Evolutionary Computing Techniques to aid the Acquisition and Analysis of Nuclear Magnetic Resonance Data

Helen Frances Gray

PhD

Department of Computing
School of Informatics
City University
London

January 2007

Table of Contents

List of Tables

List of Figures

Acknowledgements

Some of the work was carried out at Aarhus University, Denmark. Facilities were provided by the Department of Computer Science (DAIMI) and guidance from Professor Brian Mayoh.

Later work was carried out at the Gray Cancer Institute, Northwood, UK with kind permission of Professor Barry Michael. The Magnetic Resonance hardware and software were operated by Dr Ross Maxwell, who also provided much guidance.

My supervisor at City University, Dr Peter Smith, has helped me in many ways.

Sheila Munton, a librarian at City University, was very helpful.

Financial support for the collection of the brain tumour data was provided by the Karen Elise Jensens Fund; CICYT SAF93-0582-CO2-01 and SAF96- 0147; Conselleria de Sanitat, Generalitat de Catalunya, Spain; EC Concerted Action Biomed-I PL920432. Neurosurgery services were provided at the participating hospitals in Barcelona and Madrid, Spain.

I would like to thank my family for their support.

Declaration

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Abstract

Evolutionary computation, including genetic algorithms and genetic programming have taken the ideas of evolution in biology and applied some of the characteristics to problem solving. The survival of the fittest paradigm allows a population of candidate solutions to be modified by sexual and asexual reproduction and mutation to come closer to solving the problem in question without the necessity of having prior knowledge of what a good solution looks like.

The increasing importance of Nuclear Magnetic Resonance Spectroscopy in medicine has created a demand for automated data analysis for tissue classification and feature selection. The use of artificial intelligence techniques such as evolutionary computing can be used for such data analysis.

This thesis applies the techniques of evolutionary computation to aid the collection and classification of Nuclear Magnetic Resonance spectroscopy data. The first section (chapters one and two) introduces Nuclear Magnetic Resonance spectroscopy and evolutionary computation and also contains a review of relevant literature. The second section focuses on classification. In the third chapter classification into two classes of brain tumours is undertaken. The fourth chapter expands this to classify tumours and tissues into more than two classes. Genetic Programming provided good solutions with relatively simple biochemical interpretation and was able to classify data into more than two classes at one time. The third section of the thesis concentrates on using evolutionary computation techniques to optimise data acquisition parameters directly from the Nuclear Magnetic Resonance hardware. Chapter five shows that Genetic Algorithms in particular are successful at suppressing signals from solvent while chapter six applies these techniques to find a way of enhancing the signals from metabolites important to the classification of brain tumours as found in chapter three. The final chapter draws conclusions as to the efficacy of evolutionary computation techniques applied to Nuclear Magnetic Resonance Spectroscopy.

List of Abbreviations

ADF	Automatically Defined Function
AI	Artificial Intelligence
Ala	Alanine
CH_2	Creatine
CH_3	Creatine
CT	Computerised Tomography
DMSO	Dimethylsulphoxide
EC	Evolutionary Computation
EP	Evolutionary Programming
ES	Evolutionary Strategy
FID	Free Induction Delay
FPGA	Field Programmable Gate Arrays
GA	Genetic Algorithm
GAOT	Genetic Algorithm Optimisation Toolbox
Gln	Glutamate
Glu	Glutamine
GP	Genetic Programming
^1H	Proton, hydrogen atom with zero neutrons
kNN	k Nearest Neighbour where k is a number
LDA	Linear Discriminant Analysis
MHz	Megahertz
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MRS	Magnetic Resonance Spectroscopy
MRSI	Magnetic Resonance Spectroscopic Imaging
NAA	N-acetyl-aspartate
NLM	Non-Linear Mapping
NMR	Nuclear Magnetic Resonance
NMV	Net Magnetic Vector
NN	Neural Network
PC	Principal Component

PCA	Principal Component Analysis
PET	Positron Emission Tomography
PLS	Partial least Squares
ppm	parts per million
RF	Radio-Frequency
RFi	Raw Fitness
SF	Standardised Fitness
SLDA	Standard Linear Discriminant Analysis
SNR	Signal to Noise Ratio
SOM	Self Organising Map
TE	Echo Time
TMS	Tetramethylsilane
TSP	Trimethylsilyl-2,2,3,3-tetradeutero-propionate sodium

Chapter 1

Introduction to Nuclear Magnetic Resonance

1.1 Introduction

Nuclear Magnetic Resonance (NMR) is a technique used both in chemistry and medical diagnosis as a non-invasive, non-toxic method of examining the structure and features of a sample. It is based on the idea of using magnetism to align nuclei of atoms into a direction which will give rise to a magnetic force which can be measured. The collection of data can concentrate on spatial features which will allow an image to be constructed, or on the chemical composition of the sample which will give rise to a spectrum. The latter technique is used in chemistry, as well as in medical situations and the terms NMR and Magnetic Resonance Spectroscopy (MRS) would be used to describe the techniques. Imaging is the technique more used in clinical situations and here the word nuclear is often dropped from the description because of negative connotations with radioactivity (which is not required for NMR) leaving the titles Magnetic Resonance (MR) and Magnetic Resonance Imaging (MRI) as those most commonly used. The terms MR and NMR are interchangeable and the use of one or the other usually depends only on context. There are many books available explaining MR (Schild 1990; Westbrook and Kaut 1993, 1998; Gadian 1995; Hornak 1997-1999)

MRI in a clinical setting can be used as a diagnostic tool, in examining patients for the presence and size of tumours or in other soft tissue investigations. In imaging, the examination is typically of a slice, or series of slices, through a subject. The results are displayed in a two-dimensional grid, with different tissues distinguished by different intensities on a grey-scale. It has advantages over both X-rays and Positron Emission Tomography (PET) as imaging techniques partly because of the lack of toxicity i.e. exposure to ionising radiation. However, X-rays offer better

results in looking at bone mass and PET has advantages in tracer studies (e.g. for radiolabelled drugs binding to specific molecular receptors). The lack of toxicity and invasiveness also allows the possibility of repeat investigations which may have advantages in evaluating treatment outcomes.

MRS in chemistry is used on samples in vitro and examples of its use are in identifying the chemical structure of a newly synthesised compound or a potential drug extracted from a plant.(Gadian 1995; Assion *et al.* 1998; Alam and Alam 2005)

NMR has a role to play in medical research where the information collected by both imaging and spectroscopy can be utilised to provide a fuller picture. NMR techniques are used in the fields of cancer research, neuroscience and cardiology amongst others.

1.2 Obtaining Signals from Nuclei via NMR

1.2.1 Atoms

An atom has a central nucleus and surrounding electrons. The nucleus consists of nucleons which can be subdivided into protons and neutrons. Both protons and electrons are electrically charged whereas neutrons have no charge. Electrons are negatively charged whereas protons are positively charged.

All elements have one or more isotopes which are atoms with the same number of protons, but with differing numbers of neutrons. All isotopes of an element will have the same atomic number as that is calculated from the number of protons it has, but will have different mass numbers as that is calculated from the total number of protons and neutrons in the nucleus.

Hydrogen atoms contain one proton and one electron. Three isotopes of hydrogen occur, with zero, one and two neutrons respectively. The most common hydrogen isotope is ^{1}H, containing no neutrons and is known as protium, or more frequently as proton (although this really refers to the nucleus only). ^{2}H, containing one neutron is called deuterium (water in which ^{1}H is replaced by ^{2}H is known as heavy water) and

the isotope containing two neutrons, ^{3}H, is called tritium. This is a radioactive atom. A proton isotope of hydrogen (henceforth referred to as a proton or ^{1}H) has an atomic number of 1 and a mass number of 1. When the mass number is odd the nucleus acquires a magnetic moment. The magnetic moment has vector properties of size and direction.

A property of nuclei, which is utilised in NMR, is spin. This is based on the number of unpaired protons and neutrons in the nucleus. The unmatched nucleons are what produce the observable spin. Protons have a spin of 1/2 and nuclei from nitrogen (^{14}N) have a spin of 1. Only those nuclei that have observable spin can produce NMR spectra so that carbon (^{12}C) and oxygen (^{16}O) cannot be used.

A nucleus of spin I has 2I + 1 possible orientations, ^{1}H therefore has two. Because of this the ^{1}H nucleus can be thought of as a bar magnet with two states. When an 'MR-active' nucleus, such as hydrogen, is placed in an external magnetic field it will align its axis of rotation to that of the external magnetic field. The nuclei will align either parallel or anti-parallel to the external field. Alignment in parallel requires lower energy than alignment anti-parallel and so in any sample there will be a small difference in the number of nuclei in each direction with the greater number aligned in parallel. The energy difference is proportional to the size of the applied magnetic field. In order to acquire a signal from the nuclei, they need to be perturbed from this initial state. A nucleus can switch states from lower to higher by absorbing a photon which has energy equal to that between the two states. Nuclei with spin values greater than 1/2 have more complex magnetic properties (e.g. affecting relaxation).

Quantum mechanics describes restrictions to magnetic nuclei which means that they do not align precisely parallel or anti-parallel to the external magnetic field but at an angle. The presence of the external magnetic field also causes the nuclei to precess around the direction of the external magnetic field. The speed of precession and size of the precessional path are dependent on the type of nucleus and strength of the external magnetic field. Figure 1.1 shows the protons aligned with the magnetic field B_0

The forces of nuclei aligned parallel and anti-parallel to the external magnetic field cancel each other out. The non-cancelled ones will be in the lower-energy, parallel state. All the forces from these nuclei can be added together to produce a field longitudinal to the external field. This can be displayed as a net magnetic vector.

The value of the precessional frequency is calculated by the Larmor equation

$$w_0 = B_0 * \gamma$$

where γ is the gyromagnetic ratio i.e. the relationship between spin and Larmor frequency of each MR active nucleus and B_0 is the external magnetic field. The gyromagnetic ratio of hydrogen is 42.57MHz/T. A nucleus with a large gyromagnetic ratio has a stronger magnetic vector than one with a small gyromagnetic ratio. Hydrogen has almost the strongest magnetic vector.

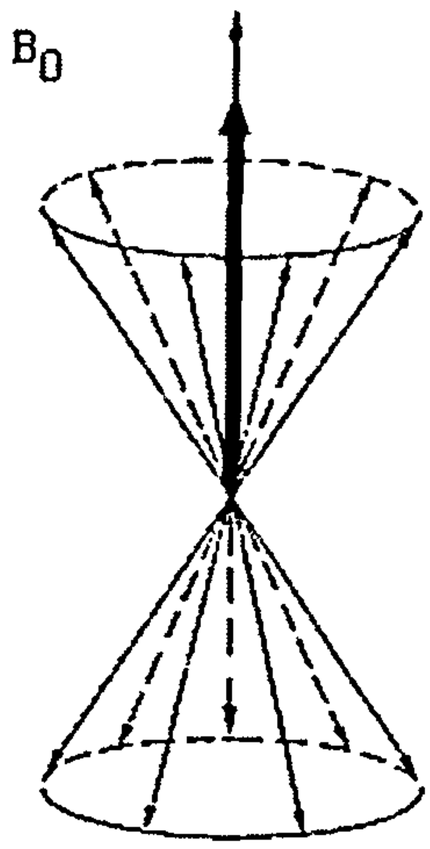


Figure 1.1 Protons aligned with the magnetic field B_0.

1.2.2 Resonance

A nucleus gains energy and resonates if external energy is applied at its precessional (Larmor) frequency. Other MR active nuclei will not resonate as their Larmor frequency is different. The application of a Radio-Frequency (RF) pulse which causes resonance is known as excitation. The absorption of energy from the RF pulse will lead to more nuclei absorbing energy and switching from the low energy

parallel orientation to the high energy anti-parallel orientation. The energy difference between the two populations corresponds to the energy required to produce resonance by excitation.

The net magnetic vector (NMV) will move out of alignment with B_0 as the nuclei change their precessional path. The angle to which they do that is known as the flip angle and is often adjusted to be 90^0 or 180^0. The RF pulses which cause these flip angles are referred to as 90^0 or 180^0 pulses. The nuclei gain enough energy from a 90^0 pulse that the NMV moves to the transverse plane. The nuclei will still precess at the Larmor frequency and they will also all be in phase, that is, they will all lie at the same point in the precessional path.

Once the RF pulse is switched off, relaxation begins, i.e. the nuclei return to the state prior to the application of the RF pulse. The NMV returns to alignment with B_0. If another 90^0 pulse is then applied the NMV will tilt to the transverse plan again. If a second 90^0 pulse is applied before the relaxation is complete, the NMV will move beyond 90^0 and the resulting signal will be different i.e. smaller.

1.2.3 Magnetic Resonance Signal

A coil is placed in the transverse plane and a signal is produced when the in-phase magnetisation cuts across the coil. The component of the NMV in the transverse plane induces a current in the coil. The current in the coil constitutes the signal. The frequency of the signal is the Larmor frequency and the magnitude of the signal depends on the amount of magnetisation in the transverse plane.

1.2.4 Free Induction Decay

When the RF pulse is switched off relaxation occurs. The NMV is again influenced by B_0, moving to align with it. The amount of magnetisation in the transverse plane decreases leading to a decrease in the current induced in the coil. This is known as the Free Induction Decay (FID) and is shown in Figure 1.2

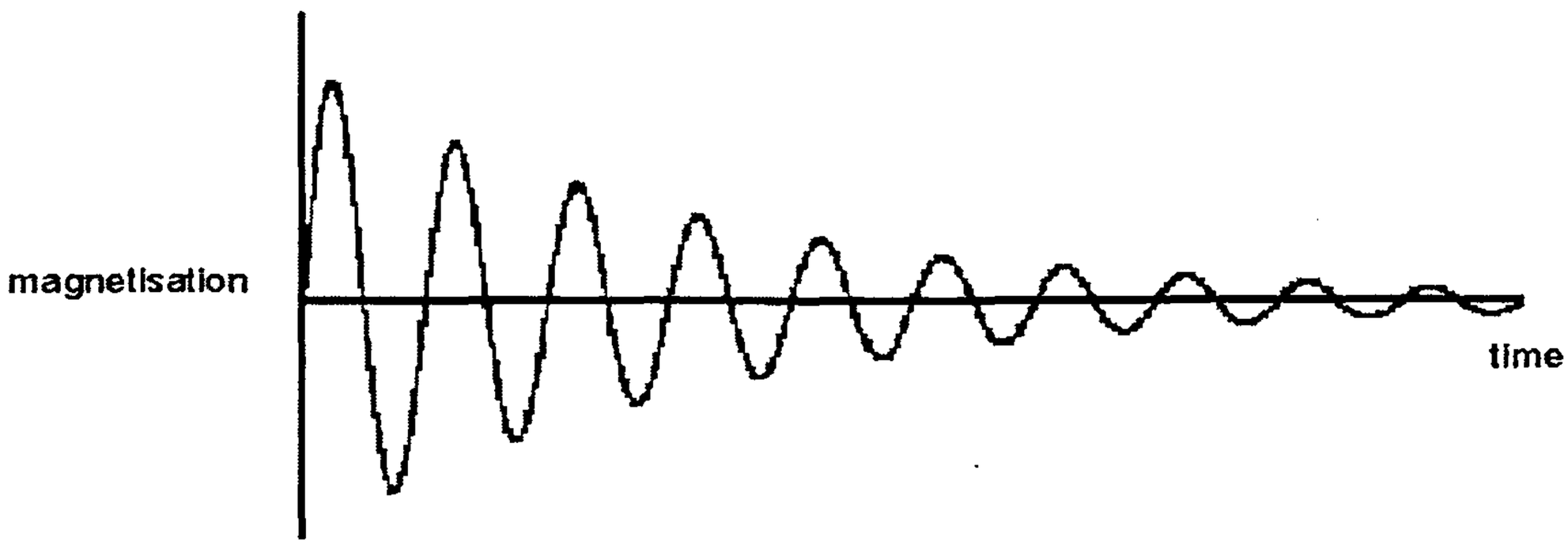


Figure 1.2 Free Induction Decay. The amount of magnetisation in the transverse plane decreases over time

1.2.5 Relaxation

Relaxation consists of two separate and independent processes, spin-lattice or T_1 and spin-spin or T_2. T_1 is longer than or equal to T_2.

T_1 is the recovery of longitudinal magnetisation through the nuclei giving up energy to the surrounding environment or lattice. The rate of recovery is exponential. T_1 is the time taken for 63% of longitudinal magnetisation to recover in the tissue. T_1 is longer in stronger magnetic fields. T_1 also varies according to the chemical and physical environment of the nuclei concerned. The characteristic differences in T_1 (and T_2) times of water in, for example, grey matter, white matter and cerebrospinal fluid in the brain can be exploited when imaging to distinguish between different tissues or to diagnose abnormalities.

T_2 occurs because the magnetic fields of nuclei interact with each other and exchange energy; T_2 is also referred to as spin-spin relaxation. This causes the decay of transverse magnetisation. The rate of decay is exponential. T_2 is the time taken for 63% of the transverse magnetisation to be lost. Essentially, the efficiency of T_2 relaxation is affected by how quickly the vector joining two interacting spins rotates.

T_2 of pure water is longer than that of water in tissues because of restricted molecular motion.

1.2.6 Chemical Shift

Nuclei are surrounded by electrons which spin on their own axis and also orbit the nucleus. Electrons are negatively charged which mean the lowest energy electrons align against an external magnetic field, unlike nuclei where the lowest energy protons align with the external magnetic field.

If the nucleus is surrounded by electrons the effect will be to shield the nucleus from the effects of the external magnetic field and so the effective field at the nucleus will be smaller than the external field, B_0 by some fraction s. Where there is no shielding, s will be zero.

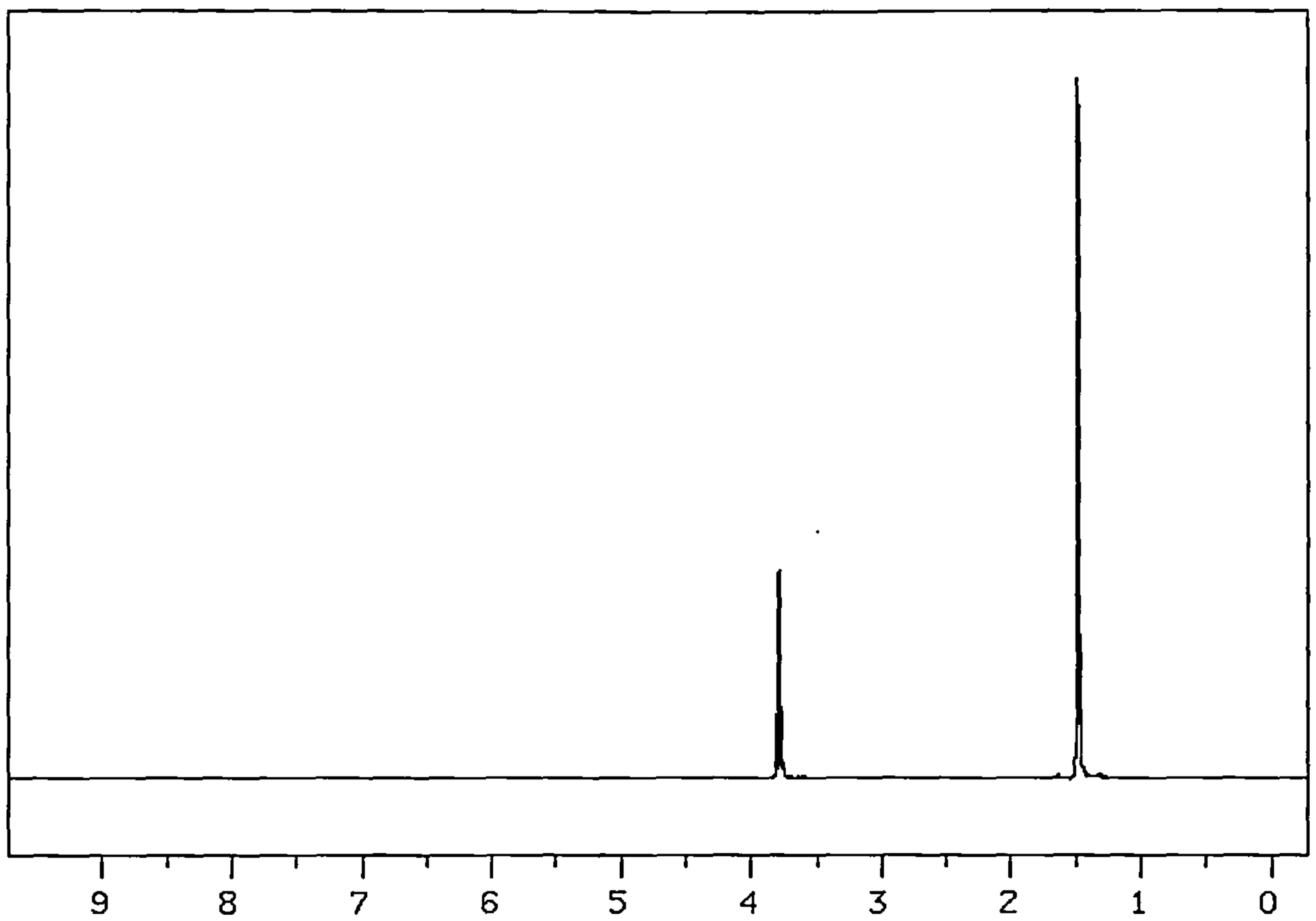
$$B_{eff} = B_0(1 - s)$$

The electron density round each nucleus varies because of the type of nucleus and the bonds in the molecule. Therefore, the shielding and the effective field will also vary. As the strength of the external magnetic field affects the energy separation of the two states of a proton, the shielding of that field by electron activity will change the energy separation and therefore will change the frequency of the RF pulse required to induce a spin transition. It could therefore be difficult to compare spectra taken from MR machines at different field strengths.

