWORK
%   ====================

%   The network has a hidden layer and 8 logsig
%   neurons in its output layer

% Get the number of hidden neurons and type of neuron

S1 = input('Enter the number of hidden neurons');
echo off
```

```
ch = menu('Enter the type of transfer function:','hardlim','hardlims',...
'logsig','purelin','satlin','satlins','tansig');

if ch == 1
 transfun = 'hardlim';
end
if ch == 2,
transfun = 'hardlims';
end
if ch == 3
transfun = 'logsig';
end
if ch == 4
transfun = 'purelin';
end
if ch == 5
transfun = 'satlin';
end
if ch == 6
transfun = 'satlins';
end
if ch == 7
transfun = 'tansig';
end

[W1,b1,W2,b2] = initff(waffle2,S1,transfun,target,'logsig');

echo on

pause % Strike any key to train the network...
clc
%   TRAINING THE NETWORK
%   ==================================

df = input('Enter the frequency of progress displays (epochs)');

me = input('Enter the maximum number of epochs(normally 5000)');

eg = input('Enter the error goal of the network(normally 0.1)');
```

```
lr = input('Enter the learning rate of the network(normally 0.01) or mim. gradient
for LM(0.0001)');

mc = input('Enter the momentum constant of the network(normally 0.95) or initial
value for MU for LM(0.001)');

tp = [df me eg lr mc];

%   Training begins...please wait...
echo off

% training the network


[W1,b1,W2,b2,ep,tr,hidlay]=
jptrnbp(W1,b1,transfun,W2,b2,'logsig',waffle2,target,tp);


echo on

[U,S,Vt]=svd(hidlay);

[NFtest, NIND, NPV] = hidnode(S, hidlay)

N=max([NFtest, NIND, NPV])

I=diag(ones(c1))';

% resize b1 , W1 and W2
% nornalize b1 to make new W1 values real;
b1=b1(1:N,:);

NewV=Vt(1:N,:);


W1=(atanh(NewV))*pinv(waffle2);

W2=W2*hidlay*pinv(NewV);
```

```
[W1,b1,W2,b2,ep,tr,hidlay2,N]=
trainbp(W1,b1,transfun,W2,b2,'logsig',waffle2,target,tp);

[SEC, SEP]=SECSEP(waffle2,W1,b1,W2,b2,target,pred)
%  All Done
```

normal.m

This program is written as a Matlab m-file for Matlab version 4.0 and it normalizes data between the values of 0.1 and 0.9. It accepts a matrix A and returns a matrix of the same size but with each column normalized.

```
function [x] = normal(a)
%Normalizes  the data in a column-wise matrix.
%[X] = NORMAL(A)
%
% where:
%
% X     is the normalized data
%
% A     is the absorbance or concentration matrix
%
% "normal" expects the data to be organized column-wise per the MLR
 %  conventions.

%       by Joseph Pompano.
col=0;
row=0;


[i, j] = size(a);

        for col = 1:j
        mx=max(a(:,col));
        mn=min(a(:,col));
                for row=1:i
                x(row,col) = 0.8*(a(row,col) - mn)/(mx-mn)+0.1;
                end
        end
end
```

jptbp1.m

This program is written as a Matlab m-file for Matlab version 4.0 and it performs 1500 iterations of a feed-forward network with back propagation. Function jptbp1 is called from nnjoe and it is passed the weight matrix for the hidden and final layer, biases for the hidden and final layer, transfer function of the hidden layer, the training matrix of input vectors, the target matrix and training parameters.

```
% Function jptbp1
function [w1,b1,w2,b2,te,tr,hidlay]=jptbp1(w1,b1,f1,w2,b2,f2,p,t,tp)
% jptbp1 Train 2-layer feed-forward network w/back propagation.
%
%       [w1,b1,w2,b2,te,tr,hidlay,N] = JPTBP1(W1,B1,F1,W2,B2,F2,P,T,TP)
%         Wi - SixR weight matrix of ith layer.
%         Bi - Six1 bias vector of ith layer.
%         F  - Transfer function (string) of ith layer.
%         P  - RxQ matrix of input vectors.
%         T  - S2xQ matrix of target vectors.
%         TP - Training parameters (optional).
%       Returns:
%         Wi - new weights.
%         Bi - new biases.
%         TE - the actual number of epochs trained.
%         TR - training record: [row of errors]
%
%       Training parameters are:
%         TP(1) - Epochs between updating display, default = 25.
%         TP(2) - Maximum number of epochs to train, default = 1000.
%         TP(3) - Sum-squared error goal, default = 0.02.
%         TP(4) - Learning rate, 0.01.
%       Missing parameters and NaN's are replaced with defaults.
```

```
if nargin < 8,error('Not enough arguments.'),end

% TRAINING PARAMETERS
if nargin == 8, tp = []; end
tp = nndef(tp,[200 5000 0.1 0.01]);
df = tp(1);
me = tp(2);
eg = tp(3);
lr = tp(4);
df1 = feval(f1,'delta');
df2 = feval(f2,'delta');

% PRESENTATION PHASE
a1 = feval(f1,w1*p,b1);
a2 = feval(f2,w2*a1,b2);
e = t-a2;
SSE = sumsqr(e);

% PLOTTING FLAG
[r,q] = size(p);
[s2,q] = size(t);
plottype = max(r,s2) == 1;

% TRAINING RECORD
tr = zeros(1,me);
tr(1) = SSE;

% PLOTTING
clg
message = sprintf('JPTRMBP: %%g/%g epochs, SSE = %%g.\n',me);
fprintf(message,0,SSE)
if plottype
  h = plotfa(p,t,p,a2);
else
  h = ploterr(tr(1),eg);
end

%   determine the number of hidden nodes
```

```
for i=1:1500


 % CHECK PHASE
 if SSE < eg, i=i-1; break, end

 % BACKPROPAGATION PHASE
 d2 = feval(df2,a2,e);
 d1 = feval(df1,a1,d2,w2);

 % LEARNING PHASE
 [dw1,db1] = learnbp(p,d1,lr);
 [dw2,db2] = learnbp(a1,d2,lr);
 w1 = w1 + dw1; b1 = b1 + db1;
 w2 = w2 + dw2; b2 = b2 + db2;

 % PRESENTATION PHASE
 a1 = feval(f1,w1*p,b1);
 a2 = feval(f2,w2*a1,b2);
 e = t-a2;
 SSE = sumsqr(e);

 % TRAINING RECORD
 tr(i+1) = SSE;

 % PLOTTING
        if rem(i,df) == 0
        fprintf(message,i,SSE)
                if plottype
                delete(h);
                h = plot(p,a2);
                drawnow;
                else
                h = ploterr(tr(1:(i+1)),eg,h);
                end
        end
hidlay = a1;
end
```

hidnode.m

This program is written as a Matlab m-file for Matlab version 4.0 and it determines the number of significant eigenvalues present in a matrix that is passed to it along with the a matrix containing the square root of the eigenvalues of that matrix. It uses two different functions to accomplish this task.

The first method used is a statistical f-test to determine the number of significant factors. The degrees of freedom of the numerator is kept at one, while the denominator's degrees of freedom starts at n-1 and decreases by one for each step. What it really determines is if a given eigenvalue is significantly larger than the mean of all subsequent(higher rank) eigenvalues. The method uses the built-in Matlab function, "vlftest.m" which returns a matrix F of f-values where the first element of F is the f-value of the 2-way test between the first eigenvalue and the sum of all the other eigenvalue of the data matrix. Each subsequent element of matrix F is the f-value of the 2-way test with the subsequent eigenvalue and the sum of all the eigenvalues after it. Once the matrix of f-values is returned by vlftest.m, hidnode.m determines how many eigenvalues are significant by comparing the f-values returned to table of critical f-values at a significance level of 0.10.

The second method for finding the number of significant eigenvalues is the variance ratio test. This test makes use of the fact that the eigenvalues are considered to be variances, and the program determines how many eigenvalues are needed to account for greater than 95% of the variance.

```
function [NFtest,  NPV] = hidnode(S, hidlay)

% hidnode      Function to perform the Matlab vlftest on the hidden node
%              output of a feed-forward nerual network
%
%       [NFtest, NIND, NPV] = hidnode(S, hidlay)
%
%       Where:
%
%       NFtest     is the significant number of factors returned for F-test
%       NPV   is the significant number of factors returned for Variance Ratio > .95
%       S     is the matrix containing the square root of the eigenvalues
%       hidlay is the output matrix of the feed-forward network.
%       A maximum of 30 hidden nodes can be checked
%       by Joseph M. Pompano
% Calculate eigenvalues of hidlay
eignv=diag(S).*diag(S);

;

%  Use Percent Variance to find the number of factors
[rPV,cPV]=size(eignv);
total=sum(eignv);
  for i=1:rPV
    num(i)=sum(eignv(1:i));
    PV(i)=num(i)/total;
      if PV(i)>.95
        NPV=i;break;
      end
   end;

% Use F-test to find the number of factors.
% critical values for F-test using one degree of freedom in the numerator using
alpha=0.10
```

```
fcrit=[39.86; 8.53; 5.54; 4.54 ; 4.06 ; 3.78 ; 3.59 ; 3.46 ; 3.36 ; 3.29 ;
3.23 ; 3.18 ; 3.14 ;3.10 ; 3.07 ; 3.05 ; 3.03 ; 3.01; 2.99 ; 2.97  ;2.96 ;
2.95 ; 2.94 ; 2.93 ; 2.92 ; 2.91 ; 2.90 ; 2.89 ; 2.89 ; 2.88];

%ftest = vlftest(hidlay,eignv);
ftest = vlftest(hidlay,eignv);
[r,c] = size(ftest);
NFtest=0;
for i=r-1:-1:1;
if ftest(i)>fcrit(i);
NFtest=i-1;break;
end;
end
```

pcarsd.m

This program is written as a Matlab m-file for Matlab version 4.0 it calculates the residual standard deviation of the eigenvalues according to a formula developed by Malinowski [64] (second to the last line in the program) and store them in a matrix rsd.

```
function [rsd] = pcarsd(matrix)
%
%       by Joseph M. Pompano
%
%      PCARSD Residual Standard Deviation of Malinowski.
%
%       RSD = PCARSD(matrix)
%       Where:
%
%       RSD      is the residual standard deviation
%       matrix  is the data matrix
%
[vc, vl] = pca(matrix);
[a,b]=size(vl);
[r,c]=size(matrix);
for n =1:a-1
rsd(n)=sqrt(sum(vl(n+1:a))/(r*(c-n)));
end;
```

vlftest.m

This program is a standard Matlab m-file for Matlab version 4.0 written in the Chemometrics Toolbox, version 2.0. It calculates the f-values to be used in a statistical f-test.

```
function f = vlftest(a, vl, maxrank)

%       by Richard Kramer.
%
%       Copyright (c) 1988-1993 by The MathWorks, Inc
%
%      VLFTEST 2-way F-test for error eigenvalues.
%       F = vlftest(A, VL, MAXRANK)
%
%
%       Where:
%
%       F        is the vector containing the f values
%       A        is the absorbance matrix which generated the factors
%       VL       is the vector containing the eigenvalues
%       MAXRANK  is optional limit to the number of values calculated
%
%       The first element of F is the F value of the 2-way test between the
%       first factor and the next factor.  Each subsequent element of F is the
%       F value of the 2-way test with the subsequent factor and the one after
%       it.  Note that the last F value is always NaN because a 2-way
%       F-test cannot be performed on a single VL.

[i, j] = size(a);
k = min(i, j);
if nargin == 3,
        l = min(k, maxrank);
```