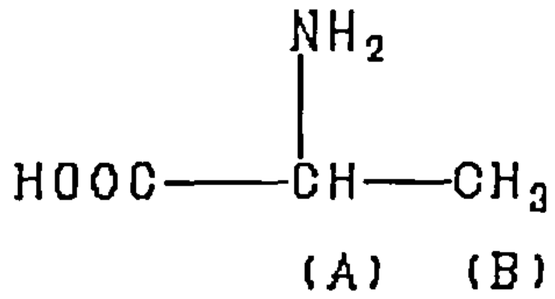
The chemical shift of a nucleus is the difference between the resonance frequency of a nucleus and a standard appropriate to that nucleus. The standard for ^{1}H and ^{13}C is usually tetramethylsilane (TMS) which is not naturally present in the body and provides a single signal at a chemical shift different to most other proton resonances, at the right edge of the spectrum because the protons in TMS are maximally shielded.

Chemical shift frequency changes are in the range of hundreds of Hz. Proton transition frequencies are nominally 60MHz for a typical clinical MR imaging

system with B_0 = 1.5 Tesla. The chemical shift changes are proportional to B_0 and are referred to as a fraction of the nominal frequency in parts per million or ppm. Figures 1.3 and 1.4 show the chemical shift of alanine and glutamine in ^1H spectroscopy along with their chemical structure.



(A) (B)



(A) (B)

Figure 1.3 ^1H spectrum of alanine showing the chemical shift, plus the chemical structure (taken from (Sdbsweb))

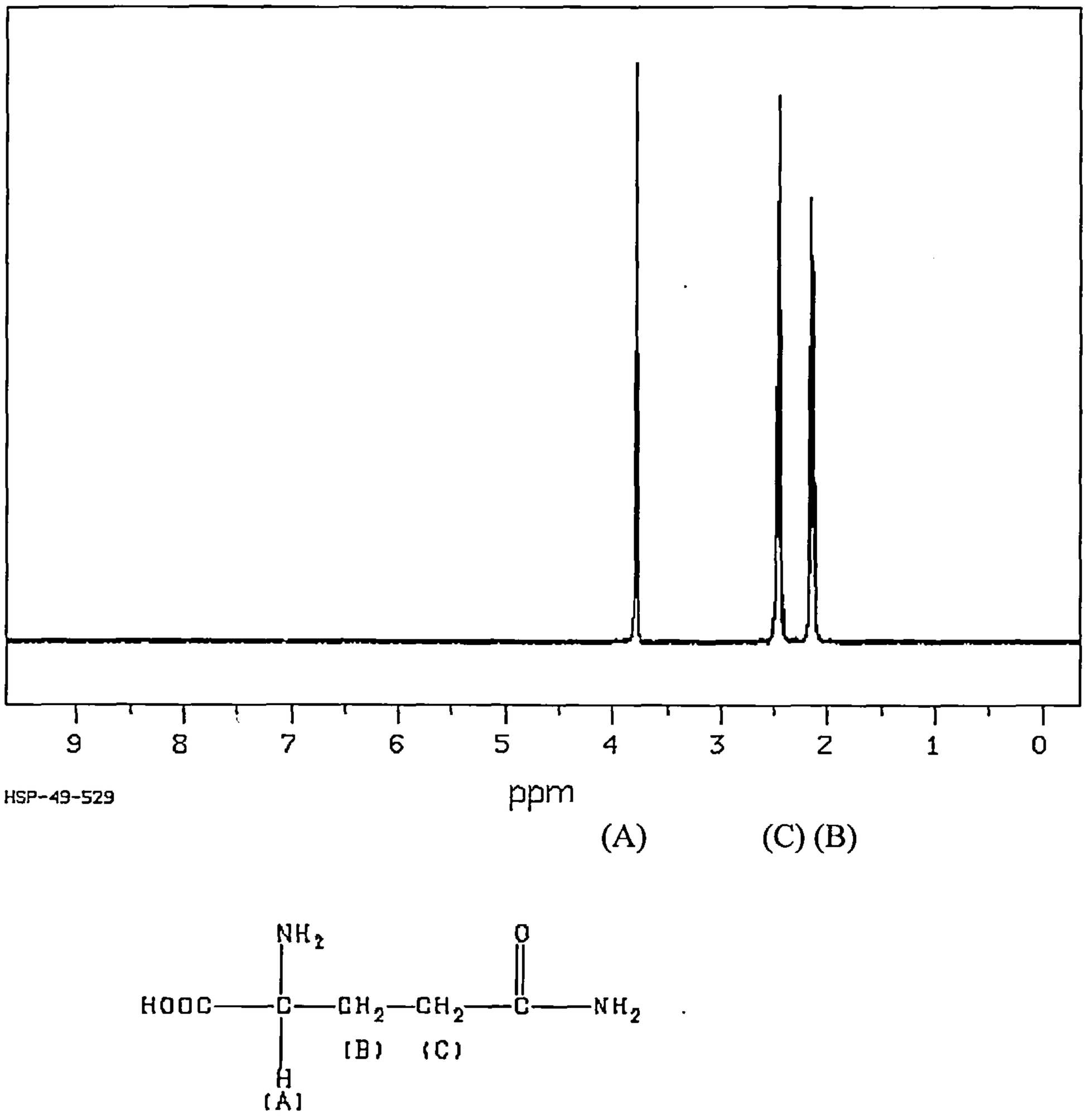


Figure 1.4 ^{1}H spectrum of glutamine showing the chemical shift, plus the chemical structure.
(taken from (Sdbsweb)

1.2.7 Fourier Transform

Radiofrequency energy can be applied at each of the frequencies that will induce resonance in protons in different chemical environments in turn - the continuous wave method. This would allow a build up of a spectrum incrementally. However, it would take a long time and spectra are not usually collected in this way. The other method is to apply an RF pulse with a wide enough range of frequencies to induce resonance in all protons whatever their chemical shift. The resulting FID therefore contains the sine waves corresponding to the resonance frequencies of all the protons (or, more precisely, the difference between the resonance frequency and the nominal (carrier) frequency). This is extremely difficult to interpret as it stands. The Fourier

transform takes the FID, which is the signal in the time domain and transforms the data so it is displayed in the frequency domain. In effect, it separates all the constituent sine waves into their frequencies and displays them in a spectrum in their chemical shift positions. Figure 1.5 shows data acquired from the NMR system in both the time and frequency domain.

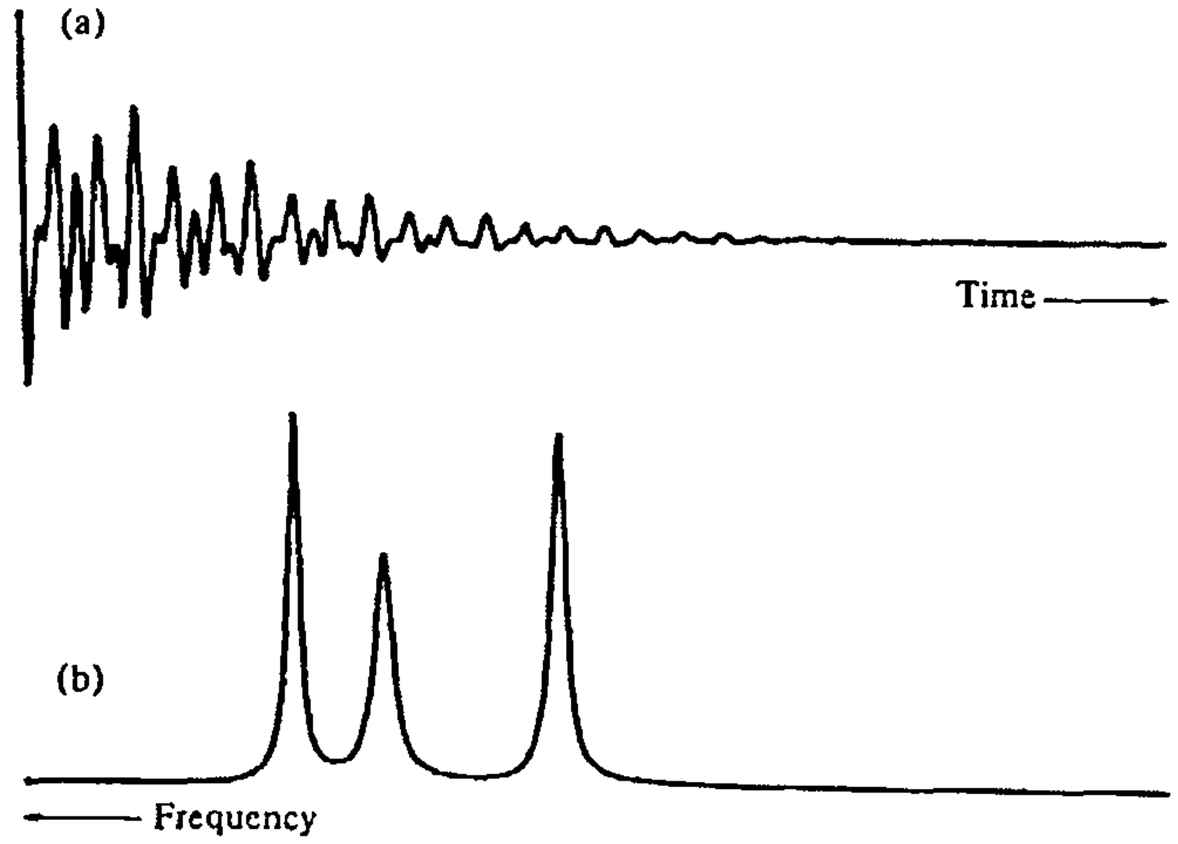


Figure 1.5 Data from an NMR system in (a) time domain (b) frequency domain

1.2.8 Signal to Noise Ratio

The signals collected from metabolites with NMR are weak and repeated data acquisition is usually applied to improve the signal to noise ratio (SNR). With n acquisitions, the improvement in SNR increases by $\sqrt{n}$. The signal increases by a factor of n while the noise, being random, increases by $\sqrt{n}$. The signal is collected at regular intervals after a pulse is applied and the data summed until an acceptable SNR is achieved.

1.2.9 NMR Pulse Sequences

In order to detect MR signals an MR magnet is used with the set up as in Figure 1.6.

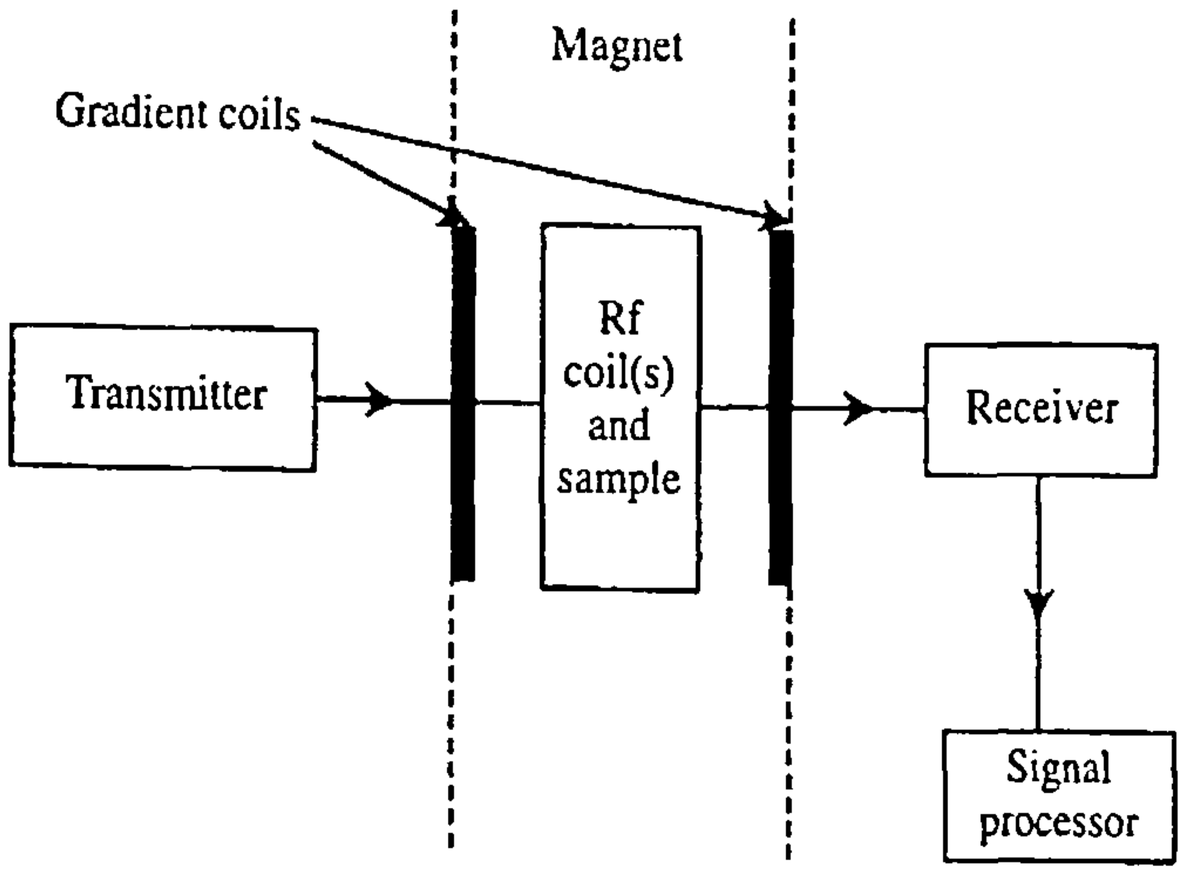


Figure 1.6 A block diagram of an MR spectroscopy or imaging system

A pulse sequence describes the series of pulse activations, waits and signal collections which are repeated one or more times in order to enhance the signals from the objects of interest and to minimise the signals from those not required or to enhance the contrast between different objects of interest

The simplest pulse sequence is a single activation pulse followed by a single data acquisition phase. More usually, a series of pulses and acquisitions are carried out. Standard pulse sequences are named and one used typically in spectroscopy is the spin-echo sequence where a 90^{o} pulse is applied followed at a set time by a 180^{o} pulse which has the effect of rephasing the protons. This pulse does not strengthen the signal, it acts as an echo, rebounding the protons to bring them back into phase. The pulse file that generates a 90^{o} or 180^{o} pulse can also be tailored for specific purposes. Square, binomial and BURP pulse shapes can all be used to generate such pulses to select particular frequencies.

Other pulse sequences such as gradient echo or for blood flow detection are more commonly used for imaging purposes rather than spectroscopy.

1.3 Factor Analysis

One of the reasons for using factor analysis is to reduce the dimensionality of multivariate data by finding underlying relationships between the data and expressing the data in one or more factors. (Reyment and Jöreskog 1993) These factors are hypothetical variables which are constructed to describe the interrelationships between the original variables and thus to simplify the data. The full set of constructed factors will contain all the information held in the original variables with the maximum number of factors being equal to the number of original variables. Factor analysis usually describes a method where finding the maximum intercorrelations between the original variables is the goal. A variation of this is Principal Component Analysis (PCA). In this method the factors are created to account for the maximum variance between the original variables. In order to use PCA the original variables need to be measured in the same units of measure and the measurements need to be of the same order of magnitude. If the original data is not of this form it needs to be standardised. PCA factors are orthogonal. The first factor is created to hold the maximum amount of variance possible. The second and subsequent factors are created to be orthogonal to existing factors whilst describing the maximum amount of remaining variance. The result of this is the first few PCA factors will contain most of the variance in the original data without holding redundant information by replicating the variance in more than one factor.

The PCA factors are designed to hold maximum variance and may therefore contain significant values from all original variables. It may be more useful to simplify the structure of the factors. Varimax (Kaiser 1958; Reyment and Jöreskog 1993) is a rotational procedure which rotates the factors in an orthogonal manner. The aim is to find a set of factors, each of which is loaded heavily on a few of the original variables and where each original variable is loaded heavily on one or a small number of factors. This creates factors which are specifically related to small groups of variables rather than being general. In the case of NMR spectra the effect of varimax rotation can be to make vectors closer to spectra of individual metabolites.

It is possible to further rotate the PCA factors using promax – a rotational technique which removes the obligation to maintain orthogonality. This allows the vectors to become simpler and more easily interpretable. However, it is the case that even after varimax and promax rotation the vectors do not always correspond to spectra from individual metabolites.

Chapter 2

Evolutionary Techniques and their Use in the Medical Domain and Techniques for Interpreting NMR Spectra

2.1 Introduction

There are two fields that come together in this thesis, firstly the use of Evolutionary Computation (EC) techniques to interpret medical data, particularly spectroscopy data. One of the main reasons for acquisition and interpretation of such data is to classify the signals into healthy or diseased or high-grade and low-grade disease. The amount of data generated by medical tests is high and does not always easily lend itself to clear interpretation by human experts. The second area is the use of EC techniques to generate data by automated or semi-automated means. A tremendous advantage of MR imaging and spectroscopy is that there is great flexibility in the choice of data acquisition parameters. This means that a wide variety of types of information can be obtained but the optimum set or sets of data acquisition conditions are not always obvious. The larger research area, in terms of published papers, is that of extracting meaning from data acquired in medical tests. This chapter will look at some of the work published from both the EC side where theories have been developed that can be used in various application areas and the NMR side where the large quantity of data that can be collected requires some automated processing to gain full value from it.

There is a large and wide-ranging body of published work dealing with the analysis and interpretation of medical data by computer methods. The use of medical diagnostic tests such as MR, Positron Emission Tomography (PET) and Computerised Tomography (CT) which produce digital data stored on a computer have allowed the collection and analysis of far larger quantities of data than

previously. The collection of such data introduces the need for computer-based analysis, both traditional statistical methods and also Bayesian and Artificial Intelligence (AI) methods. The analysis of large volumes of data to extract meaning (either explicitly defined at the outset or not) is a topic that can be described as feature extraction, classification, search or data mining.

The fields of AI, classification, analysis of computer-collected data and medical diagnosis have large bodies of work published of both theoretical background and practical applications of techniques. It is not possible to fully explore all these areas and therefore, emphasis will be placed on practical applications of techniques within the medical domain with particular consideration of feature extraction or classification.

There have been many different techniques applied to the analysis of data from medical diagnostic tests such as MRI or PET. In the main, this chapter will focus on situations where AI or other methods have been applied to MR spectroscopy data. The analysis of image data allows for AI (for instance Neural Networks (NN) and Self Organising Maps (SOM)) and other methods to aid in edge detection and segmentation.

Pena-Reyes and Sipper published a review (Pena-Reyes and Sipper 2000) in which some of the techniques of evolutionary computing are explained and their use in a medical domain is presented both as a table and an extensive bibliography. Hagberg reviewed the use of pattern recognition methods for classification of tumours from MR spectroscopy (Hagberg 1998).

The first section of this chapter is a short introduction to AI with particular reference to GA and GP. This is followed by a discussion of work relating to the PROBEN1 datasets. There will be a section on non-PROBEN1 cancer and then other non cancer related work.

The NMR section will look at the differing points in NMR collection and analysis where computer techniques have been found to be useful. These include acquisition, analysis and application areas.

2.2 Computing Techniques

2.2.1 Evolutionary Computing

AI is a term covering different methods of organising a computer to solve a problem without enumerating the steps required to do so. The term can be used to describe search techniques, natural language processing, expert systems, neural networks and evolutionary computation amongst other things. Although not specifically focussed on AI (Berthold and Hand 1999) has chapters on neural networks, fuzzy logic and stochastic methods (covering EC as well as simulated annealing) and Bayesian methods.

Evolutionary techniques use analogies from biology such as survival of the fittest, sexual reproduction and mutation. They describe a set of techniques that find solutions to problems by encoding single or populations of possible solutions and then trying to find more successful (fitter) ones by allowing the fitter solutions to survive and to take part in a swapping of part of the solution with another relatively fit solution. In order that the process can find its way out of a situation where a solution may need to get worse before it can get better, the solutions can also be randomly mutated. Evolutionary techniques have developed along three main lines with the terms genetic algorithms (GA), evolution strategies (ES) and evolutionary programming (EP) being used to describe them (Genetic programming (GP) is most similar to the GA line). The differences between them stem from which facet of evolution is emphasised (Fogel 1994). Figure 2.1 shows the life cycle of a population-based evolutionary technique.