```
else,
        l = k;
end
f(k,1)=0;
for n = 1:(l-1);
        f(n,1) = sum(  (i - (n:(k-1))) .* (j - (n:(k-1))) );
        f(n,1) = f(n,1) / ( (i - n + 1) * (j - n + 1) );
        f(n,1) = f(n,1) * vl(n) / sum( vl((n+1):k));
end
f(l, 1) = NaN;
```

WRKDATA.xls

WRKDATA is programmed for the Microsoft Excel spreadsheet for Microsoft Office 97.

Column A - m/z ratio from the original mass spectrum of the selected mass spectrum

Column B - original abundances of the corresponding m/z ratios of column A

Column C - the truncated abundances of Column B

Column D - the sum of the abundances for a particular m/z ratio. See for example, truncated m/z ratio 90 has two abundances, 89 and 68. This produces a 157 in column D.

A	B	C	D
90.15	89	90	89
90.9	68	90	**157**

Column E - this column places a 0 for all abundances except the last one in a sum. See for example m/z ratio 200.

A	B	C	D	E
90.15	89	90	89	**0**
90.9	68	90	157	157

Column F - this displays the number of abundances for a given truncated m/z ratio.

Column G - this is the quotient of Columns E and F.

Column H - this is a copy of Column C except that all m/z ratios with 0 abundance is Column G are set to 0.

A	B	C	D	E	F	G	H
90.15	89	90	89	0	1	0	**0**
90.9	68	90	157	157	2	79	90

Column I- this is a copy of Column G

Column J is blank

Column K contains the 47 m/z ratios used in the neural network

Column L are the abundances corresponding to Column K

Example of Excel Program WRKDATA

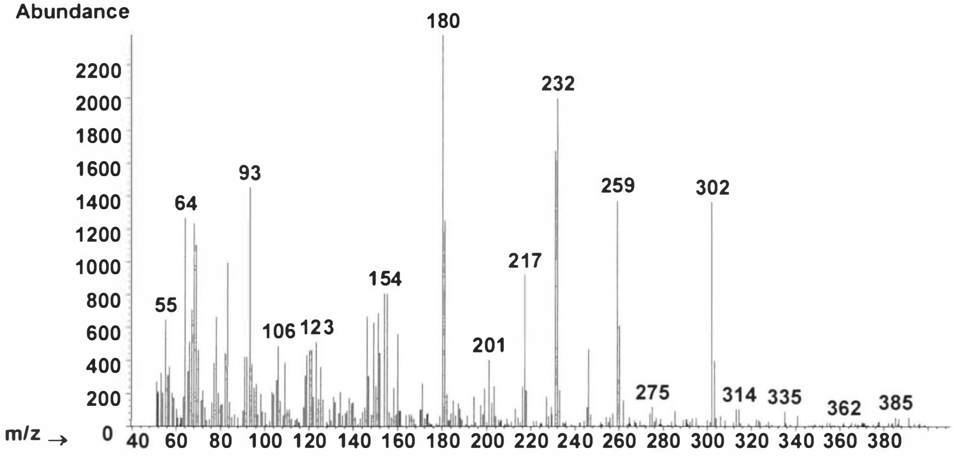
Program
WRKDATA.XLS
Joseph Pompano

A	B	C	D	E	F	G	H	I	J	K
		trunc	sum							Neural
Original	Data									Network
from MS										m/z
m/z	abund.	m/z	abund	abund	Count	avg	m/z	abund		
						abund				
90.15	89	90	89	0	1	0	0	0		104
90.9	68	90	157	157	2	79	90	79		105
92.1	155	92	155	155	1	155	92	155		106
93	130	93	130	130	1	130	93	130		108
94	42	94	42	0	1	0	0	0		110

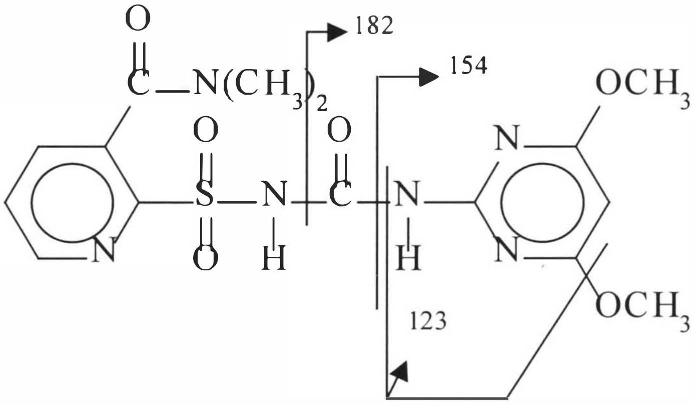
94.9	50	94	92	92	2	46	94	46	111
97.75	45	97	45	45	1	45	97	45	113
98	26	98	26	0	1	0	0	0	119
98.4	31	98	57	57	2	29	98	29	120
99.25	56	99	56	0	1	0	0	0	121
99.75	95	99	151	151	2	76	99	76	123