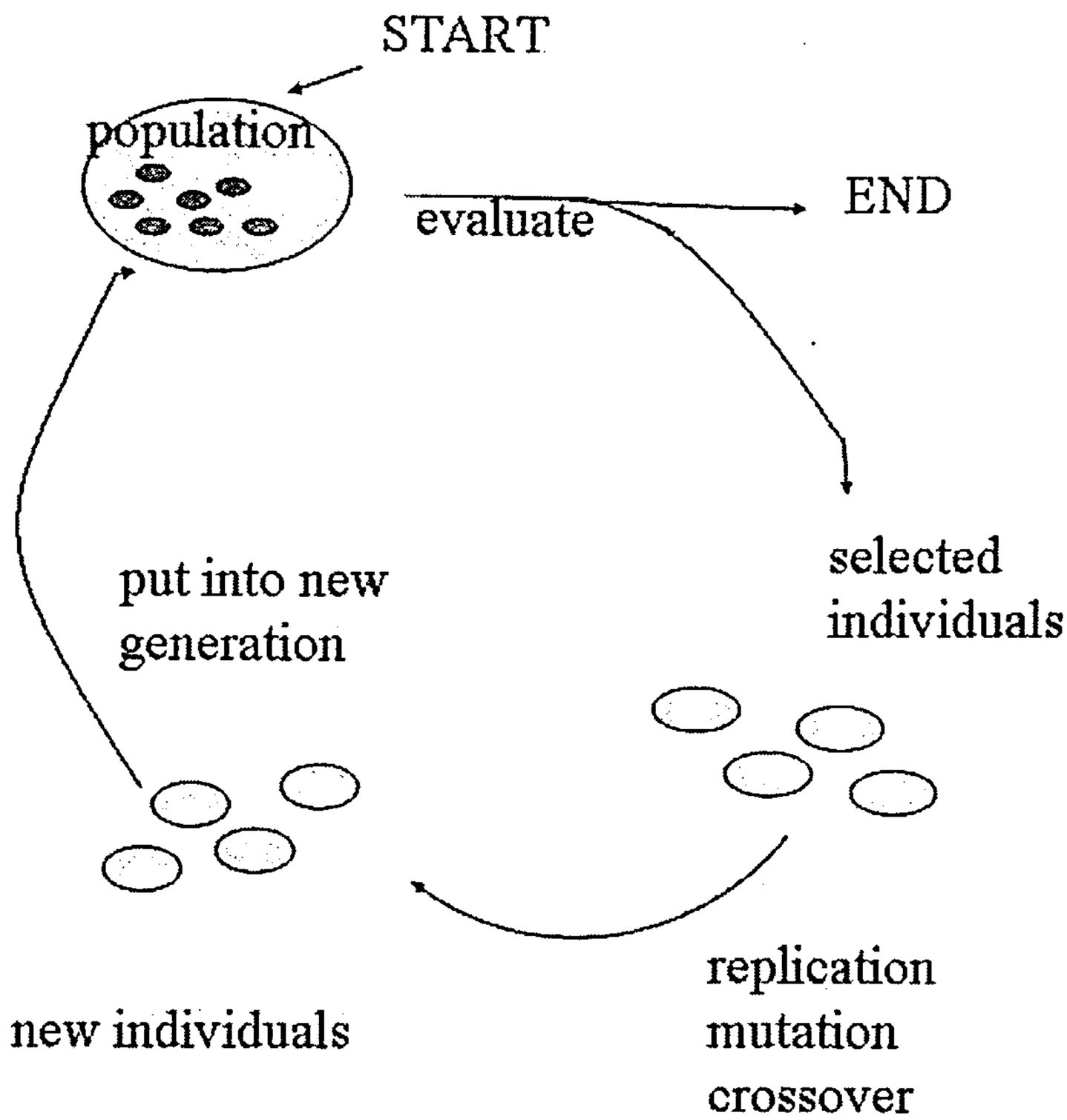


Figure 2.1 The life cycle of a population-based evolutionary algorithm

2.2.2 Genetic Algorithms

Genetic Algorithms have been described in works by Holland (Holland 1975), Goldberg (Goldberg 1989) and Mitchell (Mitchell 1998). At first GA used a binary-coded vector to describe a candidate solution to a given problem, where each bit of the vector could take one of only two values. The vector was referred to as a chromosome and often needed to be decoded to produce the solution.

In GA a population of chromosomes is created. Each of these chromosomes is tested against a set of examples of the problem to be solved. How well a chromosome performs against this training set of examples is known as its fitness. The fitter solutions are more likely (through one or more of a set of possible selection procedures) to be selected to be involved in the next generation of chromosomes. The standard operations available are replication, mutation and crossover. In replication, a chromosome is copied across to the next generation unchanged.

Mutation involves the flipping of a single bit of the chromosome to the other possible value. Crossover (initially single point) selects the same point in two chromosomes and swaps the contents of the part after the crossover point with each other. The chromosomes in this next generation are then assessed for fitness. The process continues until a fit enough solution is found or a set number of generations have happened. Binary-coded GA is still used but real-valued GA is more widespread as it increases the range of problems that can be encoded. The encoding of a single real number in binary coded GA could take 32 elements (bits) of a chromosome or one element with real-value coding. The shortening of a chromosome will have effects on the time taken for fitness values to be calculated and for genetic operators such as crossover and mutation to be performed. Mutation and crossover will have different effects depending on the coding method selected.

2.2.3 Genetic Programming

Genetic Programming was first described by Cramer (Cramer 1985) and popularised by Koza (Koza 1992) There are good introductory and more complex texts on GP in (Koza 1994; Banzaf 1998; Koza *et al.* 1998; Baeck *et al.* 2000; Koza *et al.* 2003). GP follows on from GA with the use of populations of candidate solutions. The basic data structure used by GP is a tree and in Koza's terminology is referred to as an individual. The tree contains function and terminal nodes (variables or constants). The alphabet of the terminal nodes and functions are selected as being sufficient to describe a solution to the problem. The individual has its variables instantiated with test data and the result of the function application over all test data will determine its fitness.

The data structure used allows for different size and shapes of individuals to be constructed. In Koza's description there is a description of closure by which all functions accept arguments and return results of the same type and all terminals are of that type. This means that all mutation and crossover operations will result in a syntactically correct individual. Most GP still works on typeless individuals but there has been much reference to the strongly typed GP of Montana (Montana 1993). With typed GP the representations of solutions may be closer to the real-world nature of a solution but the need to type all functions leads to more complexity in setting up the GP process.

GP emphasises crossover as the main genetic operator, unlike GA which has tended to use a greater proportion of mutation.

2.3 The Use of Evolutionary Computing Techniques for Cancer and Other Disease Diagnosis

The UCI repository of real world databases (Blake and Merz 1998) contains both medical (breast cancer, thyroid disease) and non-medical (classifying glass types, approval or not for a credit-card) data. These real world datasets have been donated by researchers in order to allow benchmarking of machine learning techniques

Schiffmann (Schiffmann *et al.* 1992) describes a technique where GA is used to optimise the topology of a NN. The NN is then used to classify the examples. The dataset is not referenced but it has the same number of examples and type of input features as that described in the UCI repository. The results, which show that GA can optimise NN topology, may not be directly comparable with others using the dataset.

Prechelt (Prechelt 1994) has taken some of the UCI datasets and encoded them specifically as NN benchmarking datasets described as PROBEN1. He has dealt specifically with null values and incomplete data. He has separated data into training, validation and test sets so that results reported would be directly comparable. Where the datasets are small, such as with glass, different partitioning may lead to differing results, and so he has created three different partitionings of the data which differ only on the order of the examples. This means that results can be reported for each separate partitioning. The paper also describes how the datasets should be used so as to allow real comparison between techniques. It also gives some baseline results for the datasets including breast cancer.

Some features of the classification datasets in PROBEN1 are shown in Table 2.1

Name	No. features	No. examples	No. outputs
Cancer	9	699	2
Card	51	690	2
Diabetes	8	768	2
Gene	120	3175	3
Glass	9	214	6
Heart	35	920	2
Horse	58	364	3
Mushroom	125	8124	2
Soybean	35	683	19
Thyroid	21	7200	3

Table 2.1 Features of the PROBEN datasets

The PROBEN1 datasets are suitable for supervised NN learning as each example has an output value, and so are also suitable for evolutionary techniques.

Brameier and Banzhaf (Brameier and Banzhaf 2001) use Linear GP applied to some of the PROBEN1 datasets. Linear GP uses a sequence of instructions from an imperative language as its data structure. The linear structure is represented by a variable length string containing simple C instructions. Each instruction involves an assignment to a variable and thus multiple program outputs are simpler than with tree-based GP where explicit coding of variable assignment is required. An advantage of Linear GP reported in the paper is that 'introns', another analogy from biology here referring to non-useful pieces of code, can be removed prior to execution, thus speeding up execution time.

The experimental set up involved the division of the population into 'demes', a type of island model in which individuals can migrate to one other specified deme only.

The results reported show that Linear GP can reach a similar generalisation performance as NN using a backpropagation learning algorithm. They also report

that runs using demes perform as well as those without but that effective training time is reduced.

Oltean and Grosan (Oltean and Grosan) use infix form GP on the same datasets. This technique has individuals as strings encoding complex mathematical expressions in infix form. They report results that are similar to those from Linear GP but with a smaller population size and vector length. The results are reported for the cancer dataset as well as those of diabetes, heart disease and horse colic.

The production of comprehensive rules for classifying data is the focus of (De Falco *et al.* 2002). They use a series of one class versus the rest to distinguish between classes. They use a function set containing logical (AND, OR, NOT) and relational (<, <=, = ,>,>=) operators which lead to trees that can be interpreted as if <condition> then <class>. The fitness function has a parsimony element to encourage more compact solutions. The reported results are slightly worse than those reported by Banzhaf and Brameier but the authors claim that the simplicity and compactness of the rules evolved make their technique preferable. The inclusion of arithmetic functions alongside logical ones does improve performance but at the expense of comprehensibility.

Feature selection and fuzzy modelling are used with the PROBEN1 datasets in several papers. Emmanouilides' paper (Emmanouilidis *et al.* 1999) uses multiple-criteria GA and tests it on the cancer dataset although there is no real discussion of the results. Yang and Honavar (Yang and Honavar 1997) use a GA to select feature subsets in the automated design of NN pattern classifiers. They report success of this method on the PROBEN1 datasets (they do not use the cancer dataset).

Pena-Reyes (Pena-Reyes and Sipper 2001) describes a method of using co-evolutionary algorithms with fuzzy systems and test it on the cancer dataset. Co-evolution refers to the simultaneous evolution of two or more populations of individuals where the fitness of the populations is coupled. A cooperative rather than competitive fitness strategy is used to coevolve two species which fulfil the criteria for two interconnected search processes required for fuzzy modelling – that for the membership functions (operational parameters) and that for the rules

(connective parameters). The search is undertaken through an evolutionary approach. They report that their system is more successful than previous work on the dataset using both genetic fuzzy and other approaches.

A hybrid of GA and kNN (k Nearest Neighbour, where k is a number) methods to classify is described in (Raymer *et al.* 1997). The authors aim to reduce the number of features required for classification by identifying those that are useful and eliminating those that are not as they state that even those features that may contain some useful information can reduce the accuracy of the classifier when the number of training cases is low, as with many medical applications. Their method involves the GA chromosome being divided into two sections. The first is a selection vector which has a set of bits per feature, set to 0 or 1. If a majority of the bits are 1, the feature will be included, and not otherwise. The use of a set rather than a single bit per feature reduces the effect a single mutation has. The second half of the chromosome is a weight vector with a single value per feature. The feature values of each sample in the test set are multiplied by both the selection and weight vectors and the resulting values passed to a kNN classier. The fitness of the GA is measured by the accuracy of the kNN classifier and the number of features used in the classifier, with fitness increasing as the number of features used decreases. The method was applied to environments of water molecules bound to protein surfaces and to clinical test results for patients with suspected thyroid dysfunction. The results show that classification performance was good in both cases, with the use of the selection vector, allowing similar classification accuracy using fewer features than that of the GA without the selection vector.

Other papers that have reported work on different techniques and diseases include Sierra and Larranaga (Sierra and Larranaga 1998) who use a GA to induce Bayesian Networks based on different methods and then comparing it with a Naïve-Bayes network. They apply it to predicting survival from malignant skin melanoma after one, three and five years. The GA induced networks had a maximum accuracy of 94% for prediction of survival after one year. Survival after five years is more difficult to predict with the networks having an accuracy of 69% - 78%.

Dracopoulos and Kent use GP rather than GA to classify patients into those at risk from oral cancer and those not at risk, based on age, smoking and drinking habits and regularity of dental visits (Dracopoulos and Kent 1997). They have a large dataset of 991 training samples and 132 test samples. The samples are heavily weighted in favour of negative examples with 95% samples in both sets being negative examples. They report results on a par with those found from a NN but poor compared to those from an experienced manual screener. Langdon also uses GP in combination with NN for drug discovery (Langdon *et al.* 2004) and for data mining DNA chip data from cancer patients (Langdon and Buxton 2004). The use of GP with a classification tree is described in (Marmelstein and Lamont 1998) where their results show hat this approach is faster than standard GP, the solutions are less prone to bloat and are easier to understand (although simplicity of solutions is not necessarily a feature of either GP or classification trees). The disadvantage of this approach is the computational time taken to find solutions.

Inza (Inza *et al.* 2001) uses GA for feature subset selection in the survival of cirrhotic patients and Aliferis (Aliferis *et al.* 2002) classify non-small cell lung cancers. A study looking to classify cervical cancer by identifying sub-visual changes to cells taken from cervical smears (Hallinen 2001) used a GA to select input features and a weight vector which were then used by a NN to identify the presence of one of three types of abnormal cells or the absence of any. The study was unusual in medical data analysis by having a large set of data; 300 samples in each of the training and test sets and a balance of normal to abnormal data (an approximate ratio of 2:1 abnormal to normal). The authors compare their technique with standard linear discriminant analysis (SLDA) and show that the AI method finds more true positives but also finds more false positives than SLDA. The AI technique is also more generalisable than SLDA, though no significantly so. Their conclusion is that while such feature selection is valuable, the computational overhead required is too high to make it a feasible alternative to the statistical approach.

Wang (Wang *et al.* 1995) used data from CT scans as input to a self-adaptive expert system to diagnose brain tumours.

2.4 NMR Spectroscopy and Pattern Recognition

Much of the early NMR work has focused on the image output with image segmentation (Bezdek *et al.* 1993; Özkan *et al.* 1993; Worth and Kennedy 1994; Poli) the largest area of research. The acquisition and analysis of spectroscopy data has been described in both medical and chemistry fields. The use of NMR in chemistry will not be discussed here but a review of multivariate analysis of NMR in chemistry as well as food science and materials can be found in (Alam and Alam 2005)

The application of AI techniques to the field of NMR, and particularly to Magnetic resonance spectroscopy (MRS), can be divided into the following areas (Figure 2.2). In data acquisition GA has been applied to pulse design (Freeman and Wu 1987; Geen and Freeman 1991). Data analysis and quantification of the resulting spectra can involve (peak) fitting (Ala-Korpela *et al.* 1995; Pearlman 1996; Stoyanova and Brown 2002). An application of NMR in the biochemical field involves searching for the structure or conformation of proteins or similar. Much work has been done in this area with GA (Unger and Moult 1993; Wehrens *et al.* 1993; Bayley *et al.* 1998; Piccolboni and Mauri 1998; Krasnogor *et al.* 1999)

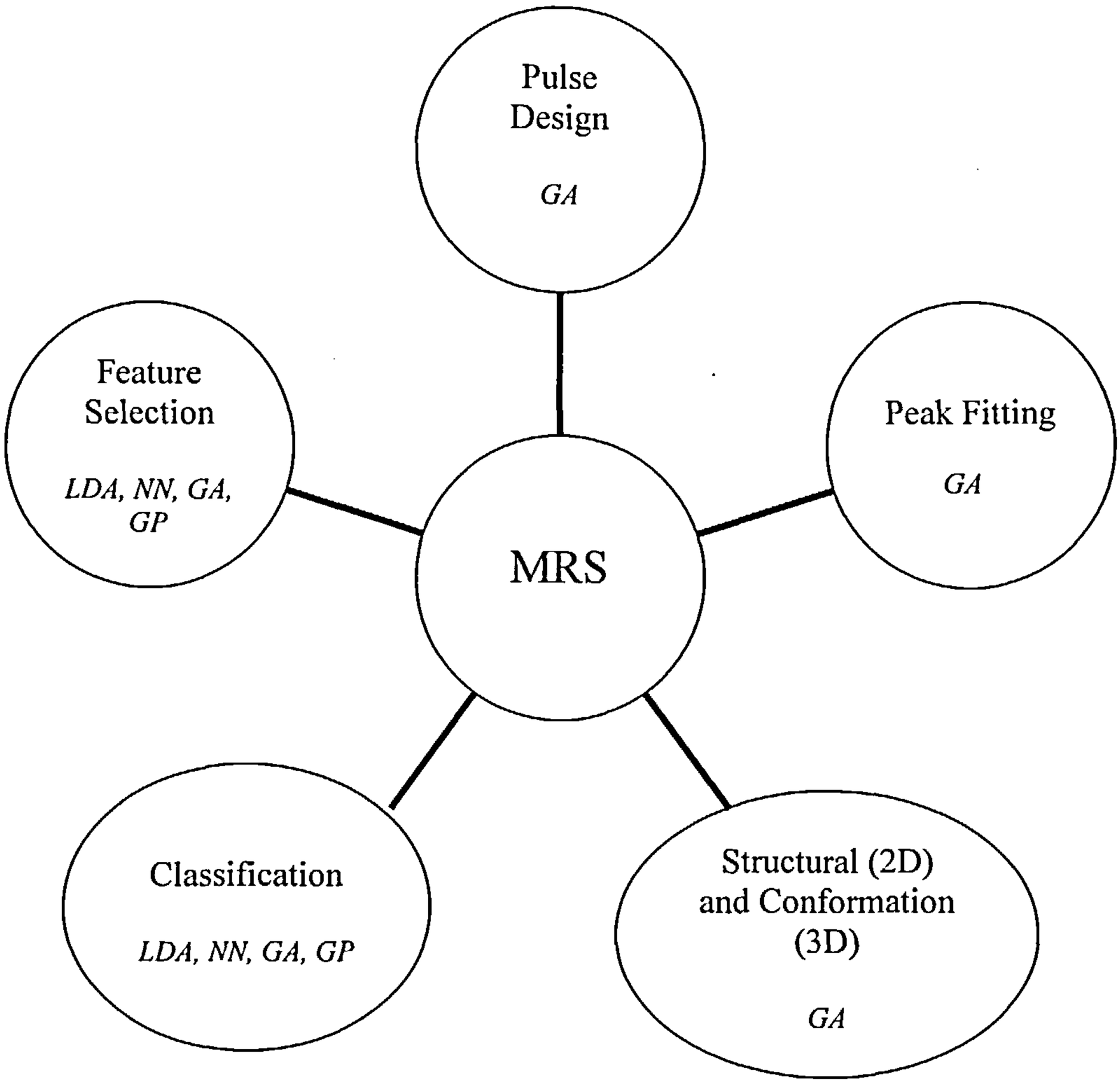


Figure 2.2 Application of AI to MRS

Another application is classification of data based on spectra and includes work on pre-processing of data as well as feature selection. Both GA/GP and NN have been used for classification along with statistical and other methods (Rutter *et al.* 1995; Usenius *et al.* 1996; Bakken, I *et al.* 2001; Mountford 2001). Much of the literature in this field uses the term feature extraction to describe the same sort of work as would be described as classification or data mining in the EC literature. El-Deredy describes different pattern recognition approaches in his review which covers both dimension reduction of data (including by PCA) and techniques (including NN and GP) which classify data.

A paper was published in 1989 (Thomsen and Meyer 1989) in which the authors suggest that theirs is the first example of a NN being applied to classification of NMR spectra. Since then there have been many papers published on the use of pattern recognition methods for classification of spectra. Many of these have dealt with NN (see Table 2.2) but others have used a range of techniques.

A special edition of NMR in Biomedicine on Pattern Recognition was published in 1998. One of the articles was a review of methods used for tumour classification with NMR spectroscopy (Hagberg 1998). In this article the author examines techniques for pre-processing of data (digitisation into chemical shift ranges, wavelet transform), feature extraction (PCA, Linear Discriminant Analysis (LDA)) and classification algorithms (cluster analysis, NN). A table is shown of MRS classification of tumours, showing the methodology used and results obtained. There is only one evolutionary computation technique in this table, a GA used as part of a consensus diagnosis (Somorjai *et al.* 1995).

Other papers in the same edition of the journal show different methodologies and some of these papers will be described below.

Anthony (Anthony *et al.* 1994) describes the use of both PCA and non-linear mapping (NLM) to reduce the dimensionality of NMR spectra data. Measurements are taken from 16 metabolites and the distances between the objects are formed by using least-squares minimisation for the NLM. Alternatively PCA is applied to the metabolite values from the spectra. NLM refers to the reduction of dimensionality of data so that it can be visualised in two or three dimensions. The relative distances between objects are preserved. The use of NLM by Anthony uses the same measurement of error (least squares minimisation) as with Sammon Mapping (Sammon 1969) but it is unclear from Anthony's paper whether other features of the NLM are the same as those described in Sammon's paper. Sammon Mapping is a widely form of NLM utilised in this field.

Lisboa (Lisboa, P *et al.* 1998) uses PCA and also partial least squares to reduce dimensionality of NMR spectroscopy data with the resulting variables passed to statistical techniques. The use of statistical techniques is compared with NN for

tissue classification. The statistical methods are linear discriminant analysis (also reported by Tate (Tate *et al.* 1998)) and 1st nearest neighbour (kNN, where k=1). LDA seeks to provide linear combinations of variables so that the separation of the class means is maximised relative to the variance within each class. kNN with k=1 allocates each sample to the class of its nearest neighbour. These techniques are compared with the use of a multi-layered perceptron. The results are reported as comparable for both statistical and NN techniques.