100.9	15	100	15	15	1	15	100	15	124
102.5	40	102	40	40	1	40	102	40	125
103.25	34	103	34	0	1	0	0	0	126
103.6	23	103	57	57	2	29	103	29	127
104	9	104	9	0	1	0	0	0	128
104.25	32	104	41	41	2	21	104	21	136
105.25	147	105	147	147	1	147	105	147	140
106	114	106	114	114	1	114	106	114	149
107.15	64	107	64	64	1	64	107	64	151
108.2	18	108	18	18	1	18	108	18	153
109.4	6	109	6	6	1	6	109	6	154
110.15	27	110	27	27	1	27	110	27	155
112.9	36	112	36	36	1	36	112	36	156
114	60	114	60	60	1	60	114	60	157
115.15	22	115	22	22	1	22	115	22	158
116	12	116	12	12	1	12	116	12	159
117.65	12	117	12	12	1	12	117	12	165
119.15	52	119	52	52	1	52	119	52	166
120	108	120	108	108	1	108	120	108	175
121	100	121	100	100	1	100	121	100	177
122.15	24	122	24	24	1	24	122	24	180
123	138	123	138	138	1	138	123	138	181
124.1	43	124	43	43	1	43	124	43	184
125.4	4	125	4	4	1	4	125	4	185
126.4	11	126	11	11	1	11	126	11	186
127.4	42	127	42	42	1	42	127	42	187
128	17	128	17	17	1	17	128	17	190
129	6	129	6	6	1	6	129	6	191
130.75	43	130	43	43	1	43	130	43	193
131.75	11	131	11	11	1	11	131	11	199
133.2	5	133	5	0	1	0	0	0	205
133.9	72	133	77	77	2	39	133	39	210
135.25	22	135	22	22	1	22	135	22	212
136.5	27	136	27	27	1	27	136	27	221
138.15	15	138	15	0	1	0	0	0	234
138.8	3	138	18	18	2	9	138	9	241
139.75	16	139	16	16	1	16	139	16	
140	16	140	16	16	1	16	140	16	
142.5	6	142	6	0	1	0	0	0	
142.9	6	142	12	12	2	6	142	6	

APPENDIX II.

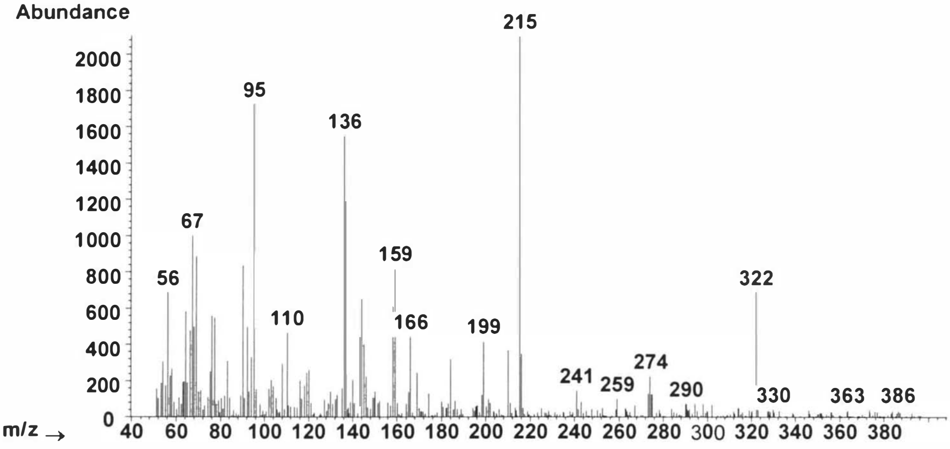
MASS SPECTRA AND FRAGMENT IONS OF SULFONYLUREA HERBICIDES



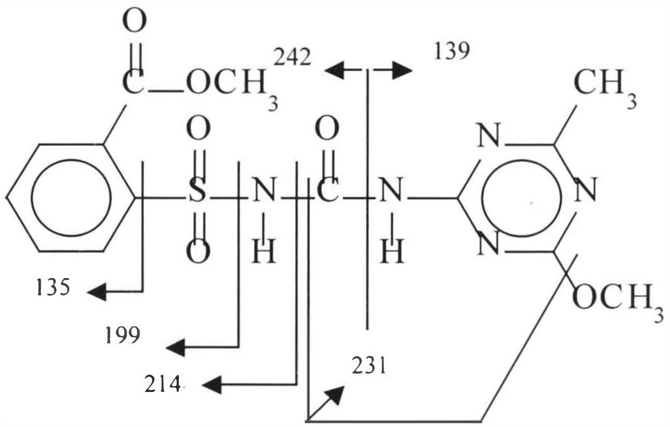
Accent



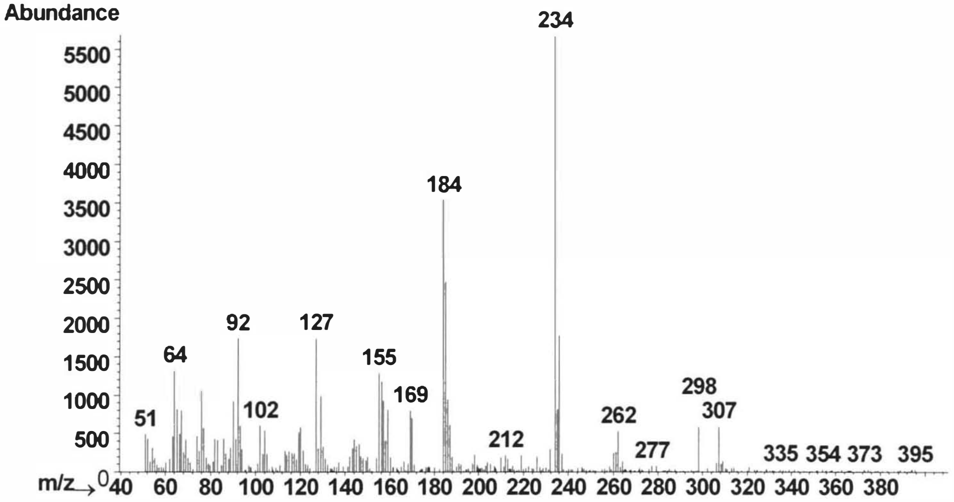
Accent - Nicosulfuron



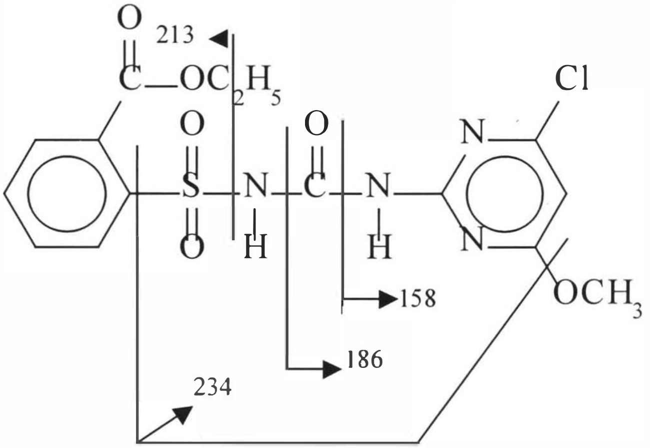
Ally



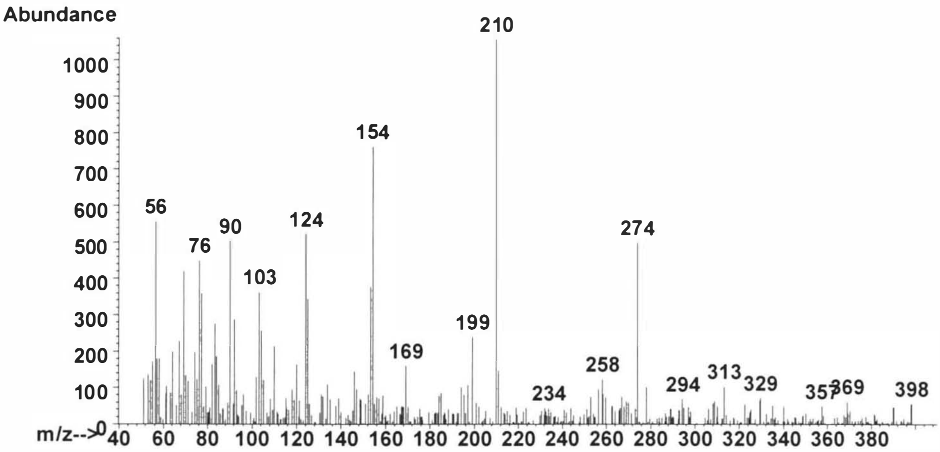
Ally - Metsulfuron methyl



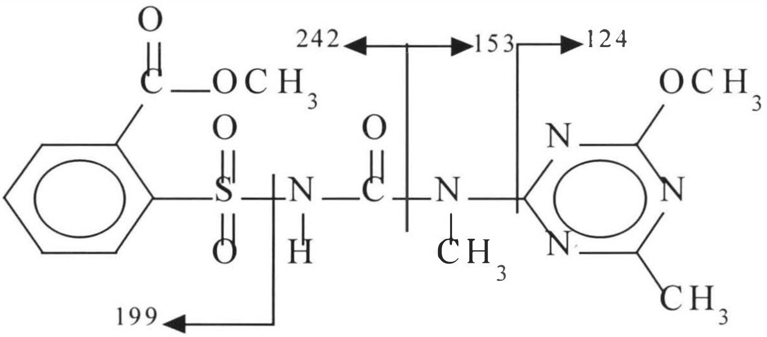
Classic



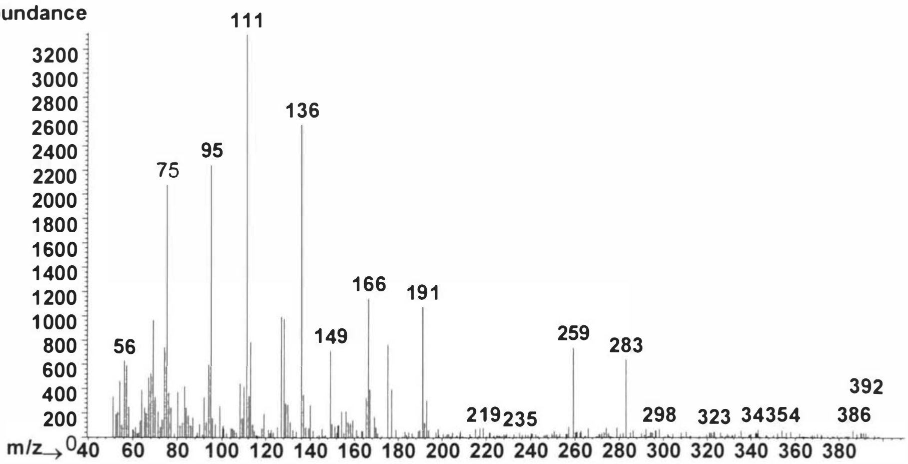
Classic - Chlorimuron ethyl



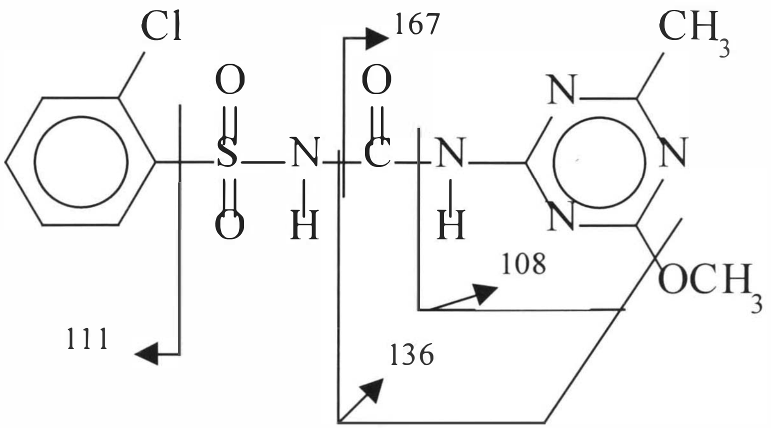
Express



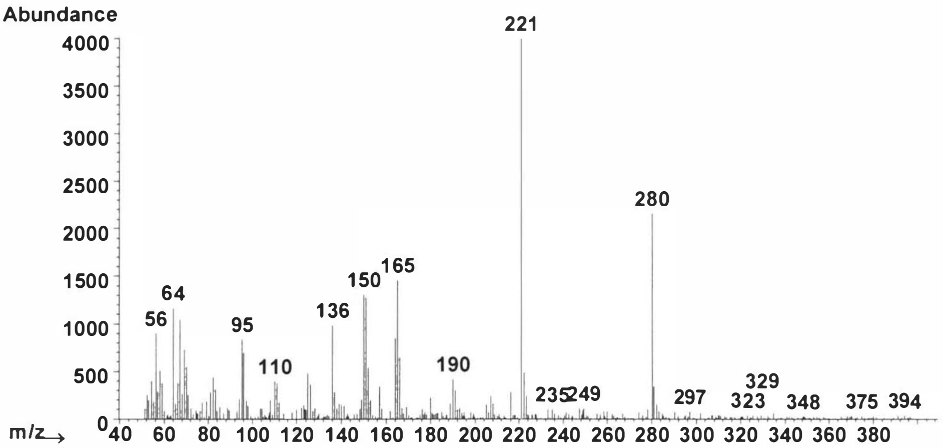
Express - Tribenuron methyl



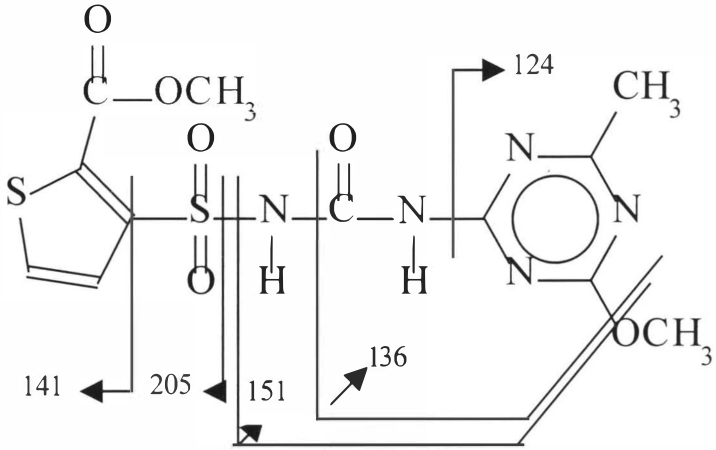
Glean



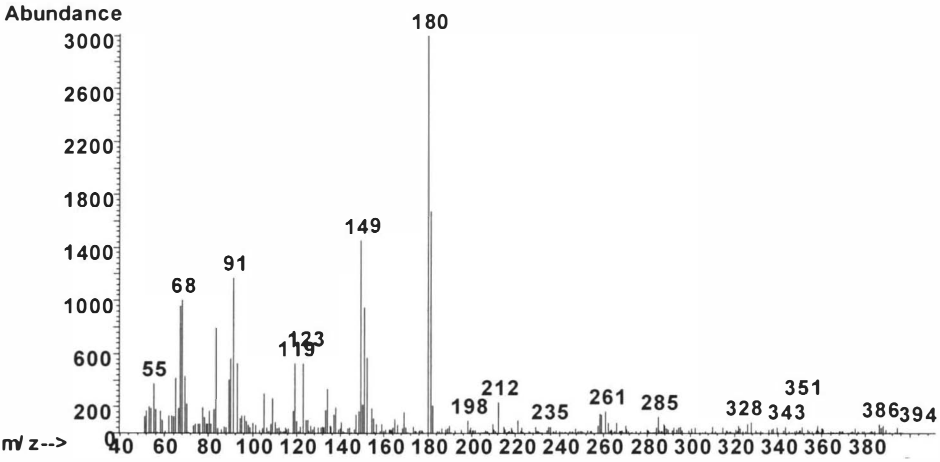
Glean - Chlorsulfuron



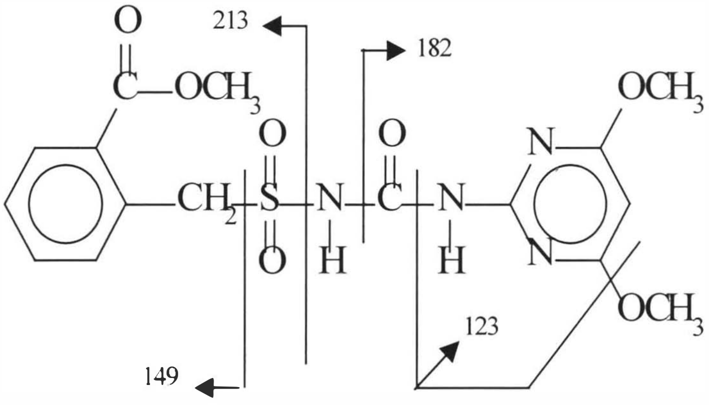
Harmony