Holmes (Holmes *et al.* 1998; Holmes *et al.* 2000) uses a multi-class form of PCA, called SIMCA, where each class is described by a separate PCA model. She suggests that the technique performs poorly where there are a large number of classes or subsets. She also draws attention to the fact that using a large number of principal components may lead to over-fitting. She suggests using pre-screening to remove outliers from the data set.

Although many approaches have been used to classify disease, including cancer, from MR data there appears to be a benchmark of success at approximately 80%. It is difficult to assess completely the success of one method against another because of the differing sizes of samples, number of classes and use or otherwise of validation and test sets as well as the differing techniques but there is a commonality of feature analysis. Table 2.2 summarises some of the work in this field and highlights the reported results. The table is not an exhaustive list of all the work in the field but shows some of the methods that have been used.

The focus of most papers is that of research where a small number of samples are used to determine whether a technique can produce meaningful information from spectra. The uptake of such techniques in a clinical setting has not followed the reportedly good results possible in many fields. The use of magnetic resonance spectroscopic imaging (MRSI), where spectra are taken at the same time as MR imaging, is described in (Gruber *et al.* 2005). In this paper the standard technique of brain lesion classification, biopsy, has risks but has a diagnostic accuracy of up to more than 90% depending on tumour. There is less accuracy with grading of tumours and with differentiation between tumour and non-malignant abscesses. The use of MRSI at this stage can help with treatment strategies. The authors give

examples of individual patients where MRSI aided diagnosis and treatment of brain tumours. They predict that the continuing development of such techniques will lead to an improvement in patient survival rates. However, the costs involved as well as technical difficulties of collecting good data, mean that such systems will probably only be used for brain tumours, at least for the next few years. The analysis of spectra described in this paper use both spectrum fitting programs and pattern recognition techniques.

An alternative approach to examining a single voxel as in standard spectroscopy is to use multivoxel MRSI where a grid of voxels is examined. This allows segmentation into different tissue types based on spatial as well as metabolic information. In (De Vos *et al.* 2007), (Laudadio *et al.* 2007) successful nosological classification of tumours is carried out by a technique based on Canonical Correlation Analysis. The technique results in coloured segmentation images showing areas of different tissue types including tumour, necrosis, normal tissue and mixed tissue.

The use of MRSI may help with the problem of small data sets but it appears that the use of AI techniques to aid in medical applications has not yet passed from a research to a clinical setting in the way that authors of early papers had predicted.

Reference	Subject area	Sample	Purpose and area studied	Computing technique	Classes	Sample size	Results reported
(Anthony *et al.* 1994)	Nephro-toxocity	Urine	classification	PCA, NLM	Various	54	PCA better than NLM at discriminating between sites of action of treatments
(Somorjai *et al.* 1995)	Cancer	Biopsies of thyroid neoplasms	Classification	LDA, NN, GP	2	107	
(Ala-Korpela *et al.* 1996)	Lipo-protein abnorm-alities	Plasma	Lipoprotein quantification -0.019–0.019ppm (101 data points)	NN, Kohonen net		55	
(Usenius *et al.* 1996)	Cancer	*in-vivo* brain	classification 0.3 – 3.4ppm (206 data points)	NN	4	33 patient, 28 control	82% success
(Maxwell *et al.* 1998)	Cancer	Perchloric acid extracts of	classification	PCA, NN	2 (meningiomas / non-meningiomas) 4 (grades of glial tumour)	118 47	85% correct 62% correct
(Bakken, I J *et al.* 1999)	Epilepsy		classification	NN	2 (Healthy, diseased)	15 patient, 13 control	
(Gribbesta d *et al.* 1999)	Cancer	Perchloric acid tissue extracts	Metabolite composition and concen-tration 2.8 – 3.5ppm (1000 points / 17 PCs)	PCA Probabilistic NN		16 patient	

Reference	Subject area	Sample	Purpose and area studied	Computing technique	Classes	Sample size	Results reported
(Poptani *et al.* 1999)	Cancer and non-tumour cerebral disorders	*in-vivo* brain	classification 0.6 – 3.4ppm (75 data points)	NN	5 (2 grades of glioma, tuber-culomas, abscesses, healthy)	98 patient, 40 control	73% (low-grade gliomas versus rest) 98% (high-grade gliomas versus rest)
(Bathen *et al.* 2000)	Cancer, CHD†	Blood Plasma	Quantification of lipids and apolipo-proteins 0.4 –1.4ppm (1900 points)	PCA, PLS‡, NN	Up to 14	52	PLS and NN both good
(Gerstle *et al.* 2000)	Cancer			NN and LDA*	2 (SCCA, muscle)	16 patient, 12 control	
(Axelson *et al.* 2002)	Parkinson's Disease (PD)	*In vivo* basal ganglia	classification 1.5 – 4.0 ppm (274 data points)	Feature selection by GA, NN	2 (disease or no disease) 4 (no dis-ease,probable, possible, atypical PD	31 patient, 14 control	2 class excellent results, 4 class 88% correct.
(Pulkkinen *et al.* 2002)	Cancer	MRSI spectra of *in vivo* tumour (glioma)	Chemical shift correction (241 data points)	SOM§ prior to LF and NN	Unsupervised but SOM gave 5 clusters	71 patients, 14 controls	94% test set correct
(Lee *et al.* 2000)	Cancer	MRS spectra of cysts and brain tumours	classification 0.5 - 4.2ppm (194 data points)	Pre-filtering or ICA∞ followed by LDA	6 classes of tissue using pairwise discrimin-ation	98	56% to 96% discrimination between pairs of classes
(Huang *et al.* 2003)	Cancer	MRS spectra of brain tumours	classification	ICA or MBVS◊	2 (Astro-cytomas and glio-blastomas	41	62% - 90% depending on method

† Coronary Heart Disease
‡ Partial Least Squares
§ Self-Organising Maps
* Linear Discriminant Analysis
∞ Independent Component Analysis
◊ Multivariate Bayesian Variable Selection

Table 2.2 Techniques used to extract meaning from MR data

Chapter 3

The Use of Genetic Methods to Classify Brain Tumours from NMR Spectra

3.1 Introduction

A tumour is a growth of tissue made up of abnormal cells. There are nearly 100 types of brain tumour. Tumours that occur in the brain can be primary (the site of the first occurrence of this cancer) or secondary (where cancer cells travel from the originating site elsewhere in the body to the brain via blood circulation). These are also referred to as metastatic tumours, or metastasis. Primary brain tumours are often named after the type of cell they developed from. The main cell type in the brain is the neurone which rarely produces cancer cells. Meninges is the name given to the membranes that cover the brain. Meningiomas are benign tumours, meaning that they are slow growing and the cells do not spread to other tissues. However, the growth of any tissue in the brain can cause problems due to the rigidity of the skull. Glial tissue supports the neurones and nerve fibres and types of glial cell include astrocytes and oligodendrites. Most malignant brain tumours occur in glial cells and include gliomas, astrocytomas and oligodendrogliomas. Malignant tumours are graded 1 - 4, high grades (3 - 4) describe a more aggressive tumour, low grades (1 - 2) a more slow growing one. Glioblastoma multiforme is a high grade astrocytoma (grade 4) and is the most common type of primary malignant brain tumour in adults. Medulloblastomas occur in the cerebellum and are high grade malignant tumours which are one of the most common brain tumours in children.

Malignant tumours are not homogeneous; in addition to areas of tumour cells there can be areas of necrosis (dead tissue) and oedema (swelling due to extra-cellular

fluid). It may also be difficult to distinguish the edge of a tumour so there can be a mixture of cell types. Benign tumours tend to be more homogeneous.

Classification into two classes is common in all areas including medical applications. A distinction between presence and absence of disease is the most crucial step in diagnosis. Further division into types or grades of disease can then be undertaken. As classification tasks can be generally be posed as a series of one or more one-versus-the-rest binary classifications it is important to show that a computing technique can produce good results on a binary classification task. It can also be useful to show that data from a technique such as NMR can be used to distinguish between categories.

The application discussed in this chapter is to classify tumours from ^{1}H NMR spectra (which give information about biochemical compounds in a tissue). These spectra can be obtained *in vivo*, but more detailed information is available if they are obtained from chemically extracted biopsy samples. It would be advantageous if analysis of spectra data such as these could be used for diagnosis of cancer type or prediction of treatment response. If this is possible it would be useful to know what information in the spectra is being used to distinguish between different types of tumour. The specific data used was a database of 'H NMR spectra of human brain tumour biopsies. This database was collected as part of a European Community Concerted Action (Biomed-I PL920432).

Two types of evolutionary computation (GP and GA) were used with this data. GP and GA both use a population-based strategy whilst differing in the representation of candidate solutions present in the population. As NN have been used on the same data set with success a comparison of NN with GA and GP was made.

3.2 Methods

The first stage of starting the classification was to collect and organise the data into a form that could be input into the classification software. The original data were collected in the following way; 75 brain tumour biopsies were taken during routine

surgery, frozen in liquid nitrogen and extracted with 0.5M perchloric acid, as described in (Remy *et al.* 1994). Histology (carried out by the pathology services of the collaborating hospitals) showed that 28 were meningiomas and the remainder (non-meningiomas) included a variety of different tumour types: (astrocytoma (14 samples), glioblastoma multiforme (9), medulloblastoma (5), metastasis (5), oligodendroglioma (3), haemangiopericytoma (3), haemanioblastoma (1), lymphoma (1), schwannoma (1), chordoma (1), radiation necrosis (1), pineal teratoma (1), histiocytosis (1), angiofibroma (1)). This database is a sub-set of that described by Maxwell *et al.*,(Maxwell *et al.* 1998) consisting of the first 75 samples collected in that study.

'H NMR spectra were obtained at 360 or 400 MHz in $^2H_20$ at pH* 7.0 (pH* = pH meter reading uncorrected for the deuterium isotope effect) and 25°C. The spectroscopy parameters were: 90° pulse, pulse repetition time 10 s, spectral width 4803 Hz, 16000 data points, water presaturation and 64-1024 averages depending on initial tumour weight.(Griffiths 1996)

The spectra were digitised at intervals of 0.010 ppm over the range 4.5-0.5 ppm, giving 400 variables. Variables arising from lactate (1.26-1.40 ppm, corresponding to the variables 311-325; and 4.09-4.18 ppm, corresponding to the variables 33- 42), mannitol (3.63-3.88 ppm, corresponding to the variables 63-87) and a signal at 1.64 ppm (variables 283-292) were all set to zero because they were considered to be unreliable. Lactate is expected to increase during the interval between tumour excision and freezing, and it was difficult to control this period during routine surgical procedures. Mannitol is given to patients, in varying amounts, to reduce brain oedema. The 1.64 ppm signal was considered to be an artefact of the extraction procedure. This is attributed to acetone which is used for cleaning tubes and possibly used in the extraction process. Finally, the digitised spectra were normalised to the sum of all (remaining) variables. Figure 3.1 shows the mean spectra from the two main tumour classes; meningiomas and non-meningiomas.

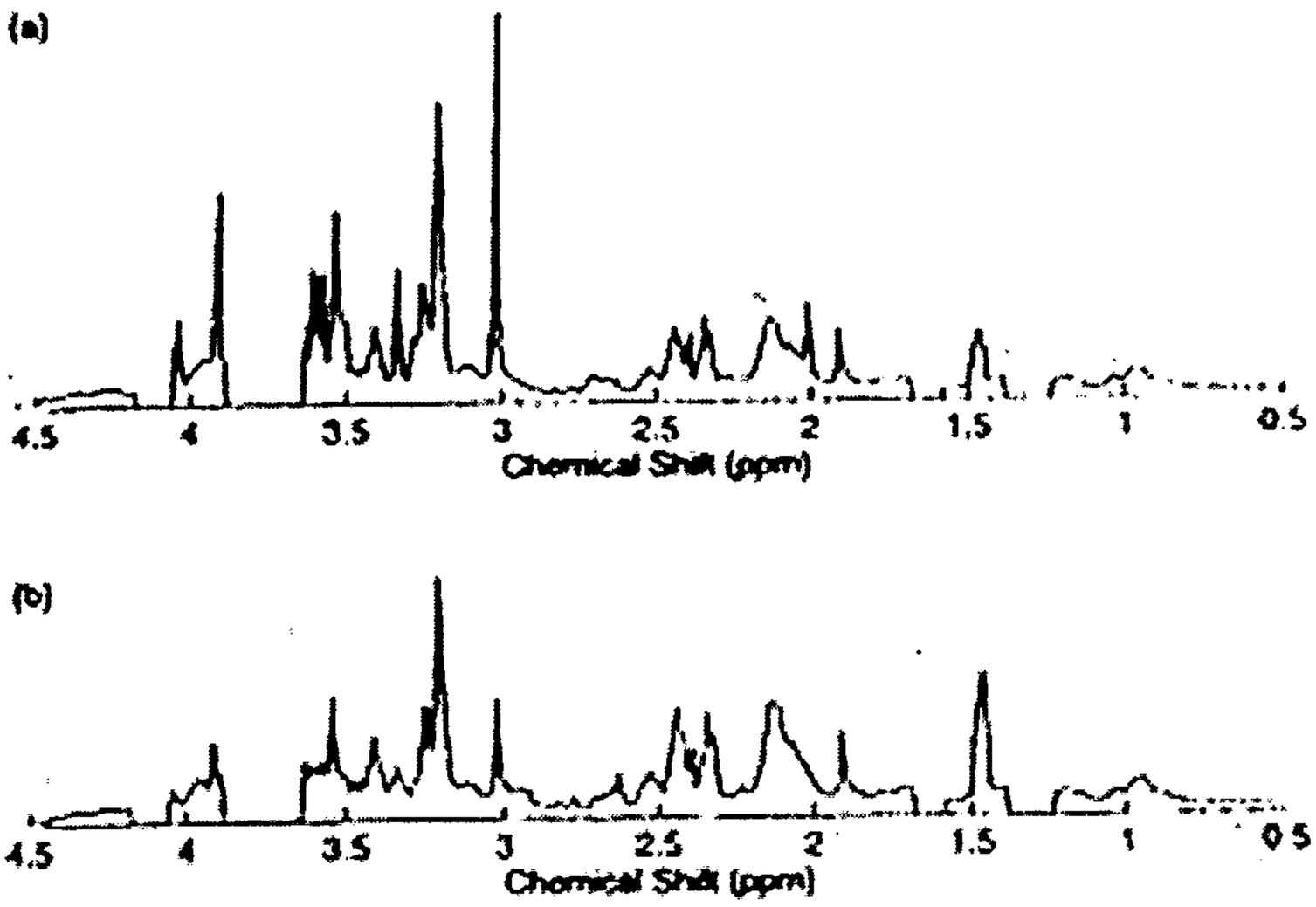


Figure 3.1. The mean spectra from the two main tumour classes, redigitised in 0.01ppm windows. Regions corresponding to lactate, mannitol and around 1.64ppm have been zeroed and spectra normalised to the sum of the remaining variables as described in the text. (a) Non-Meningiomas; (b) Meningiomas

Principal Component analysis showed that the first 20 PCs accounted for 99% of the variance. The PC factors were simplified by varimax rotation. Figure 3.2 shows the cumulative variance of PC factors and those resulting from varimax rotation.

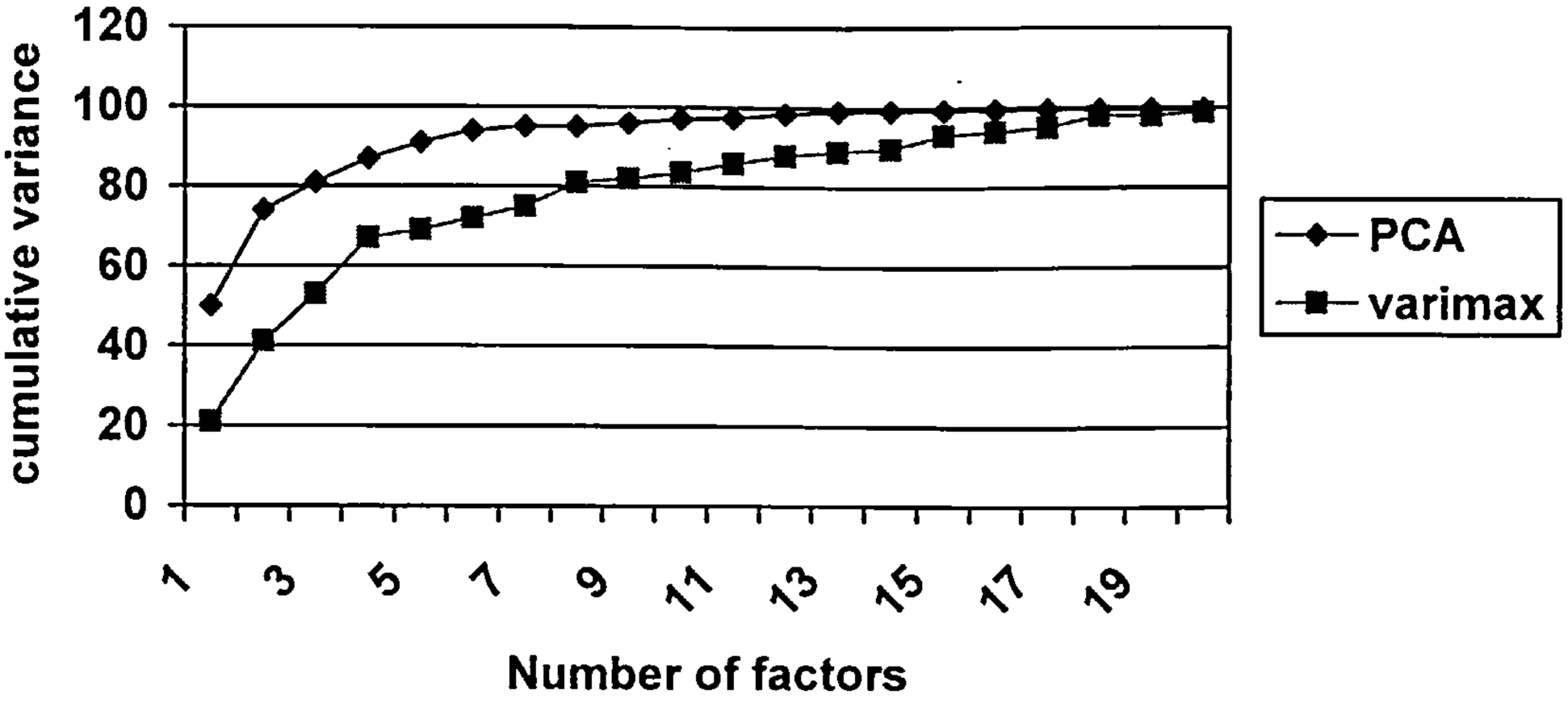


Figure 3.2. The cumulative variance from PC and varimax factors

Several of the varimax factors were very similar to 'H NMR spectra from individual metabolites (e.g. glutamine, alanine, creatine, glutamate, taurine). Figure 3.3 shows such a comparison for alanine and glutamine.

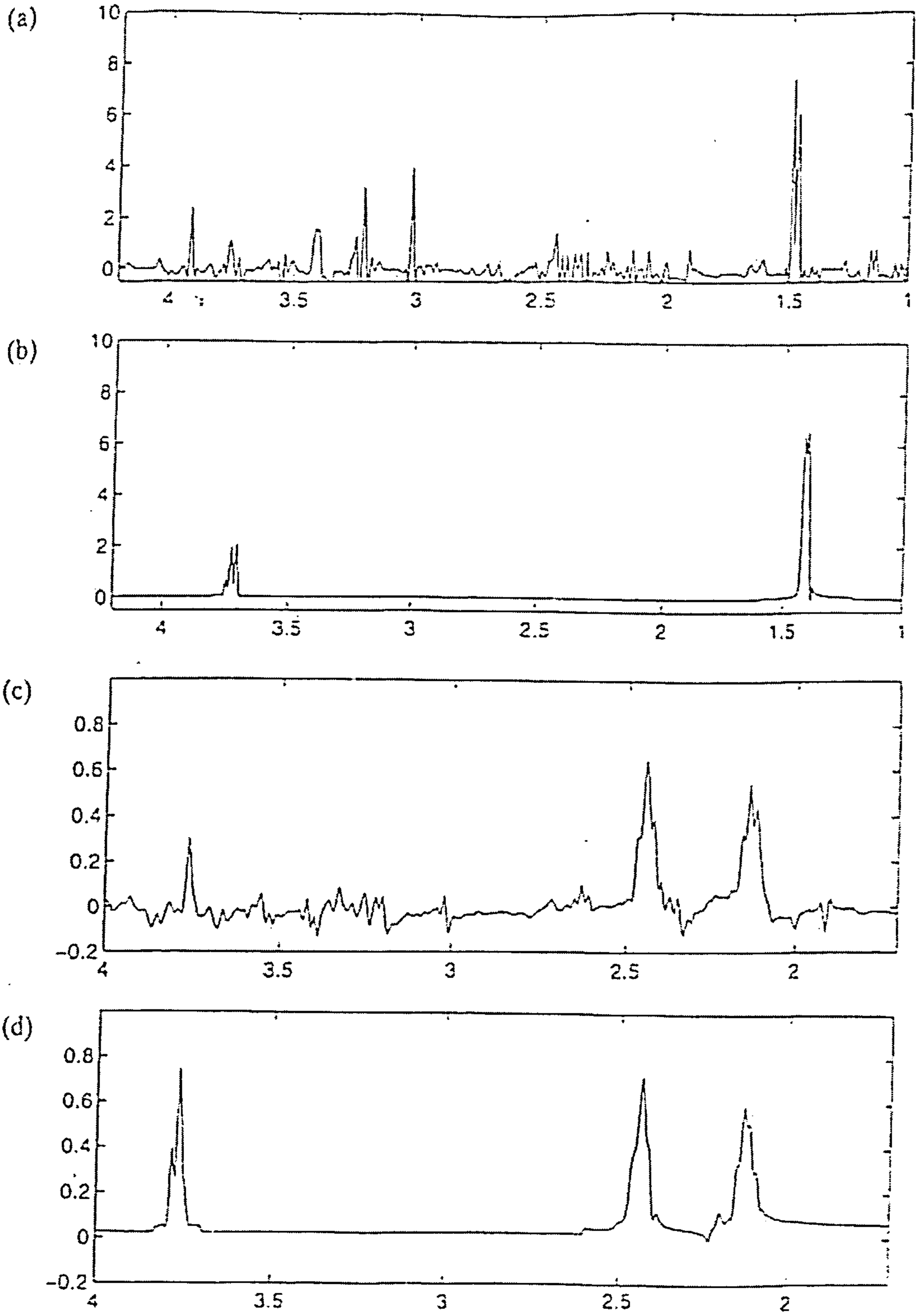


Figure 3.3 Comparison of varimax factors with ^{1}H NMR spectra (re-digitised in 0.01 ppm windows) from pure compounds. (a) Vector 'R'; (b) Alanine spectrum; (c) Vector 'B'; (d) Glutamine spectrum. These spectra are shown before the zeroing of the signal at 1.64 ppm.

Scores were calculated by a dot product of the varimax vectors and spectrum inputs for each sample. This gave 20 scores (varimax scores) for each sample which were used as inputs for the GP and GA classification. The class labels were used as outputs with two possible classes (meningioma = -1; non-meningioma = + 1).

The GP program used was in written in the computer language Lisp, following Koza, (Koza 1992). The terminal set consisted of the 20 varimax scores, labelled alphabetically A-S and U (T being a reserved word in Lisp). The functions used were taken from those shown in Table 3.1.

Function	No. Arguments	Comments
+	2	
-	2	
*	2	
%	2	Protected division, if divisor is 0, returns 1.0
IFLTZ	3	Returns 2nd argument if 1st argument < 0, otherwise returns 3rd argument
tan	1	Standard trigonometric function
myand	2	if both arguments >= 0 return +1 otherwise return -1
myor	2	if one or both arguments >= 0 return +1 otherwise return -1
mynot	1	if argument is negative returns +1 otherwise returns -1

Table 3.1. Functions used in GP

Although, in general, the function set used in GP is problem-dependent, in this case a variety of function types were included (arithmetic, trigonometric, logical and conditional). The arithmetic functions used were standard except for division: a protected division was employed such that division by zero yielded a real number (1.0). This allows for the 'closure' of the function set whereby the output of any

function can be used as the input to another. This is in contrast to strongly typed GP (Montana 1993) where each function and variable has a type and can only be used in places where that type is appropriate. Typed functions can be more powerful but remove some simplicity and elegance of closed GP. Logical functions (myand, myor and mynot, Table 3.1) were included but these were written to accept real valued inputs (and return a real valued result) instead of truth values. Inclusion of a conditional function ('if less than zero', IFLTZ) allowed further variability in the function structure. The minimum function set required to produce good solutions to this problem was not known in advance and so a larger function set than possibly required was made available to the system.

The selection method used was fitness-proportionate whereby fitter individuals are selected more often in proportion to their fitness. The generation method was 'ramped half and half' which creates individuals of different size and depth in the initial random population.

The standardised fitness was taken to be equal to the number of samples minus the number of hits, therefore a low standardised fitness score was better with zero being perfect fitness.

$$SF = N - \text{no. hits} \qquad (1)$$

A hit was scored when the function applied to a sample produced a value with the same sign as the class label. Runs were tried using least mean square as the fitness function. On its own it did not increase population fitness during a run, and used in conjunction with hits it drove the run to premature convergence (i.e. the majority of individuals within the population converged to a sub-optimum solution) and so its use was abandoned. The only parameters, apart from function set, that were changed from run to run were random seed, number of generations and size of population.

The GA used was GAOT (the Genetic Algorithm Optimsation Toolbox) written for Matlab. Each chromosome was of length 21 with 20 input values and one class value as above. Each input gene in the chromosome was initialised with a real value between -1 and +1. These were then each used as a multiplier to the equivalently placed value in the training set example. The values returned were then summed to

give a single result. A hit was scored in the same way as with GP, if the sum had a sign the same as the class label for that example.

The GA used a population of 200 over 200 generations with 20 crossovers and 20 mutations per generation. Selection was by a tournament of size 5. Tournament selection involves randomly picking n chromosomes (where n = size of tournament) and comparing the fitness of each. The fittest is the winner of the tournament.

Three sets of experiments were run on this data. The first set of experiments involved using all 75 samples to train with no testing set. This was done using both GP and GA.

A second set of experiments was run to compare the performance of NN, GA and GP. Ten samples were randomly removed from the original data set for use as a test set. All processing stages were then performed on the 65 remaining samples and the learnt NN or GA or the best GP functions applied to the test set. The NN used back propagation with an architecture of 20 input units and two hidden layers using the C-program described by (Pao 1989). Although two hidden layers were used here, it is now considered common practice to use a single hidden layer (Pinkus 1999; Bishop 2005)

The third GP experiment compared learning performance using varimax scores and PC scores. It was performed using a C- language implementation of GP; lilgp (Zongker and Punch). Ten independent GP runs (with different random seeds) were carried out with a population of 700 and with 50 generations for each set of inputs.

Sensitivity plots for each class were obtained from NN learning as described by (Lisboa, P J G *et al.* 1993). This involves learning, using a NN without a hidden layer, and then examining the weights which have been learned. For NN this was performed using both the 400 'raw' variables and the 20 varimax variables. In the latter case, it was necessary to multiply the sensitivity plot of varimax variables by the corresponding weights [e.g. as in Figure 3.3(a) and (c)] so as to show the sensitivity plot in terms of the 'H NMR spectrum. In an analogous way, sensitivity plots were obtained from GP learning by multiplying the GP function by the relevant

varimax weights (GP was not performed using 'raw' variables). Spectrum sensitivities are shown in Figure 3.4 as the absolute difference between the sensitivity plots for the two main tumour classes, meningiomas and non-meningiomas. NN sensitivity plots show the absolute differences between 'within class' and 'out of class' spectra

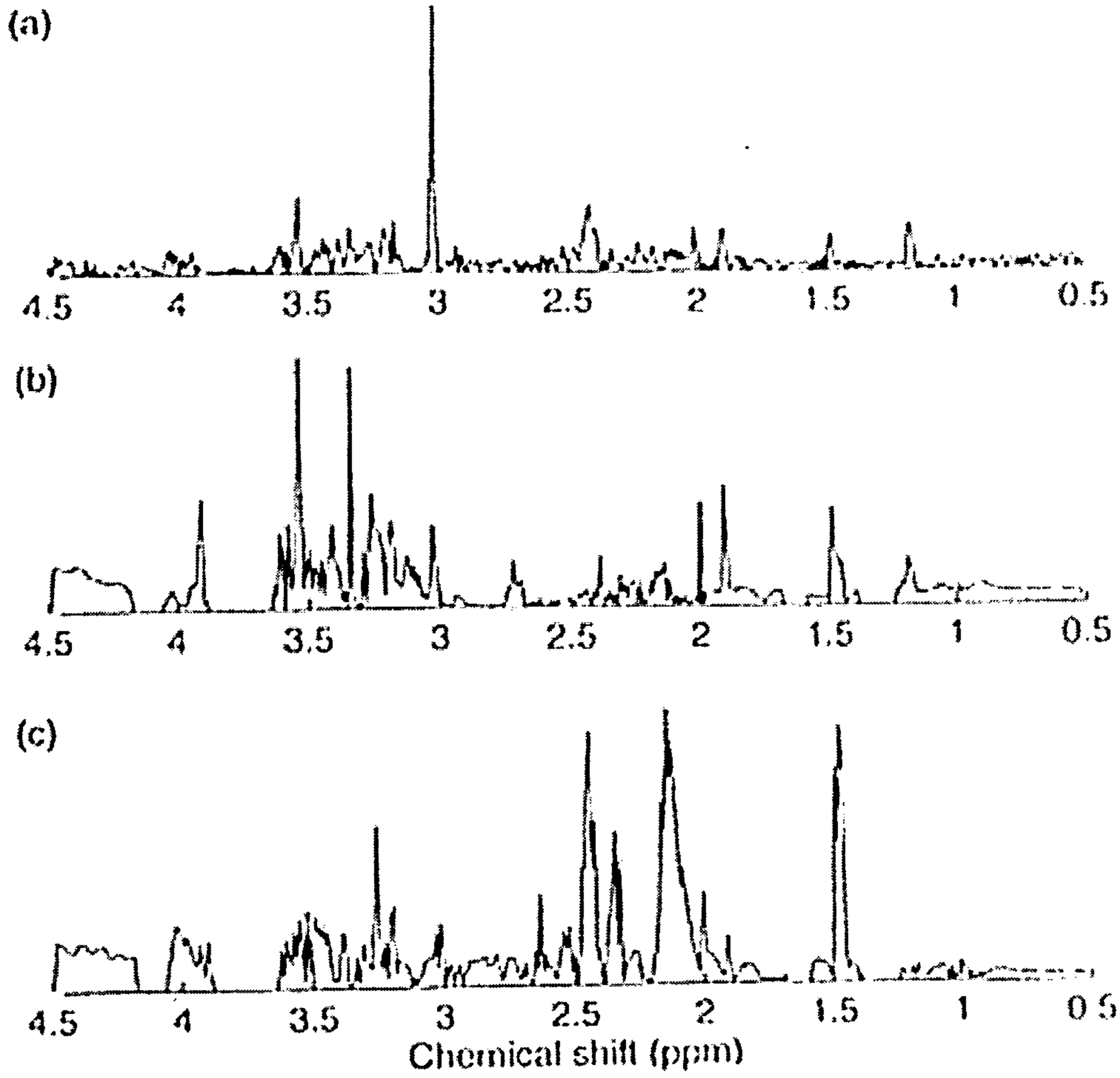


Figure 3.4. Spectrum sensitivity plots for NN and GP learning. (a) NN learning from 400 'raw' variables; (b) NN learning from 20 varimax variables; (c) GP-learned function J – 3B +2R +2K

The feature-extraction capabilities of NN and GP were tested as follows. The best four varimax variables were selected from the NN sensitivity plot and from the GP classification and used as inputs for NN classification in 'leave-one-out' mode using two hidden layers, each of eight nodes.

3.3 Results

On the first GP experiment the best individual classified 73/75 (97%) samples correctly. This was obtained using a function set containing all of the functions shown in Table 3.1. The parameters were random seed of 1.0, number of generations 41 and size of population 700. However, one of the most striking features of the analysis was that good classification (80-85% correct) could be obtained using very simple individuals with few function applications (Table 3.2).

Function (LISP representation)	Standard representation	Biochemical representation	Percentage correct
(-J B)	J – B	$(\text{-Glu})^a$ – Gln^b	80
(- (+ R N) B)	R + N – B	$(\text{-Ala})^a$ +(– acetate)^a – Gln^a	84
(- R B)	R – B	$(\text{-Ala})^a$ – Gln^b	85
(+ (- (+ (- (+ (+ (J B) R) K) B) R)	J – 3B + 2R + 2K	$(\text{-Glu})^a$ – (3 * $\text{Gln})^b$ + (2 * $(\text{-Ala})^a$ + (2 * K)^c	90

[a] Varimax vectors, J, N and R are inverted (along the *y*-axis) compared with the spectra of Glutamine (Glu), acetate and Alanine (Ala), respectively.

[b] Varimax vector B is similar to the spectrum from Gln

[c] Varimax vector K is complex, and has not been assigned to a particular compound

Table 3.2. Some of the most successful functions obtained by GP

Examination of the varimax vectors (Figure 3.3) corresponding to these functions showed that it was actually possible to classify most of the samples based on a very simple combination of metabolites. Vector B corresponds to glutamine (positive weighting) and vector R to alanine (negative weighting): therefore, the function R-B is minus one, times the sum of glutamine plus alanine. Other successful functions also included the glutamine and alanine vectors together with functions corresponding to glutamate (J) and acetate (N). The biochemical assignment of the most useful varimax vectors is summarised in Table 3.3. This information is incorporated into Table 3.2 to give biochemical interpretations of GP functions.

The GA produced solutions that correctly classified 72/75 (96%) of the training cases. The best individual from each of 10 runs which had 72 hits were compared and their mean and standard deviation are shown in Figure 3.5. For comparison, t-tests were performed to determine whether there were significant differences between the varimax scores for each class, as summarised in Table 3.4. The GA solutions show a close resemblance to each other; possibly the minimum which these solutions have found is a large shallow dip rather than a small steep one. Once in this sub-optimum minimum it is hard to get out of. There is no advantage to be gained from moving around in it.

Vector	Assignment
A	Creatine (CH_3 and CH_2) + *N*-acetylaspartate
B	Glutamine (Gln)
F	Creatine (CH_2)
J	Glutamate (Glu)
K	Complex
N	Acetate
R	Alanine (Ala)
S	Complex

Table 3.3. Biochemical assignment of selected varimax vectors

The t-tests show that 80% of the variables are significant to the problem, i.e. where h = 1, there is a very significant difference ($\log_{10}P < -2$) between class means of the varimax scores. These contain those variables which GP finds significant plus others. The number of features that can be shown to have no effect on the classification is small and of little practical use given the number remaining which may have influence over the classification.

GP finds fewer significant features than GA perhaps because some of the initial population of GP individuals are forced to be small (because of the use of ramped half and half) and then can grow. GA individuals are all the same size and all contain

all variables. It is by having the multiplier close to or equal to zero that the feature or variable becomes insignificant. GP can be thought of as starting with a minimum set of features and growing to cover all examples whilst GA (and NN) start with a maximum set and possibly shrink whilst still covering all examples. It is interesting to note that in all the successful GA individuals shown here, it is the same three examples that cannot be classified.

Variable	Variable No. (as in Fig. 3.5)	h	$Log_{10}P$
A	1	1	-7
B	2	1	-7
C	3	0	0
D	4	1	-6
E	5	1	-5
F	6	1	-7
G	7	1	-5
H	8	1	-6
I	9	1	-3
J	10	1	-7
K	11	1	-7
L	12	1	-3
M	13	1	-5
N	14	0	-1
O	15	0	-1
P	16	1	-3
Q	17	1	-7
R	18	1	-7
S	19	0	-1
U	20	1	-3

Table 3.4. t-tests on difference between class means for each varimax score

On the second experiment, which divides the data into training and testing sets, the best GP individual classified 90% correctly on the training set and also 90% on the

testing set. The results were comparable with those from NN (80% correct, 10% uncertain, 10% wrong). (The NN gave two outputs with a target [1 0] for class one and [0 1] for class two. A correct result was defined as [>0.75 <0.25] for class one and [<0.25 >0.75] for class two, a wrong result being the reverse of these and an uncertain result being any other combination.) In this case, the winning GP individual again included varimax variables corresponding to glutamine and alanine.

The individual with the best performance on the test set was found on the first run of the program, with a population size of 200 running for 20 generations. All functions except tan were included in the function set. It is interesting that the best function,

(-(-(-(-(mynot B) (+E B)) R) (+E B)) R)

used only three of the 20 inputs and only three of the available functions.

Other runs produced individuals on the training set which gave a 98% successful classification, but which performed less well (70%) on the test set. The individual that produced this result contained 39 operations on 12 of the inputs, as opposed to the seven operations on three inputs of the best function. In such cases the GP system had become over-trained on the training set, fitting relatively complex functions, leading to a loss of generalization ability.

GA was also used on the data divided into the training and test sets. The individual which performed the best on the test set classified all 10 correctly. However, on the training set four were misclassified. Four other individuals from separate runs classified 62/65 on the training set and 9/10 on the test set. The mean and standard deviation of these 5 individuals are shown in Figure 3.5. Those variables where the mean is near to zero (3 (C), 15 (O) and 19 (S)) in Figure 3.5 are variables which have been shown to be not significant in the t-test of table 3.4. This reinforces the result of the first experiment where GA was shown to have as good a classification ability on this data as NN without having a good feature selection capability.

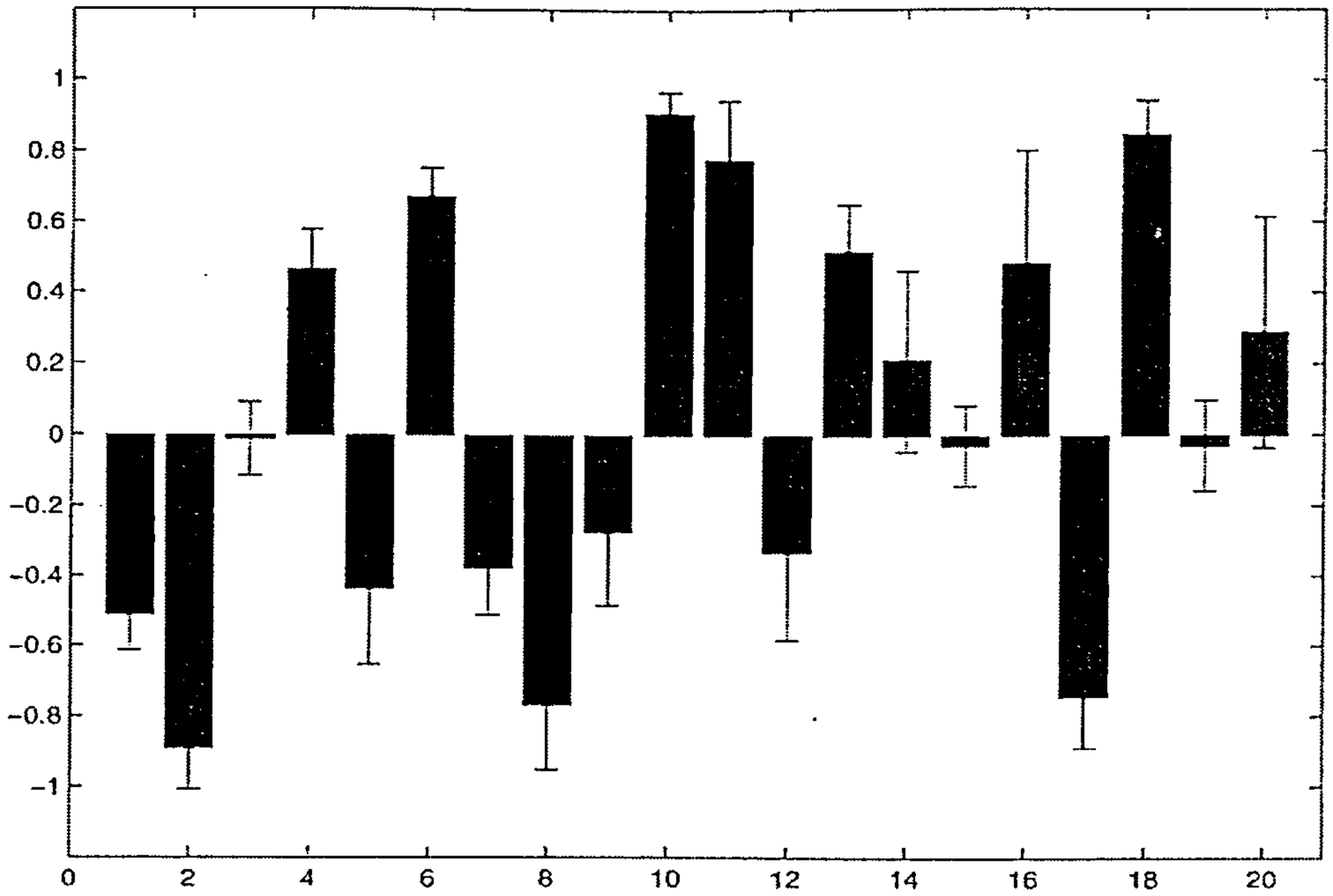


Figure 3.5 Mean and standard deviation of the best GA chromosomes

There was no significant difference in the ability of GP to learn the classification when using PC scores rather than varimax scores (Table 3.5). However, the results suggested that the complexity of the solutions found were greater when principal component scores were used, than with varimax scores (i.e. the tree depth and number of nodes (elements) in each GP program were both larger).

Inputs	Complexity for 85% correct		Complexity for 90% correct		Complexity for 95% correct		% correct in 50 generations (Mean +/- s.d.)
	Nodes[a]	Depth [b]	Nodes[a]	Depth [b]	Nodes[a]	Depth [b]	
Varimax	12	3	25	6	98	11.5	98.1 +/- 1.3
PC	21	5	44	7.5	128	14.5	97.5 +/- 1.2

[a] Number of nodes (elements) in first GP solution giving specified percentage correct during training (median of 10 runs with different random seeds).

[b] Tree depth of first GP solution giving specified percentage correct (median of 10 runs).

Table 3.5. Complexity of GP learning using varimax and PC scores.