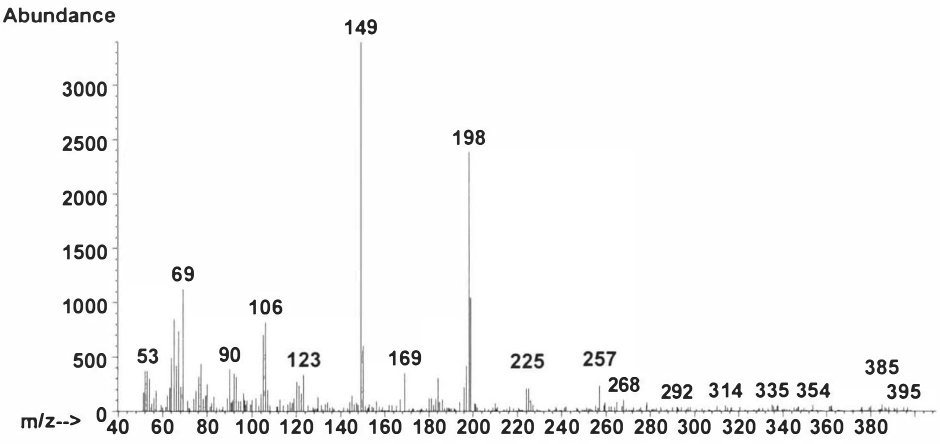
Harmony - Thifensulfuron



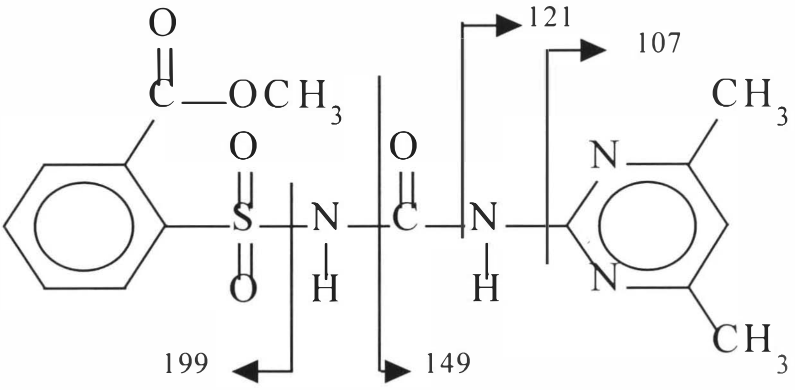
Londax



Londax - Bensulfuron methyl



Oust



Oust - Sulfometuron methyl

APPENDIX III

NORMALIZED TRAINING MATRIX

Normalized Training Matrix

M/z	Accent	Ally	Classic	Express	Glean	Harmony	Londax	Oust
104	0.1106	0.4615	0.5252	0.2897	0.1020	0.1244	0.1318	0.2356
105	0.2262	0.2563	0.2098	0.1536	0.1011	0.1115	0.1769	0.2537
106	0.2542	0.1559	0.1057	0.1106	0.1154	0.1126	0.1195	0.2475
108	0.1061	0.1717	0.1131	0.1197	0.1614	0.2424	0.1036	0.1148
110	0.1234	0.7271	0.1042	0.2176	0.1107	0.6648	0.1061	0.1023
111	0.1037	0.1193	0.1015	0.1025	0.9000	0.2848	0.1096	0.1030
113	0.1022	0.1039	0.1098	0.1025	0.2233	0.1087	0.1035	0.1011
119	0.5200	0.3674	0.2265	0.1595	0.1049	0.1056	0.3636	0.2071
120	0.3843	0.7538	0.4073	0.1679	0.1031	0.1046	0.1086	0.3094
121	0.3262	0.2286	0.3351	0.1074	0.1008	0.1027	0.1045	0.1760
123	0.2975	0.1060	0.1093	0.1125	0.1009	0.1049	0.1737	0.9000
124	0.1717	0.1032	0.1334	0.4739	0.1015	0.1297	0.1462	0.1694
125	0.2148	0.1334	0.1030	0.3290	0.1133	0.4255	0.1687	0.1048
126	0.1807	0.1218	0.1000	0.1113	0.1051	0.8101	0.1181	0.1016
127	0.1061	0.1049	0.4282	0.1021	0.2795	0.1932	0.1087	0.1011
128	0.1006	0.1032	0.1140	0.1005	0.2858	0.1162	0.1020	0.1009
136	0.1030	0.9000	0.1042	0.1133	0.3560	0.7278	0.1041	0.1084
140	0.1085	0.8691	0.1072	0.1080	0.3669	0.4784	0.1109	0.1050
149	0.8175	0.1060	0.1191	0.1025	0.1046	0.1113	0.9000	0.7870
151	0.3241	0.3076	0.1030	0.1015	0.1025	0.1376	0.1446	0.2132
153	0.1134	0.1109	0.1030	0.3620	0.1024	0.1105	0.1014	0.1021
154	0.5757	0.1084	0.1069	0.8715	0.1022	0.1037	0.3678	0.1032
155	0.5737	0.1014	0.2793	0.1726	0.1010	0.1067	0.3082	0.1009
156	0.1514	0.1014	0.2029	0.1018	0.1185	0.1000	0.1252	0.1018
157	0.1057	0.1000	0.1653	0.1023	0.1010	0.4129	0.1007	0.1016
158	0.1788	0.1119	0.3113	0.1002	0.1022	0.2319	0.1007	0.1034
159	0.1093	0.1046	0.3232	0.1007	0.1013	0.1127	0.1060	0.1000
165	0.1037	0.1306	0.2015	0.1012	0.1139	0.1427	0.1011	0.1053
166	0.1063	0.8726	0.1242	0.1028	0.4890	0.9000	0.1014	0.1039
175	0.1024	0.1018	0.1018	0.1005	0.2921	0.1127	0.1016	0.1016