Sensitivity plots obtained from NN are shown in Figure 3.4(a) and (b). The sensitivity plot from the 400 'raw' variables was dominated by signals at around 3 and 3.9 ppm, due to creatine. The sensitivity plot obtained using varimax variables was quite different, with significant contributions in the 'choline' region (around 3.2 ppm), the glutamine/glutamate region (2.1-2.5 ppm), from N-acetylaspartate (2.0 ppm) and acetate (1.9 ppm). Figure 3.3(c) shows a sensitivity plot obtained from the moderately complicated GP function, J - 3B + 2R + 2K, which gives 90% correct classification on the training set. This is dominated by signals from glutamine, glutamate and alanine.

Feature selection from NN sensitivity plots gave varimax variables A (corresponding to total creatine and N-acetylaspartate signals), B, F (probably from the creatine CH_2 signal) and S (a complex mixture of signals) as the most important. GP classification selected varimax variables B, J, N and R as the most useful. Only one variable (B, corresponding to glutamine) was present in both sets. When these two sets of variables were used as inputs for NN classification in leave-one-out mode, the GP-selected variables gave 73% correct and 17% wrong, while the NN-selected variables gave 72% correct and 21% wrong.

3.4 Discussion

The use of the factor analysis techniques of PCA and varimax rotation for reducing the dimensionality of the data greatly facilitated the implementation of GP. Since each variable must be assigned a name and is then randomly allocated to the initial set of programs, the inclusion of many more variables would make it increasingly cumbersome to set up a GP run and to interpret the results. It is also likely that any successful programs would be considerably more complicated. The main risk with using factor analysis is that information contained in the unused factors (e.g. principal components or varimax vectors) is lost. However, this is unlikely to be the case so long as enough factors are used (i.e. accounting for a large proportion of the variance in the data set). Indeed, optimal choice of the number of factors might reduce the noise in the data, without significant loss of information.(Malinowski 1987). In the present study, 20 varimax vectors, accounting for more than 99% of the variance, are used. In general, the use of varimax rotation does not guarantee that the spectra of individual metabolites will be separated in a pure form but quite 'clean' results are often obtained from datasets of this kind for about 6-10 metabolites. Further improvements have been achieved using an additional, non-orthogonal, rotation method (promax): this subject is discussed in more detail elsewhere (Maxwell *et al.* 1998).

Varimax rotation gave an advantage (over PCA alone) because the rotated vectors could be more easily associated with individual metabolites, thus enabling interpretation of the GP results in biochemical terms. Although the use of varimax scores rather than PC scores gave no significant advantage in terms of GP learning (on the training set), the tendency for the solutions to be less complex (with varimax, Table 3.4) would be expected to lead to better generalization performance and to further simplify the interpretation. It also implies that the arrangement of information into simplified vectors (often corresponding to individual metabolites) was more natural for classification purposes than the original PC vectors, which are arranged according to the amount of variance that they account for.

The use of sensitivity plots (Lisboa, P J G *et al.* 1993) [Figure 3.4(a) and (b)] showed that it was possible to identify which features were of most importance in a given NN classification, at least for a NN without hidden layers. (Although it is also possible to obtain sensitivity plots for NN with hidden layers, this has not been implemented in the current study.) A substantial difference was noted depending on whether 400 'raw' variables or 20 varimax variables were used as inputs. It is a straightforward task to reconstruct a sensitivity plot from a GP function [e.g. Fig. 4(c)]. The GP sensitivity plots were consistently different from NN sensitivity plots obtained using the same inputs (i.e. 20 varimax variables). This is consistent with the different ways in which GP and NN solutions are obtained and also with the different errors in classification reported by Somorjai (Somorjai *et al.* 1995). Some care should be taken in interpreting these differences given the comparison between NN sensitivity plots obtained from 'raw' variables and from varimax scores; i.e. they also show big differences even though it is expected that the information content is approximately the same in both cases

The differences between NN and GP methods can be summarised as follows:

(1) For this two-class problem, satisfactory learning of how to classify the training set was achieved with both methods. It is difficult to compare the computational effort since the two methods were written in different languages and run on different computers. However, the GP implementation appeared to be more demanding.

(2) It was very important to reduce the dimensionality of the data for the GP (LISP) implementation. It was convenient, but not essential, to do this for the NN approach.

(3) Classification results on the small test set were at least as good for GP as for NN.

(4) In the present study, GP gave relatively simple functions which made feature selection an easy task.

(5) Features selected by GP gave at least as good classification as those selected from NN sensitivity plots, when they were used as inputs to a NN classifier running in leave-one-out mode.

The experiments using GA also used the varimax scores as inputs. This allowed the information about metabolites to be utilised but was not so crucial a decision as GA has a fixed-length chromosome. The disadvantage of this fixed length proved to be

that more features were considered significant; the simplest solutions were not found in this implementation..

3.5 Conclusion

Good classification of human brain tumours, based on ^{1}H NMR spectra of biopsy extracts, could be obtained using a GP approach. In addition, the most significant aspect of the analysis was that very simple functions gave classification results that were almost as good as the 'best-ever' functions. The use of principal component analysis followed by varimax rotation meant that many of the inputs to GP corresponded to 'H NMR spectra of individual metabolites. Combinations of metabolites, such as glutamine plus glutamate or glutamine plus alanine, were found to give approximately 85% correct classification (for the training set) between meningiomas and non-meningiomas.

It should also be noted that even a single GP run can provide a large number of good solutions (of varying complexity) and hence there will be a variety of possible GP sensitivity plots. In future applications, it may be advantageous to consider a whole population of GP solutions rather than just one or a few of the best ones.

The information contained in *in vivo* 'H NMR data is dependent on the specific acquisition conditions employed (e.g. pulse sequence and echo time). The importance of glutamine in this classification suggests that if *in vivo* 'H NMR data are to be used to distinguish between these tumour types, then short echo time (30 ms or less) acquisitions (sensitive to glutamine and glutamate signals) will provide better discrimination than long echo-time data, either at 1.5 T or at lower field (0.5T)(Prost *et al.* 1997). In addition, it should be possible to use GP to give feature selection for other classification tasks: e.g. to determine brain tumour grade from *in vitro* or *in vivo* 'H NMR spectra. Unfortunately, the size and composition of the present data set are not adequate to permit this.

Chapter 4

The use of Genetic Methods to Classify into Multiple Classes

4.1 Introduction

The use of evolutionary methods to classify MRS data into two classes has been proved to be useful as shown in the previous chapter. In order, however, for GP to become a technique of choice for classification in this and other spheres it needs the ability to divide data into the required number of groups, whether this is two or a larger number. NN techniques have this ability (an example is shown below). The advantage of GP over NN in classifying into multiple categories is the same as for the two-class problem – as a non-black box solution, those variables used in the classification are easily identified. This may help in focusing data collection so that signals from important features can be enhanced.

Classification of data from MRS has been carried out using NN (Branston *et al.* 1993; Lisboa, P J G *et al.* 1993; Usenius *et al.* 1996; Ala-Korpela *et al.* 1997; El-Deredy *et al.* 1997; Bakken, I J *et al.* 1999; Gribbestad *et al.* 1999; Poptani *et al.* 1999; Bakken, I *et al.* 2001; Axelson *et al.* 2002; Hiltunen *et al.* 2002). NN appears to be able to cope with multi-class classification with ease. The number of output nodes on a NN can be freely set with the usual case that each output is used to classify one class. There would need to be processes that can handle the case that more than one output node is activated or than none are. On the data set used here a NN with one hidden layer of 40 nodes classified 47/49 samples correctly, with one of the remaining 'probably correct' and the final 'uncertain'.

Early work in GP classification focused on dividing the data into two categories. If greater numbers of categories were required, repeated 'one versus the rest' was applied. This is also the case with other classification methods (Tate *et al.* 1998). More recent work has examined ways of dividing data into multiple classes. Kishore's work set the scene for a new interest in more complex classification tasks (Kishore *et al.* 2000; 2001). More recent work has explored the use of classification with images (Zhang *et al.* 2003; Zhang and Smart 2004; Smart and Zhang 2005) and using GP with a grammar to produce rules that can be applied to data in order to classify it (Dounias *et al.* 2002).

Once there are more than two target classes the relationship between those classes is necessarily more complex than with two classes. With binary classification the boundary between the classes is usually clear-cut. In the previous chapter the boundary between the two classes was set to be zero, a common convention. The classes were separated by the result of a function application to data returning a positive or negative result.

With multiple classes the relationship between classes may be of several forms. Some classes may be sub-sets of others, a class may have a closer relationship with one other than with a third, or there may be no relationship at all depending on the data, the variables used to describe the data and the representation of the class. The relationship between these classes may be known. It is possible to use techniques such as cluster analysis (Tate *et al.* 1998) or Sammon mapping (Sammon 1969) to indicate relationships.

If the data is thought of as linear, with the planes dividing the classes parallel to each other, then a restriction of values for the classes that have fixed boundaries on both sides will contrast with classes that have only one fixed boundary. It is possible to visualise a division of the solution space into three classes with each having the same relationship to both other classes in a two dimensional space, but an increase to four classes involves a different relationship between some classes than between others – (see Figure 4.1). A two dimensional model with four classes fits neatly into a Cartesian co-ordinate grid where the classes are divided by two perpendicular planes. The values can be thought of as (x,y) co-ordinates and is shown in Figure 4.1b.

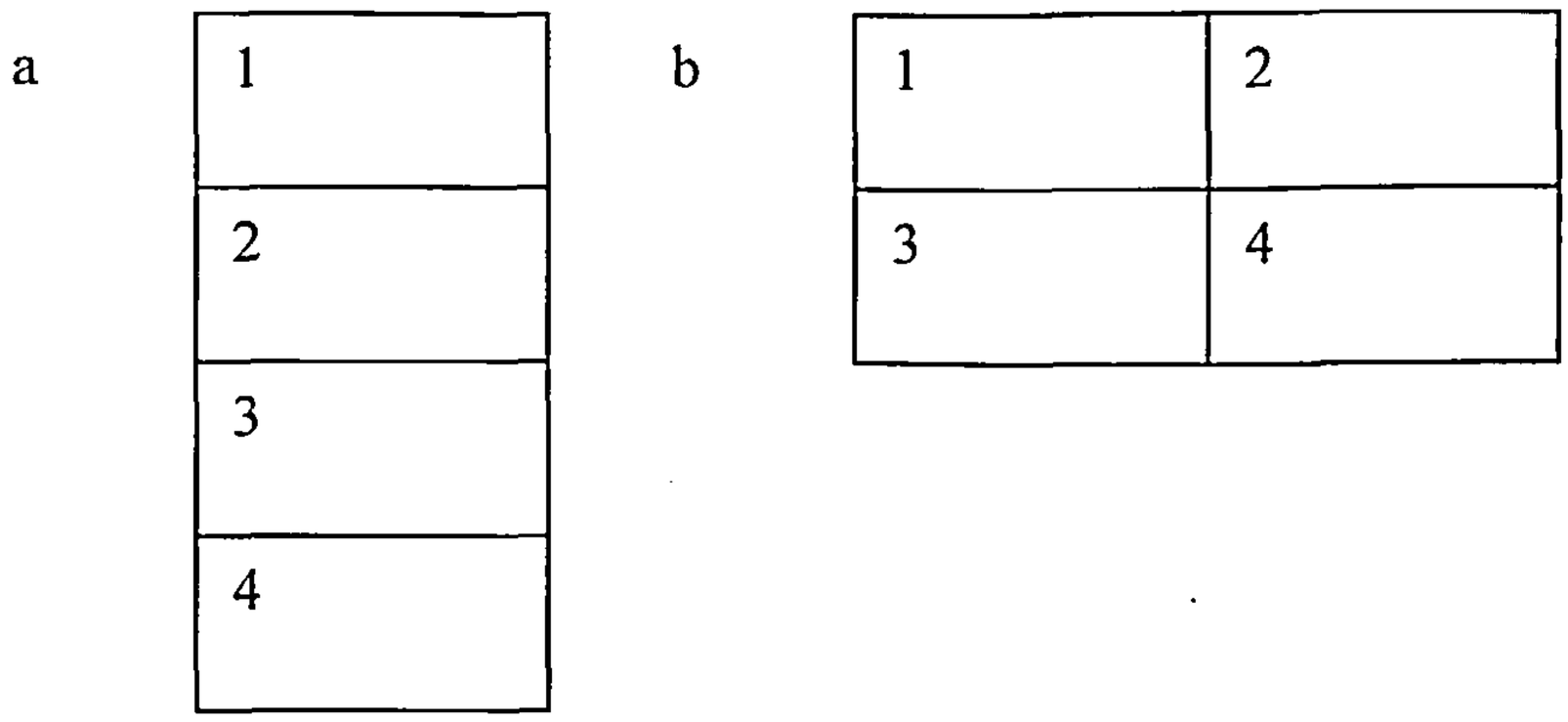


Figure 4.1. Relationships between classes. a shows a situation where classes have different relationships with each other, b shows the same number of classes with each class having a boundary with each of the others.

It is possible to have categorical data in which the classes have no relationship to each other and non-membership of one class does not help in classifying it into another. If the classes were hats, coats, gloves and shoes, non-'hatness' would not help in assigning class gloves, coats or shoes (depending on the variables chosen to describe the data). On the other hand with classes of apples, pears, oranges and lemons non-'orangeness' may help in deciding between other classes (again dependent on data variables used). The division of fruit into two classes (citrus and non-citrus) each divided into further classes is possible. Decision trees exploit this by dividing and subdividing the data into classes.

Many data sets assign categorical data as variables, others have range data. It is possible to move range data into categorical data – age can be given in a range or in categories such as 0-17, 18-25 with a yes/no value to each category. This will lose some information contained in the range but could be more useful in some circumstances.

In the tissue/tumour samples used in this chapter the values of variables are in a range and it is the meaning of various parts of that range, combined with ranges of values from other variables that drive the classification.

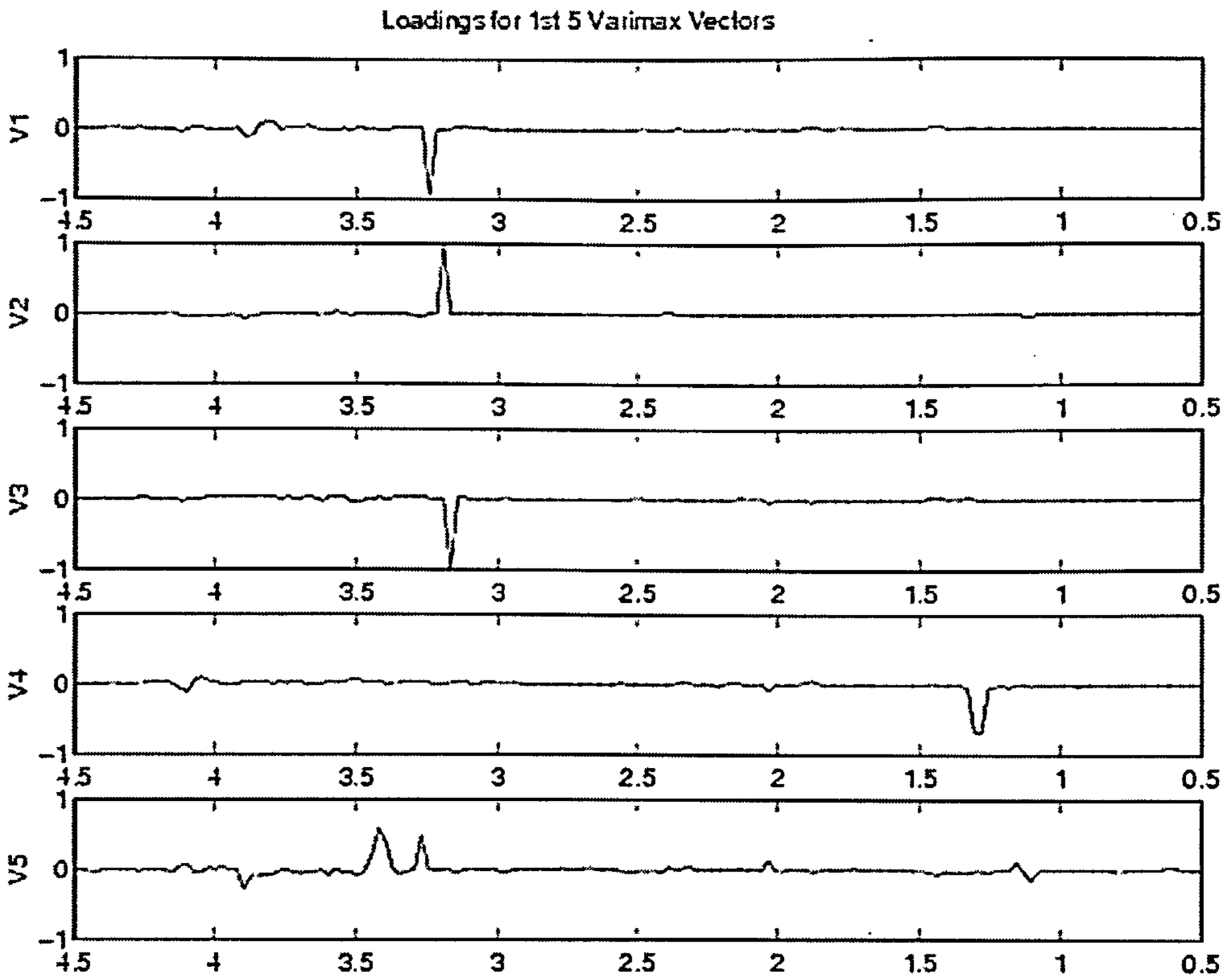
If there is no relationship between classes, and even where there is, one method of multi-class classification is through a set of 'one versus the rest' binary classifications either in parallel or in series. NN do this where the number of output nodes is equal to the number of classes. The weights on the connections are a combination of weights that allow a series of 'one versus the rest' classifications to occur. However, examining the weights and connections and disentangling the relative importance of the different classes to that weight is not a trivial task.

A general multi-class classification system would be able to cope with all types of relationship. In order to find a general classification tool with GA/GP it is necessary to examine various methods. The data used here is a set of MRS measurements taken from the liver, kidney and two types of hepatoma (a liver tumour) of rats.

4.2 Data and Experimental Setup

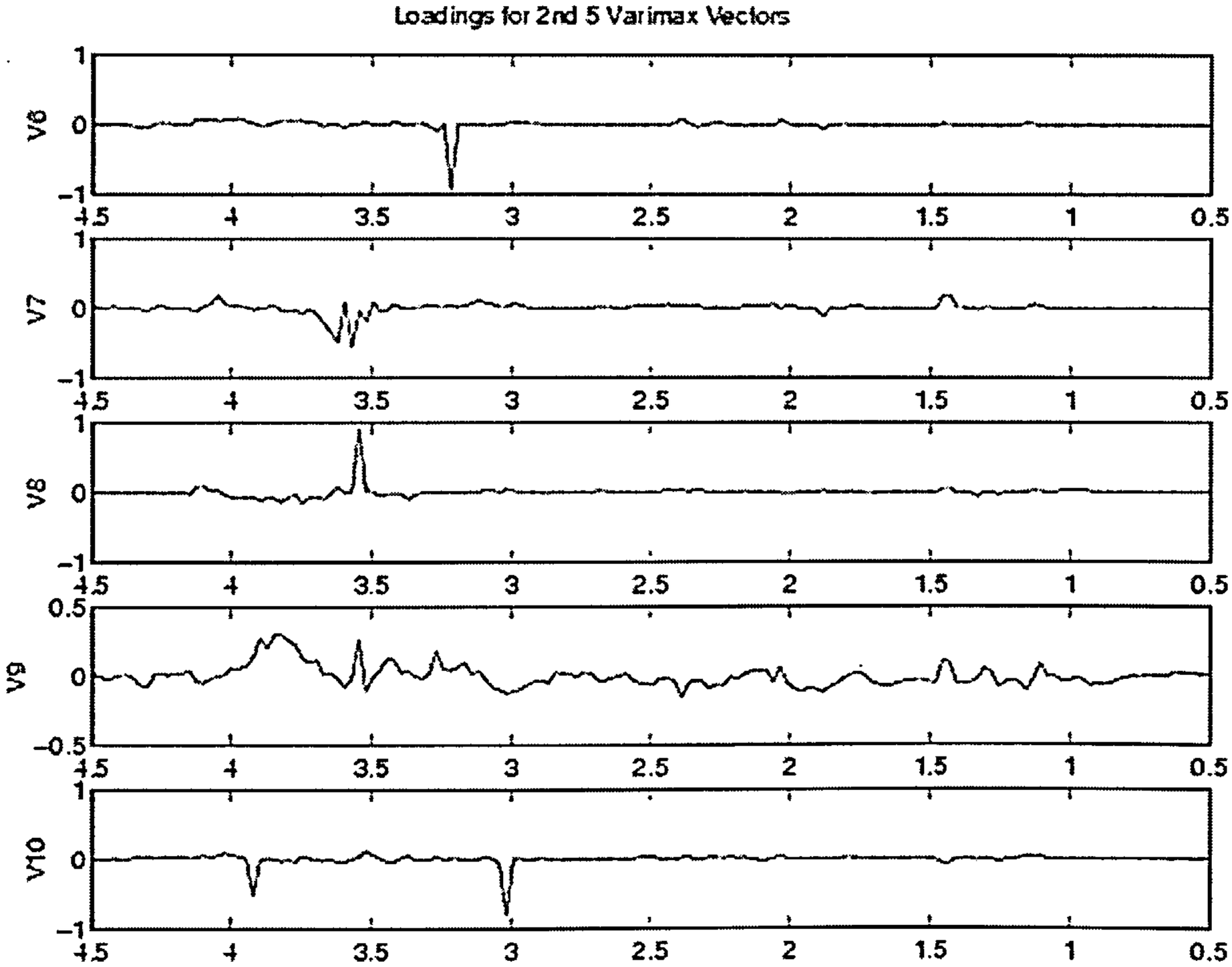
The tumour samples were acquired from biopsies of tumours implanted subcutaneously in the flanks of rats. The tissue samples were acquired from the same rats. Details of extraction and spectroscopy performed can be found in [Howells 1992].

The pre-processing involved PCA followed by varimax. 20 PCs were used, accounting for the majority of the variance. The PCs were labelled A – V (omitting G and T as these were used elsewhere in the program). Figure 4.2 shows the varimax vectors with biochemical interpretations where possible. Some of the vectors show a complex spectrum containing signals from more than one metabolite. It was felt that these vectors could still contain useful information that could aid the classification and were not deemed inappropriate solely because they could not be readily interpreted.



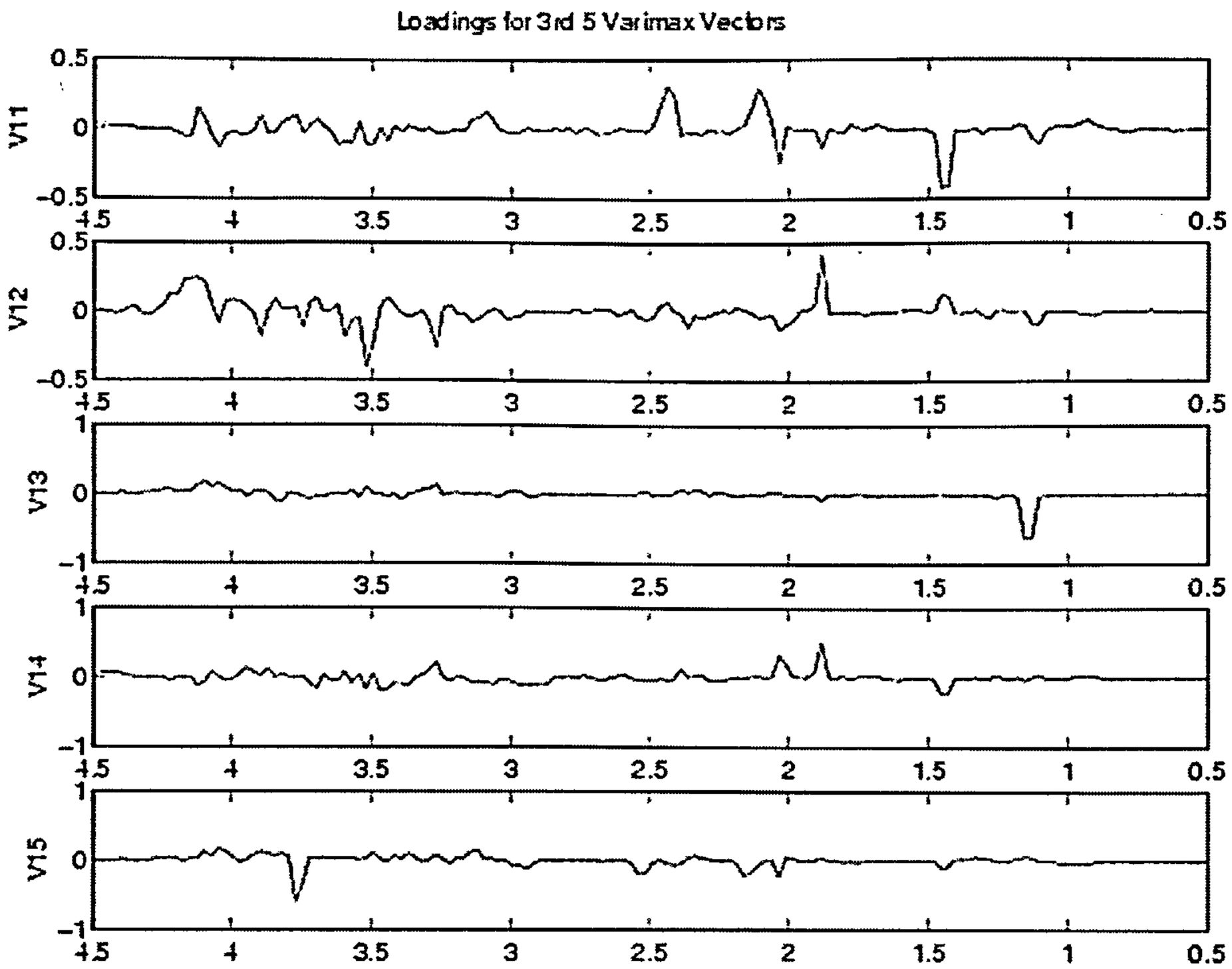
A (V1), B (V2) and C (V3) contain peaks in the choline region

D (V4) lactate, E (V5) is predominantly taurine



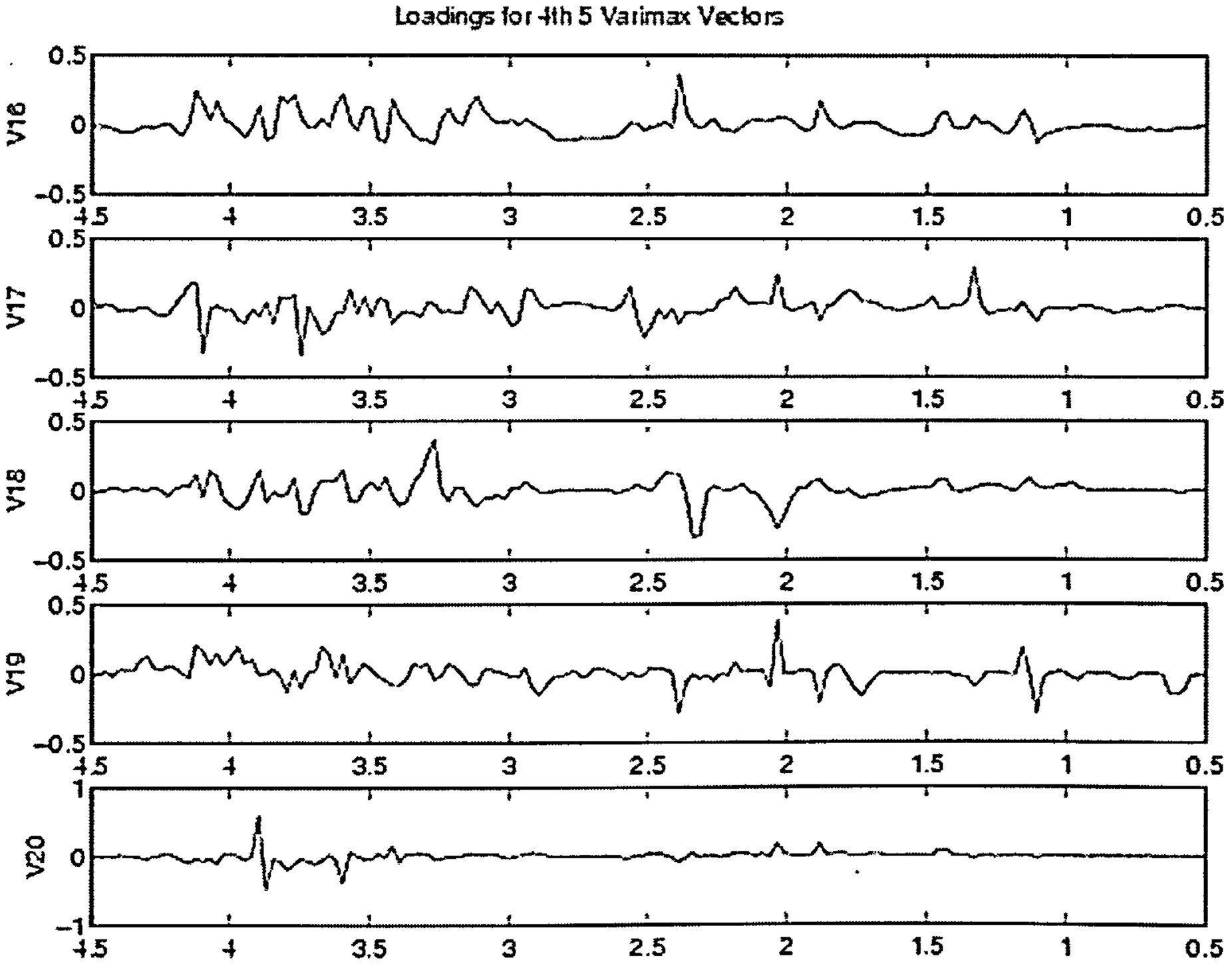
F (V6) choline, H (V7) not interpreted, I (V8) myoinositol, J (V9) not interpreted,

K (V10) creatine



L (V11) alanine plus glutamine/glutamate

M (V12), N (V13), O (V14), P (V15) are not interpreted



Q (V16), R (V17), S (V18), U (V19), V (V20) are not interpreted

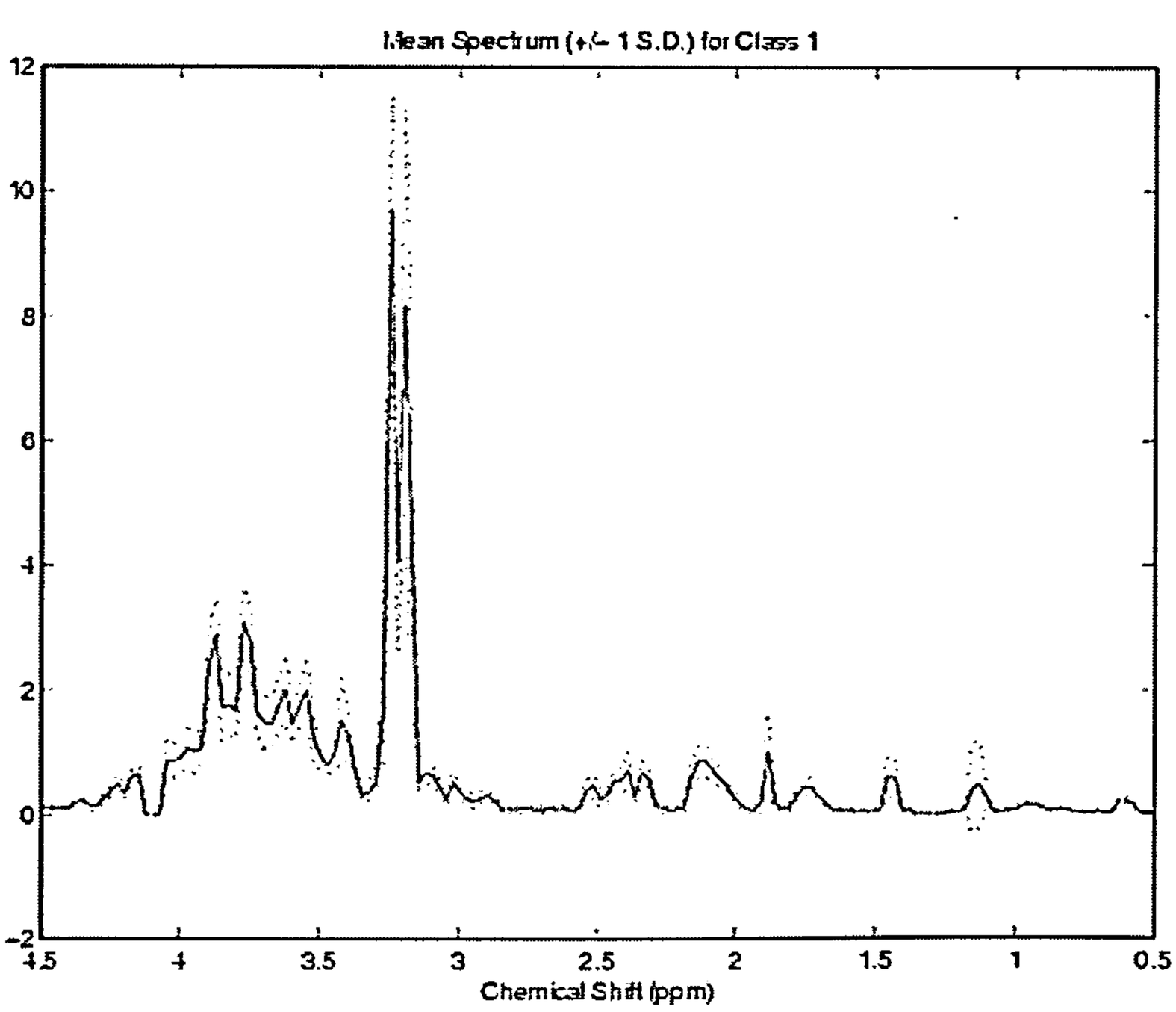
Figure 4.2 Varimax spectra for the tumour and tissue data

The data set involved 49 samples divided as in table 4.1. Each sample consisted of 20 input values, corresponding to the varimax vector scores, plus a value or set of values giving the class of the sample. Figure 4.3 shows the mean spectra for each class.

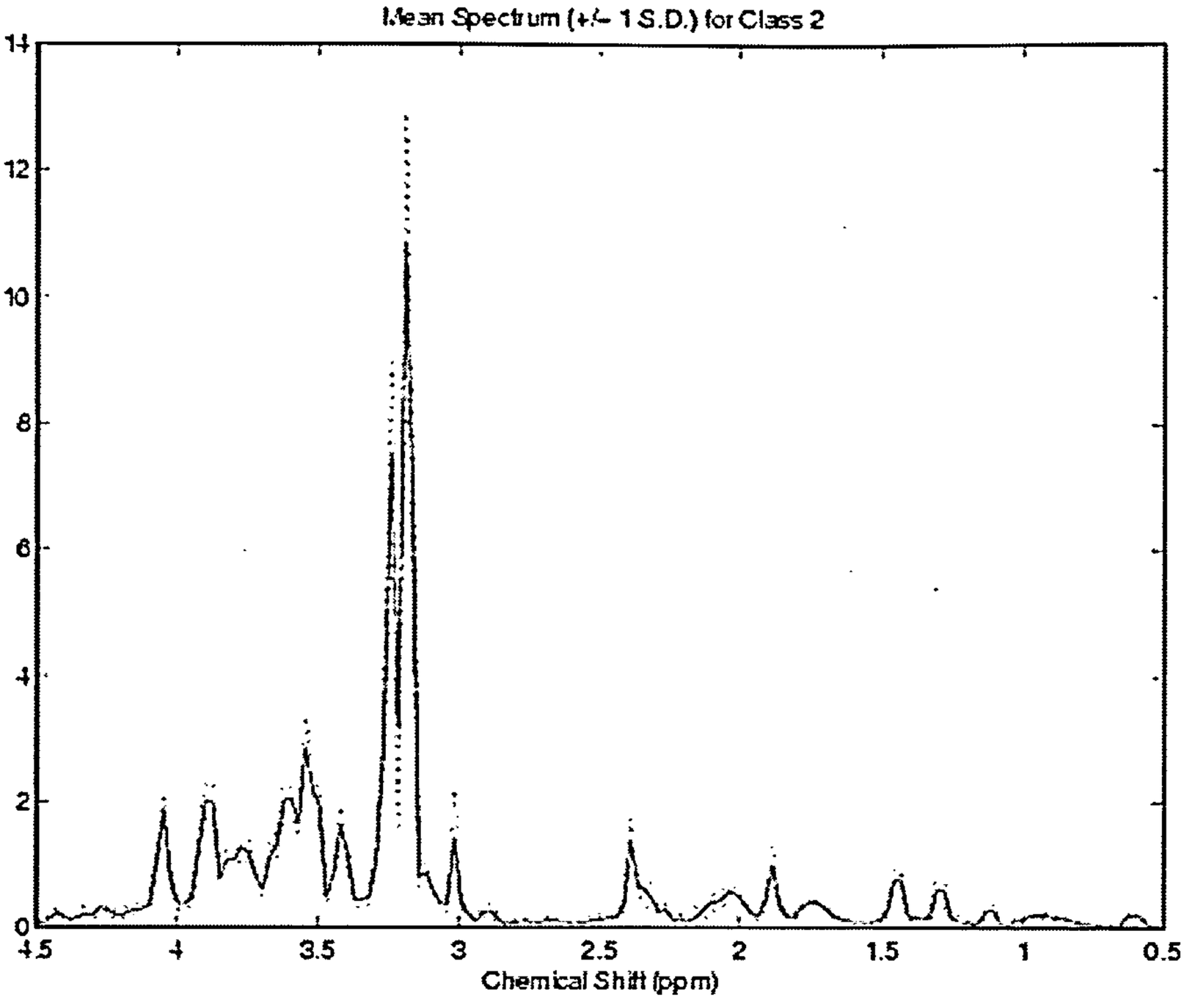
Class	No. Samples	Comments
Liver tissue	13	
Kidney tissue	13	
Hepatoma 7777	12	Liver tumour, rapidly growing, poorly differentiated
Hepatoma 9618A	11	Liver tumour, slow growing, well differentiated