177	0.1010	0.1067	0.1006	0.1005	0.1715	0.1000	0.1012	0.1002
180	0.9000	0.1000	0.1027	0.1524	0.1021	0.1027	0.3416	0.1027
181	0.6828	0.1032	0.1030	0.1128	0.1011	0.1042	0.2669	0.1014
184	0.1256	0.6161	0.9000	0.1018	0.1008	0.1037	0.1011	0.1893
185	0.1081	0.1246	0.6499	0.1017	0.1019	0.1028	0.1002	0.1071
186	0.1033	0.1105	0.3134	0.1025	0.1015	0.1000	0.1002	0.1048
187	0.1053	0.1018	0.2238	0.1000	0.1006	0.1043	0.1000	0.1014
190	0.1014	0.1000	0.1021	0.1000	0.1000	0.3459	0.1002	0.1021
191	0.1020	0.1021	0.1015	0.1008	0.2164	0.1223	0.1008	0.1005
193	0.1018	0.1011	0.1030	0.1025	0.1230	0.1036	0.1017	0.1014
199	0.1315	0.8989	0.1024	0.1672	0.1017	0.1055	0.1031	0.3573
205	0.1016	0.1014	0.1030	0.1015	0.1009	0.2426	0.1007	0.1007
210	0.1018	0.7004	0.2725	0.9000	0.1019	0.1024	0.1004	0.2888
212	0.1077	0.1478	0.1397	0.1171	0.1002	0.1018	0.1424	0.1037
221	0.1000	0.1014	0.1018	0.1008	0.1008	0.4301	0.1009	0.1018
234	0.1012	0.1070	0.1212	0.1026	0.1004	0.1021	0.1009	0.1009
241	0.1008	0.1365	0.1027	0.1530	0.1012	0.1017	0.1007	0.1447

Appendix IV

Calculation of Standard Error of Calibration in Matlab

The relative standard error of calibration (% SEC) is defined as in equation 18, where c_{mj} represents the mean value of herbicide j: $\hat{c}_{ij}$ represents the estimated value of the j-th herbicide for the i-th standard, c_{ij} represents the observed response, and n is the number of calibration standards.

$$\% \text{ SEC}_j = \frac{1}{n}(\sum_{i=1}^{n}(\hat{c}_{ij} - c_{ij})^2 \ ) / c_{mj})^{1/2} \times 100\% \qquad \text{Eq. 18}$$

The the mean value of herbicide j ,c_{mj}, equals (0.9 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1)/8 = 0.2 for each herbicide.

The Matlab function "simuff" calculates the output for a neural network where W1 and b1 are the weight matrix and bias matrix for the hidden layer, with a tansig transfer function, and W2 and b2 are the weight matrix and bias matrix for the output layer, with a logsig transfer function. The input matrix is named waffle2, and it is normalized between the values 0.1 toi 0.9 before processing. The Matlab version of Equation 18 is:

```
sqrt(sum(sum((target-a) .* (target -a))/.2))/8*100
```

From Matlab

```
---------------------------------------------------------------------------------------------
»     a = simuff(normal(waffle2),W1,b1,'tansig',W2,b2,'logsig')

a =

 Columns 1 through 8
   0.8934    0.0004    0.0003    0.0014    0.1074    0.1056    0.0846    0.0004
   0.0512    0.8844    0.0005    0.0037    0.0004    0.1066    0.1308    0.1082
   0.0000    0.0000    0.8970    0.1069    0.1015    0.1030    0.0015    0.0968
```

```
    0.0000    0.0062    0.1003    0.8831    0.1033    0.0000    0.1171    0.1071
    0.0875    0.0000    0.1000    0.1043    0.8959    0.1001    0.1127    0.0026
    0.1033    0.0070    0.0980    0.0244    0.1003    0.8835    0.1254    0.1193
    0.1014    0.1070    0.0010    0.1140    0.0928    0.1007    0.8963    0.0783
    0.0000    0.1036    0.1000    0.1108    0.0004    0.1054    0.0558    0.9013

» sqrt(sum(sum((target-a) .* (target -a))/.2))/8*100

ans =

   12.5000
```

Appendix V

Calculation of Standard Error of Prediction in Matlab

Form an input matrix of known values:

```
» pred=[ac1121, al1121,cl1121rt, ex1121rt, gl1121rt, ha1121rt, lo1121rt,
ou1121rt]

»     a = simuff(normal(pred),W1,b1,'tansig',W2,b2,'logsig')

a =

  Columns 1 through 8

    0.8844    0.0004    0.0003    0.0014    0.4131    0.1056    0.0846    0.0004
    0.0546    0.8748    0.0005    0.0052    0.0334    0.1066    0.1308    0.1082
    0.0000    0.0000    0.8970    0.0709    0.0001    0.1030    0.0015    0.0968
    0.0000    0.0056    0.1003    0.8523    0.0003    0.0000    0.1171    0.1071
    0.0772    0.0000    0.1000    0.0708    0.1012    0.1001    0.1127    0.0026
    0.1074    0.0073    0.0980    0.0227    0.0703    0.8835    0.1254    0.1193
    0.0928    0.1010    0.0010    0.1158    0.2069    0.1007    0.8963    0.0783
    0.0000    0.0928    0.1000    0.1088    0.0000    0.1054    0.0558    0.9013

» sqrt(sum(sum((target-a) .* (target -a))/.2))/8*100

ans =

    27.49
```

Appendix VI

Significance Testing of the SEC and the SEP

Excel Spreadsheet to calculate if the Standard Error of Prediction is significantly greater than the Standard Error of Calibration using Sum Squared Errors of 0.4 to 0.1

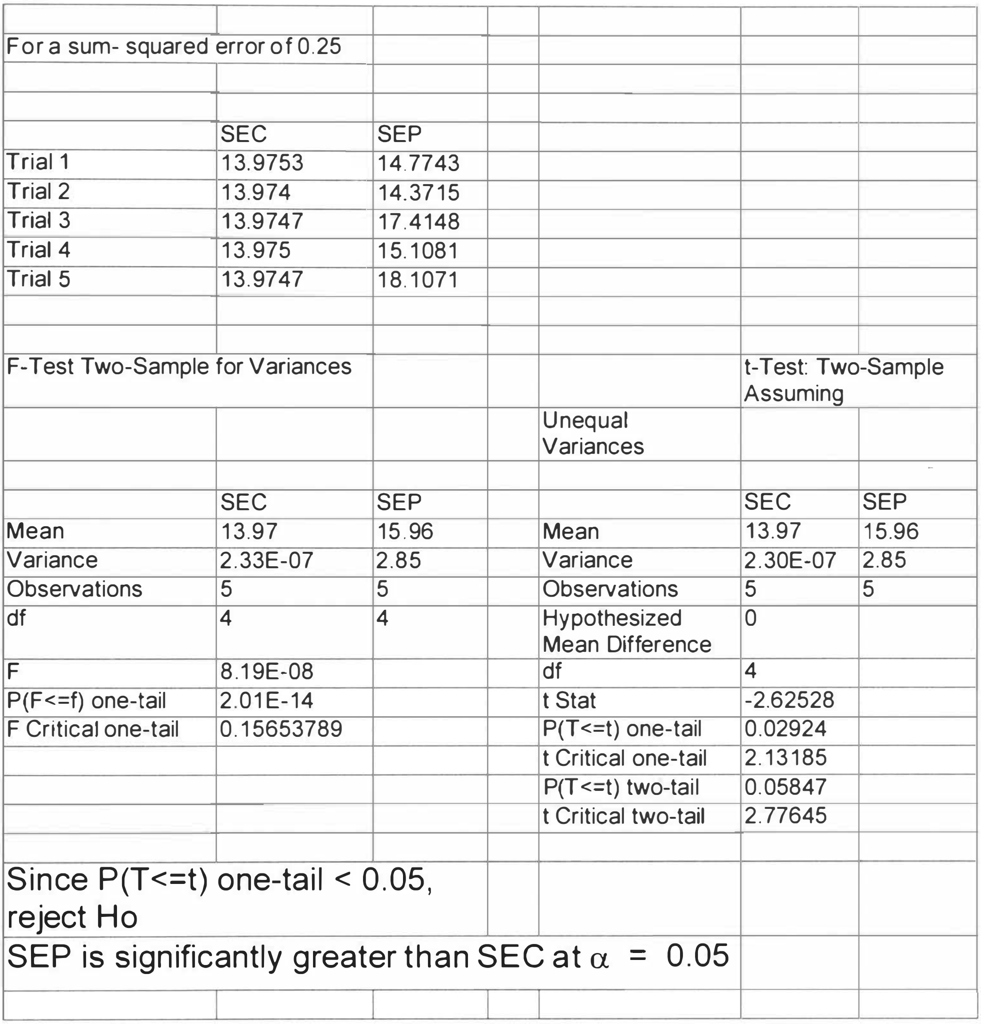
For a sum-squared error of 0.4

	SEC	SEP				
Trial 1	17.6706	17.9548				
Trial 2	17.6772	23.4651	Sum Squared Error =0.4			
Trial 3	17.6772	17.9419				
Trial 4	17.6757	32.8785				
Trial 5	17.6743	18.0593				
F-Test Two-Sample for Variances				**t-Test: Two-Sample Assuming Unequal Variances**		
	SEC	*SEP*			SEC	SEP
Mean	17.68	22.06		Mean	17.68	22.06
Variance	7.50E-06	42.21		Variance	7.50E-06	42.21
Observations	5	5		Observations	5.00	5.00
df	4	4		Hypothesized Mean Difference	0	
F	1.78E-07			df	4.00	
P(F<=f) one-tail	9.48E-14			t Stat	-1.51	
F Critical one-tail	0.16			P(T<=t) one-tail	0.10	
				t Critical one-tail	2.13	
				P(T<=t) two-tail	0.21	
				t Critical two-tail	2.78	
Since P(T<=t) one-tail > 0.05, fail to reject Ho						
SEP is not significantly greater than SEC at α = 0.05						

For a sum-squared error of 0.3

For a sum- squared error of 0.3						
	SEC	SEP				
Trial 1	15.3081	15.3914				
Trial 2	15.309	21.5109				
Trial 3	15.308	15.4014				
Trial 4	15.2788	16.3561				
Trial 5	15.308	19.6286				
F-Test Two-Sample for Variances		t-Test: Two-Sample Assuming				
Unequal Variances						
	SEC	SEP			SEC	SEP
Mean	15.30238	17.66		Mean	15.3024	17.66
Variance	0.00017	7.663		Variance	0.00017	7.6637
Observations	5	5		Observations	5	5
df	4	4		Hypothesized Mean Difference	0	
F	2.27E-05			df	4	
P(F<=f) one-tail	1.55E-09			t Stat	-1.90249	
F Critical one-tail	0.1565			P(T<=t) one-tail	0.06493	
				t Critical one-tail	2.13185	
				P(T<=t) two-tail	0.12987	
				t Critical two-tail	2.77645	
Since P(T<=t) one-tail > 0.05, fail to reject Ho						
SEP is not significantly greater than SEC at a = 0.05						

For a sum-squared error of 0.25



For a sum- squared error of 0.25						
	SEC	SEP				
Trial 1	13.9753	14.7743				
Trial 2	13.974	14.3715				
Trial 3	13.9747	17.4148				
Trial 4	13.975	15.1081				
Trial 5	13.9747	18.1071				
F-Test Two-Sample for Variances					t-Test: Two-Sample Assuming	
				Unequal Variances		
	SEC	SEP			SEC	SEP
Mean	13.97	15.96		Mean	13.97	15.96
Variance	2.33E-07	2.85		Variance	2.30E-07	2.85
Observations	5	5		Observations	5	5
df	4	4		Hypothesized Mean Difference	0	
F	8.19E-08			df	4	
P(F<=f) one-tail	2.01E-14			t Stat	-2.62528	
F Critical one-tail	0.15653789			P(T<=t) one-tail	0.02924	
				t Critical one-tail	2.13185	
				P(T<=t) two-tail	0.05847	
				t Critical two-tail	2.77645	
Since P(T<=t) one-tail < 0.05, reject Ho						
SEP is significantly greater than SEC at α = 0.05						

For a sum-squared error of 0.2

	SEC	SEP
Trial 1	12.5	14.6133
Trial 2	12.496	14.6734
Trial 3	12.5	17.9458
Trial 4	12.4997	22.9689
Trial 5	12.4997	13.2373

F-Test Two-Sample for Variances

	SEC	SEP
Mean	12.49908	16.68774
Variance	2.987E-06	15.32552
Observations	5	5
df	4	4
F	1.949E-07	
P(F<=f) one-tail	1.1391E-13	
F Critical one-tail	0.15653789	

t-Test: Two-Sample Assuming Unequal Variances

	SEC	SEP
Mean	12.4991	16.688
Variance	3E-06	15.325
Observations	5	5
Hypothesized Mean Difference	0	
df	4	
t Stat	-2.3925	
P(T<=t) one-tail	0.03748	
t Critical one-tail	2.13185	
P(T<=t) two-tail	0.07496	
t Critical two-tail	2.77645	

Since P(T<=t) one-tail < 0.05, reject Ho

SEP is significantly greater than SEC at α = 0.05

For a sum-squared error of 0.1

	SEC	SEP
Trial 1	8.8838	12.0664
Trial 2	8.8388	9.4883
Trial 3	8.8388	11.1047
Trial 4	8.8388	14.6863
Trial 5	8.8388	13.9506

F-Test Two-Sample for Variances

	SEC	*SEP*
Mean	8.8478	12.25926
Variance	0.000405	4.449894
Observations	5	5
df	4	4
F	9.1013E-05	
P(F<=f) one-tail	2.4844E-08	
F Critical one-tail	0.15653789	

t-Test: Two-Sample Assuming Unequal Variances

	SEC	*SEP*
Mean	8.8478	12.259
Variance	0.00041	4.4498
Observations	5	5
Hypothesized Mean Difference	0	
df	4	
t Stat	-3.61602	
P(T<=t) one-tail	0.01122	
t Critical one-tail	2.13185	
P(T<=t) two-tail	0.02244	
t Critical two-tail	2.77645	

Since P(T<=t) one-tail < 0.05, reject Ho

SEP is significantly greater than SEC at α = 0.05

Page 172 intentionally left blank

Appendix VII

The Hewlett-Packard Probability Based Matching Library Searching System

The following information is taken from the help file associated with the Data Analysis Module of the Hewlett-Packard Mass Spectrometer Software G1034C Version C.02.00 [76]

> PBM refers to the probability-based matching (PBM) algorithm, a library-search routine developed at Cornell University by Professor Fred McLafferty and co-workers.
>
> The algorithm uses a reverse search to verify that peaks in the reference spectrum are present in the unknown spectrum. Extra peaks in the unknown are ignored, thus allowing the analysis of a spectrum resulting from a mixture of compounds.
>
> Since not all mass-to-charge (m/z) values of a mass spectrum are equally likely to occur, the PBM algorithm uses both the mass and abundance values to identify the most significant peaks in the reference spectrum. When a spectrum is added to a library, these peaks are used to generate a condensed reference spectrum that is used by the PBM search routine.
>
> A pre-filter within the search routine then assigns a significance to each of the peaks in the unknown spectrum and uses these to find the most probable matches in the condensed reference library. The selected condensed spectra are then compared, using the reverse search described above, with the complete unknown spectrum. The pre-filter immediately eliminates approximately 95% of the compounds in the database and greatly speeds up the search (when using the default strategy parameters).
>
> Significance
> The PBM search assigns significance to each mass peak based on both the m/z ratio and its abundance. Those peaks with the greatest significance are then used for the matching.
>
> The significance of a mass peak is determined by two values, U, the uniqueness, and A, the abundance. Both the U and A values were developed by McLafferty and his co-workers from statistical studies of 79,650 spectra of 67,128 compounds.

Uniqueness

Certain masses (m/z) are more likely to occur in a mass spectrum than others. For example, m/z 43 is much more common than m/z 343. Each mass is assigned a uniqueness value between 0 and 12, the most frequently found masses (m/z 29, 39, 41, 43) being assigned a value of zero.

Abundance

The value of A is assigned based on the relative abundance of a mass in the spectrum. The higher the relative abundance, the greater the A value; the assigned values are -3, -2, -1, 0, 1, 2, 3, 4, and 5.

The significance calculations, which are the basis of the PBM searching and condensing algorithms, are used in the following ways:

- To define peak significance when selecting the 15 to 26 peaks that are stored in a condensed reference spectrum.

- In the pre-filtering algorithm so that only spectra whose most significant peaks are similar to those of the unknown are selected for comparison.

As one of the factors for evaluating the similarity of reference spectra to an unknown spectrum.

In most cases, the PBM search will retrieve a match from the database with a high match quality, and the unknown can be considered identified with a high degree of confidence. However, no search routine, no matter how sophisticated, can provide a conclusive identification 100% of the time. Consider some of the factors affecting the match quality:

The type of instrument used to collect the spectra of the unknown and the reference

The experimental conditions used to collect the spectra of the unknown and reference

Choice of spectrum for background correction

Strategy parameters used during the PBM search

The quality of the spectra in the database

Several of the factors are related to actual collection of the data. The most important aspect of a successful identification by library search is that both the reference and the unknown spectra be high quality.

For example, if the signal-to-noise ratio of the mass spectrum is too low, no amount of background correction or changes in PBM strategy parameters will improve the chances of a good match. Also, a spectrum obtained using a GC as an inlet to the mass spectrometer may be much different than one obtained using a direct insertion probe.

That is why it's a good idea to perform the search on a user-created library with reference spectra obtained on the same instrument using the same conditions.

When you are acquiring mass spectral data for identification by PBM, it is recommended that Standard Autotune be used and that the scan threshold be set to 500 for an HP 5971. This permits the scanning algorithm to detect the ions of low relative intensity that may be present in the reference spectra and expected by PBM. Also, the low end of the scan range should be set to match the low mass limit of the spectra in the library. When you are acquiring data, the goal should be to acquire all ions in the compound with relative intensity between 0.5% and 100%.

Because many factors affect the match quality and ordering of the compounds in the hit list, the list should be viewed as an interpretative guide to the unknown's identity. You should not assume that the match listed first is the one and only correct answer. In the final analysis, it is the chemist's responsibility to determine whether the match identity is correct by using the PBM results in conjunction with other information. For example, graphical comparison of the unknown's mass spectrum with that of an authentic sample, knowledge of the sample's history, and other pertinent information should be considered.

Vita

[redacted]