Table 4.1. Number of samples in each class

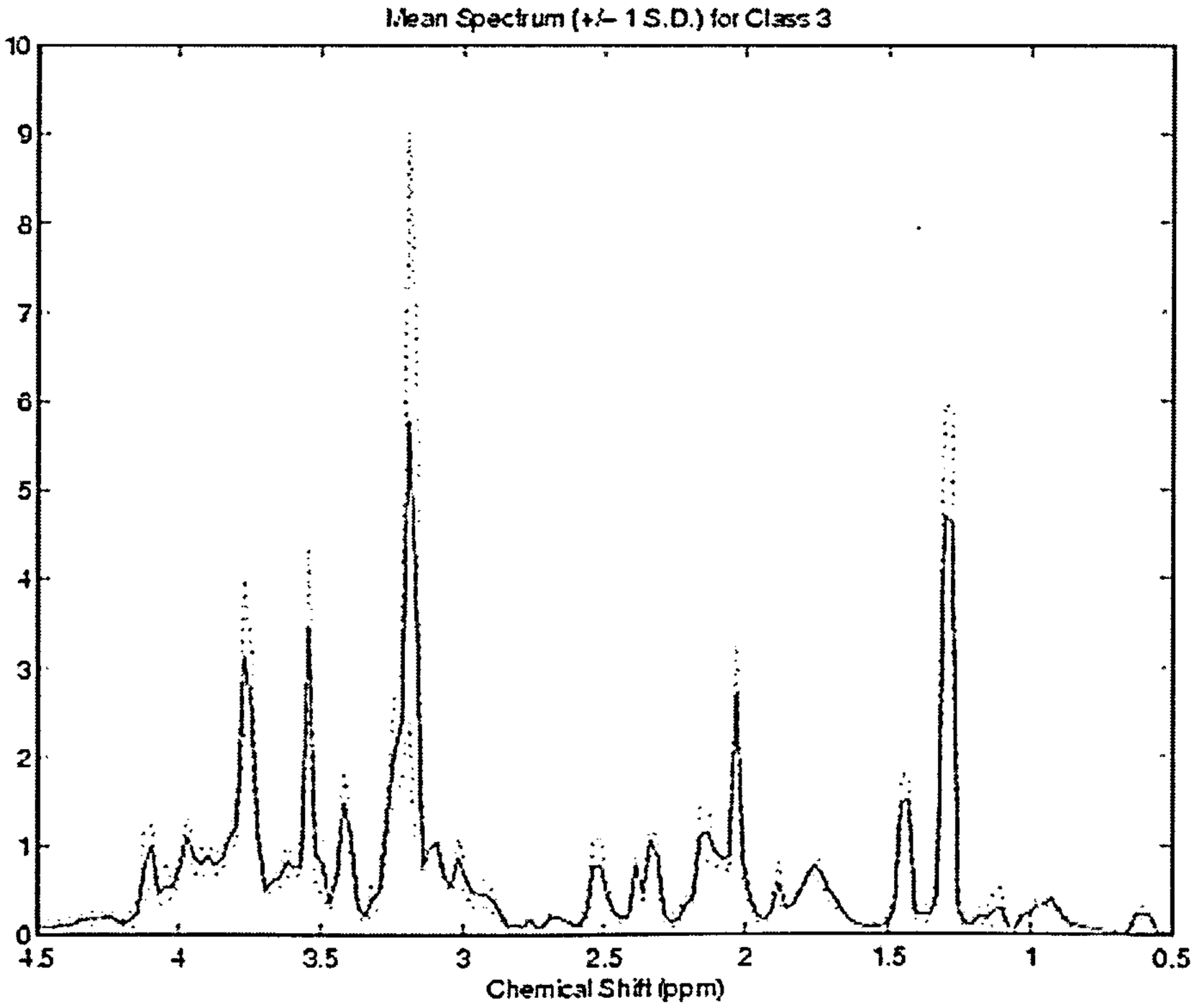
4.3a. Liver



4.3b Kidney



4.3c Hepatoma 7777



4.3d Hepatoma 9618A

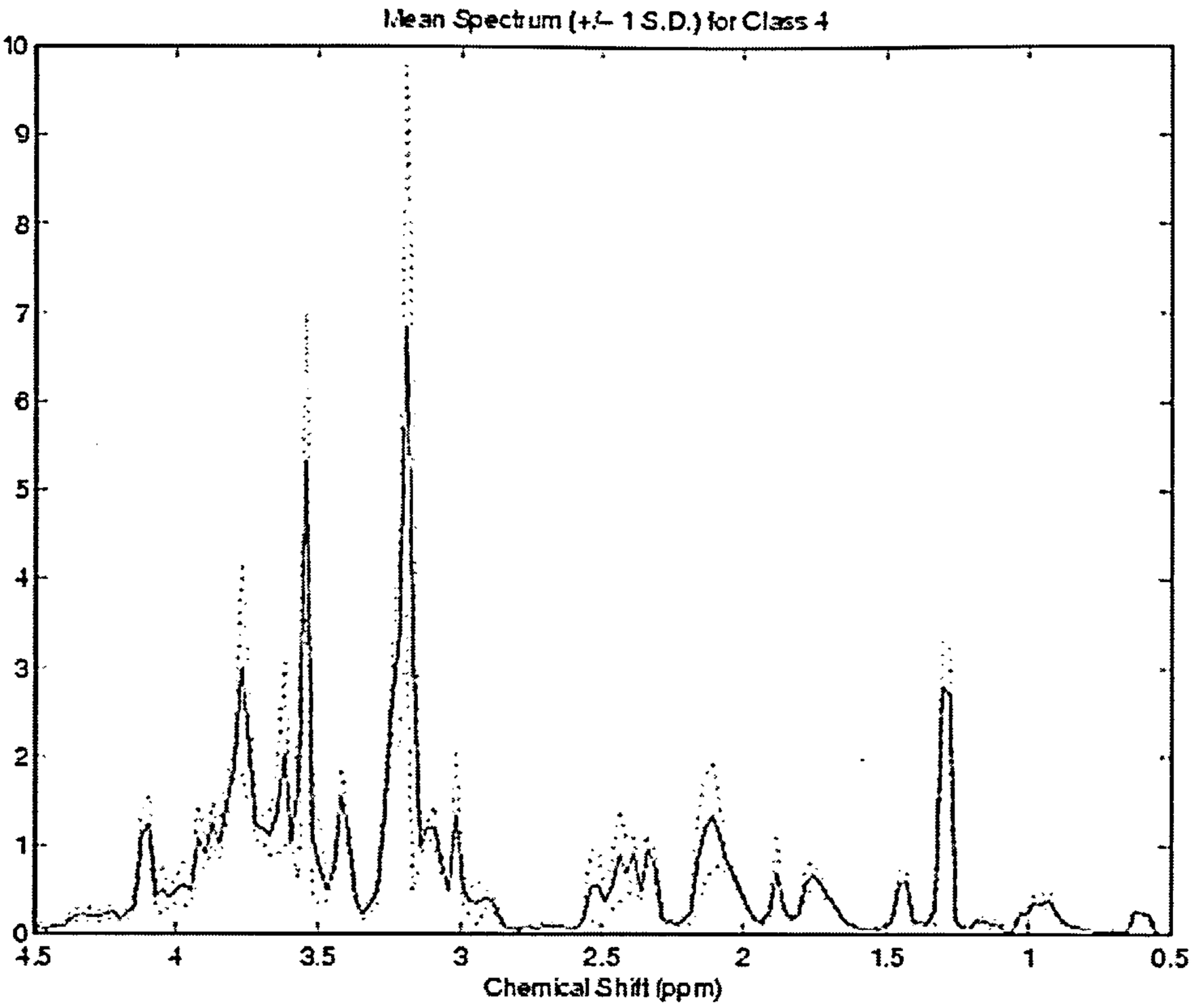


Figure 4.3 Mean spectra for each class in the tumour and tissue data

The GP program used was lilgp which was developed at the Michigan State University Genetic Algorithms Research and Applications Group (MSU GARAGe). Lilgp is written in C; the switch from Lisp to C was made as the lilgp system allowed more flexibility. In each of the following methods the input file was read into a two-dimensional matrix in lilgp with the input values then read into variables labelled A - U (excluding G and T for implementation reasons). Output values were read into the variable result (a numerical value) or result[a b ...] (an array of numerical values).

The function set used in all runs is shown in table 4.2. One of the aims of the GP classification is to obtain a clear description of the features that drive the classification and a simple function set should facilitate that. As this function set was sufficient for the binary classification in the previous chapter it is also used here.

A reduction in functions from this set to plus and minus only proved disadvantageous.

Name of operator	Comments
+	
-	
*	
%	Protected division

Table 4.2. Function set used for GP multiclass runs

In all the methods described below, the complete set of 49 samples was used as a training set, with no testing set. Therefore, the methods are compared in terms of their ability to learn classification rules rather than in terms of generalisation. There is a danger of over fitting the data to the actual samples, so the resulting classification merely produces a function which divides this sample. This is a problem with small data sets where taking out a test set may lead to too few samples to learn from. If there are too few examples of each class the generalization ability of the classifier is also impaired. The termination conditions were either the number of hits equal to the number of samples (which would also lead to a standardised fitness of zero) or the number of generations run equal to a user-set maximum.

Different methods involving GP were used to try to classify the data set into four classes. Three methods of using GP on the multi-class classification were used; a one-tree method, a four-tree method, and a two-tree method with the class depending on the result from both trees. Each will be described.

As a control the data was divided into a series of binary classifications which were run as in the previous chapter. This was used to ensure classification could occur with this data and GP. This initial classification is detailed first.

4.3 Methods

4.3.1 Initial Binary Method

To ensure that GP could cope with the data an initial binary classification was carried out. Each class was involved in a one versus the rest classification. Other divisions, involving tumour/tissue, liver/kidney and Hepatoma 7777 /Hepatoma 9618A were also carried out.

4.3.2 One Tree Method

Each individual in the GP run consisted of a single tree returning a single numerical result. This result was then constrained into a value in the range -1 to +1 using the cos function. Four equal divisions were made in this range (-1 to -0.5, -0.5 to 0, 0 to +0.5, +0.5 to +1) and the four classes required (liver tissue, kidney tissue, Hepatoma 7777 and Hepatoma 9618A) were allocated to the divisions. The order in which the classes were allocated to divisions appeared to have no significant effect on the results obtained. The class marker from the evaluation of the tree was then compared to the result value from the input file.

A hit was scored when the two compared results were the same. Any other value was ignored - there was no linear relationship between the classes, meaning a sample from class 1 was equally misclassified whether it was misclassified as class 2, 3 or 4.

Raw fitness was calculated as the number of hits. Standardised fitness was calculated in the following manner

$$SF = 1 - (RFi / N) \qquad (1)$$

Where SF = standardised fitness

RFi = raw fitness

N = number of samples

4.3.3 Four Tree Method

A straightforward way of approaching the classification is to set the number of trees in an individual equal to the number of classes in the data. Hinchcliffe (Hinchliffe

et al. 1996) calls this the multi-gene approach. In effect, this classified one class against the rest with all four classes being classified concurrently. Each tree was evaluated in turn and the result constrained to the value 1 or 0 using the following step function

$$\text{Constrained result} = \begin{cases} 1 & \text{if result} >= 0 \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

The result values in the input file were given as a set of four values (each 1 or 0) so the output from a sample in class four would be given as [0, 0, 0, 1]. The first of the set was compared with the constrained result from the first tree, the second with the constrained result from the second tree and so on. A part-hit was recorded if one of the trees of an individual produced a result the same as that given in the input file. A hit was recorded if all four trees of an individual scored part-hits. Raw fitness was calculated as follows

$$RFi = N * T^2 \qquad (3)$$

Where RFi = raw fitness

N = number of cases

T = number of correct trees per individual

The quadratic method was employed give a stronger drive towards fit individuals as it rewards more highly those individuals with a greater number of part-hits. Standardised fitness was calculated as follows

$$SF = 1-(QRF/N * C^2) \qquad (4)$$

Where SF = standardised fitness

QRF = quadratic raw fitness

N = number of cases

C = number of classes

4.3.4 Two Tree Method

If a binary classification occurs with a boundary on a line, then using two dimensions each with a fixed point on a line as a boundary and the lines perpendicular to each other could allow classification into four classes. These can be represented on an x-y grid with each quadrant representing a class. In order to maintain the notion of an individual as a function which can be applied to various

inputs, two trees per individual were required to produce the two parts of the result. The training set was amended to give the classification system shown in table 4.3. For each training sample, each tree returned a result which was constrained as in equation 2 and then compared to the result in the training set. A part hit was recorded when one tree returned the correct result; a hit was recorded when both did. Raw and standardised fitnesses were calculated using formulae 3 and 4.

Class	Result values
Liver	+1 - 1
Kidney	+ 1 + 1
Hepatoma 7777	-1 - 1
Hepatoma 9618A	-1 + 1

Table 4.3. Classification values for two-tree method

4.4 Results

4.4.1 Initial Binary Method

This coped with ease, classifying into two groups in each case. The parameters for each run are shown in Table 4.4.

No. Runs	20
Max. No. Generations	150
Population size	500
Random number seed	1 – 67
Selection method	Tournament (size = 7)
Crossover	0.8
Mutation	0.1
Reproduction	0.1
Max. Depth	7

Table 4.4 Parameters for Initial Binary method

The results of the series of runs classifying one class against the other three are shown in Table 4.5.

Classification task	Successful runs ie max. possible hits	Range of depth	Range of nodes	Range of generation found
Liver vs rest	20/20	3-7	11-47	4-12
Kidney vs rest	19/20[1]	2-7[2]	7-31[3]	3-57
Hepatoma 7777 vs rest	20/20	2-7	7-45	4-59
Hepatoma 9618A vs rest	20/20	2-7	7-71	3-9

[1]the remaining run achieved 48/49 correct classification

[2]the run with 48 hits had depth 1

[32]the run with 48 hits had 3 nodes

Table 4.5 Results of binary classification, one class against all the others

In all runs the classification succeeded with relatively small parameters. There was no parsimony measure in the fitness function as it was not the intention to discover a 'best' solution but to see whether the data could be separated in this way. However, as solutions were found relatively early on in the run, there were some results which are relatively easy to interpret.

The results of an experiment classifying the two types of hepatomas are shown in Table 4.6.

Best result	23/23 (all runs produced this)
Generation at which found	Range 0 – 3
Number of nodes in tree	Range 3 – 19
Sample functions	H – L (found twice), K + V

Table 4.6 Binary Classification of Hepatomas

Results from the experiment to distinguish kidney tissue from liver tissue are shown in Table 4.7. The best result was the single node M which on its own distinguished between the two classes. Of the 20 runs this result was found nine times. Six of the remaining runs produced a result of M plus or minus a single other variable, two included M in a more complex function and none produced a solution not utilising M.

Best result	26/26 (all runs produced this)
Generation at which found	0 (all runs found a solution in the initial population)
Number of nodes in tree	Range 1 – 5
Sample function	M (found on 9 runs)

Table 4.7 Binary classification of Tissue

The experiment to distinguish tissue from tumour produced the results shown in Table 4.8. The best result was the single node D which on its own distinguished between the two classes. Of the 20 runs this result was found two times. Eleven of the remaining runs produced a result of D plus or minus a single other variable, four included D in a more complex function and only two produced results ((P - A) and (V – A – L - N) not using the variable D.

Best result	49/49 (all runs produced this)
Generation at which found	0 (all runs found a solution in the initial population)
Number of nodes in tree	Range 1 – 7
Sample successful function	D (found on 2 runs)

Table 4.8 Binary Classification of Tumour versus Tissue

4.4.2 One Tree Method

The results using this method are poor. A series of experiments with two different parameter sets were carried out to balance finding a solution with a reasonable execution time.

Table 4.9 shows the results for 20 runs within each series. The small parameters were a population size of 500 over a maximum of 150 generations, the large parameters were a population of 3000 over 250 generations

The fitness function did not have an element to encourage smaller solutions. Although experiments were run incorporating a parsimony measure these did not show any improvement; the solutions were smaller (but still large) and no extra hits were achieved. The addition of a parsimony measure is probably only useful in situations where solutions can be found. The large and successful functions could be simplified but this task takes away the automation required to use GP successfully to classify this data set. Changes to the use of the cos function to split the results into four classes were not examined.

Creating a function that would store results into memory during execution of a tree would allow multiple results to be returned from a single tree (Teller and Veloso 1995) but the complexities of extracting features from such a tree and the loss of pure function application meant that this route was not explored.

Series No.	Parameter Size	Successful runs (49 hits)	Range depth	Range nodes	Range gen. found	Range of hits (max = 49)
1	Pop. 500 Gen. 150 Max. depth 7	0/20	4 - 7	13 - 123	6 - 129	28-40 (mean = 33.4)
2	Pop. 500, Gen. 150 Max .depth 14	0/20	2 - 14	7 - 437	1 - 139	26-41 (mean = 35.3)
3	Pop. 3000 Gen. 250 Max. depth 14	0/20	10 - 14	73 - 389	37 - 237	32 – 46 (mean = 41.2)
4	Pop. 3000 Gen. 250 Max depth 20	0/20	18 - 20	163 - 513	26 - 246	39 – 46 (mean = 42.3)
5	Pop. 500, Gen. 150 Max. depth 25	0/20	2 - 25	7 - 655	1 - 147	26 – 44 (mean = 36.1)

Table 4.9 Classification Results for the One Tree Method

4.4.3 Four Tree Method

Five sets of runs were performed with this method as with the one-tree method described previously. The results are summarised in Table 4.10. This method worked, producing successful solutions with few nodes. It would appear that, although successful solutions were found with all parameters, size of population is more important than maximum allowable depth in this case. This is advantageous as extra depth could mean extra complexity whereas a larger population involves a greater part of the search space can be utilised.

Series No.	Parameter Size	Successful runs (49 hits)	Range depth	Range nodes	Range gen. found	Range of hits (max = 49)
1	Pop. 500 Gen. 150 Max. depth 7	5/20	4 - 7	22 - 126	37 - 149	40 - 49 (mean = 45.3)
2	Pop. 500 Gen. 150 Max. depth 14	6/20	4 - 14	32 - 242	40 - 150	26 - 41 (mean = 35.3)
3	Pop. 3000 Gen. 250 Max. depth 14	20/20	5 - 14	44 - 380	30 - 212	49 (mean = 49)
4	Pop. 3000 Gen. 250 Max. depth 20	20/20	5 - 20	48 - 356	33 - 146	49 (mean = 49)
5	Pop. 500 Gen. 150 Max. depth 25	7/20	5 - 24	30 - 278	34 - 148	43 – 49 (mean = 46.3)

Table 4.10 Classification Results for the Four Tree Method

Having found successful solutions, a parsimony measure was added to see whether smaller solutions could be found, similar to those found in the binary classification of this data. The fitness function was amended to have three elements

$$RFi = no\_hits + class\_bonus + parsimony\_bonus \qquad (5)$$

The run was also amended so that the GP run continued for the maximum number of generations rather than stopping once the maximum number of hits was achieved, to see whether smaller successful solutions could be found. A single set of twenty runs was carried out, with a small maximum depth and large population. The results are shown in Table 4.11

Set No.	Max. Depth	Population Size	Max. Generations	Mean No. of Hits	Fewest No. Nodes	Greatest No. Nodes
1	7	5000	250	48.8	30	48

Table 4.11 Results of the Four Tree Method with a Parsimony Term

The trees that distinguished liver from the rest were successful but showed a range of features. However, those for the other three classes showed some common solutions, shown in Table 4.12

Class	Example 1	Example 2
Liver	M- ((S – M) * D + I)	J – 1.0 – I - (U * N)[a]
Kidney	S – M – 1.0[b]	S – M - H – 1.0[a]
Hepatoma 7777	H – 1.0[b]	H – 1.0[a]
Hepatoma 9618A	S + I + 2L[c]	(H – 1.0) * D[a]

[a] The value 1.0 was obtained by simplifying J / J

[b] The value 1.0 was obtained by simplifying R / R

[c] The value 2L was obtained by simplifying L + L

Table 4.12 Common Solutions to the Four Tree Method with Parsimony Term

The features D and M, used in the binary classification of this data, also feature here. In the second example the classification used by Hepatoma 7777 is the same as that used by Hepatoma 9618A with one feature missing – a situation that has occurred many times.

4.4.4 Two Tree Method

The runs using this method were successful at finding solutions with relatively simple functions. The first set of experiments is detailed in Table 4.13 and produced such good solutions that no further sets were carried out.

Series No.	Parameter Size	Successful runs (49 hits)	Range depth	Range nodes	Range gen. found	Range of hits (max = 49)
1	Pop. 500 Gen. 150 Max. depth 7	17/20	1 - 7	4 - 70	9 - 125	44 - 49 (mean = 48.6)
2	Pop. 500 Gen. 150 Max. depth 14	16/20	2 - 14	4 - 152	9 - 125	44 - 49 (mean = 48.6)

Table 4.13 Results of the two tree method

Of the 20 runs in series 2, 18 identified the feature D in the first tree as that required to distinguish tissue from tumour, corresponding to earlier results. However the use of D on its own happens more often in this series than in the binary classification where D on its own is found twice, and with one other variable a further 11 times. The second tree, which has the effect of distinguishing between the groups (liver and hepatoma 9618A) and (kidney and hepatoma 7777) provided two small solutions

N – K and

D * (R - L)

Although the runs in series 1 produced similar results for the first tree, fewer of them produced small solutions for the second tree.

Series 3 – 5 were not run for this set of experiments because of the good results with the smaller parameters.

Evaluation of the varimax vectors involved in the results show that vector D is similar to the spectrum for lactate. Elevated lactate can be an artefact (i.e. during the time delay between cutting out the tissue and freezing it lactate may increase substantially) and is ambiguous from a biochemical point of view as there will be different levels of lactate in the original tissues and different levels of lactate based on excision times. This means that, although the classification is good in terms of the technique, it is less useful from a biochemical point of view (for excised tissues).

The original data was re-evaluated with the signals in the lactate region set to zero before principal component analysis. When the varimax spectra are compared with those of the dataset used in the previous experiment it was discovered that lactate was clearly present in only one vector, and others correspond very well to the vectors in the dataset with lactate. See for example vector J in the dataset with lactate, and K in that without lactate. The series of experiments were run again with the lactate-free dataset and the results summarised in Table 4.14.

Series No.	Parameter Size	Successful runs (49 hits)	Range depth	Range nodes	Range gen. found	Range of hits (max = 49)
1	Pop. 500 Gen. 150 Max. depth 7	19/20	3 – 7	8 - 86	10 - 143	48 - 49 (mean = 48.9)
2	Pop. 500 Gen. 150 Max. depth 14	20/20	2 – 14	8 - 240	10 - 131	49 (mean = 49)

Table 4.14 Results of the Two Tree Method used on Lactate-Free Data

In this set of experiments it appears to be slightly easier to find solutions. In both series, similar solutions are found. Table 4.15 shows solutions found on more than one run.

Tree 1	tumour versus tissue	N + H S – B S + H H - B
Tree 2	(kidney + hepatoma 7777) versus (liver and hepatoma 9618A)	J – 2L J – 2L + I

Table 4.15 Solutions from the Two Tree method and Lactate-Free data

4.5 Discussion

Both two-tree and four-tree GP methods have been able to classify the data into four classes. The results are not as satisfactory as those of the binary classification reported previously. The binary classification is a simpler task than that carried out here and it would be surprising if the binary classification performed worse than the multiclass classification. It would seem that in order to do such classification of the rat data more pre-processing of the data is required and more tuning of the GP process. The classification of this data set suffers from the problem (shared by many other medical data sets) of having very few examples to train on. It is hard to tell how robust the found solutions are and whether they are over fitted to the existing data.

The four-tree approach to classifying the rat data produced better results than those obtained by running a series of one-versus-the-rest binary classifications. This may have been to do with the extra size of the population in the former rather than the latter but it also allowed for exchange of sub-trees between trees which appears to have the effect of improving the overall fitness of the individual. This is not always the case – Muni (Muni *et al.* 2004) forces crossover to take place only between equivalent trees (see below).

Lisboa (Lisboa, P J G *et al.* 1993) used an eight class subset from the same database, to investigate methods for interpreting the features used by NNs for classification. The sensitivity parameters for two of the classes were presented and showed a relatively complex combination of the NMR variables.

The issues raised by multiclass classification differ in kind from binary discrimination. In the latter case, a boundary between the classes can be set to be a single numeric point, zero by convention. The meeting point of the classes is the only interesting boundary, values can be allowed to be as large as the system allows without endangering the classification process. There need be no problems of examples being assigned to no class or both classes. Often the discrimination required is between a class and everything else. In the medical area this could be

malignant or non-malignant abscess, diabetes or non-diabetes. In imaging it could be the difference between object and background. The second class in such task is non-malignant, non-diabetes or background.

Once a 'don't know' option is included, or more than one specific class is required the task becomes harder. Examples are distinguishing between different grades of glioma or between obverse and reverse faces of different value coins.

Some difficulties with multiclass classification with GP, as with other methods, are setting boundaries between classes, losing good partial solutions, conflict resolution where a sample is assigned to no class or to more than one class, the amount of pre- and post-processing involved, simplifying solutions to an understandable level and the computational cost involved. These will be discussed in turn.

The boundaries are set statically in the work presented here. The output range is [-1, +1] and the boundaries are set at -0.5, 0, +0.5. Any value output from a function application to a sample is applied to a standard cosine function to produce a result within range. As described in the results, the assignment of output range to class was random but fixed, different class orderings had little impact on results obtained.

Zhang and Smart describe two methods of dynamically determining class boundaries (Zhang and Smart 2004) which can be used at intervals during a run to reset the class boundaries. The first is Centered Dynamic Class Boundary which involves examining the range of output values obtained for each class and finding the centre point. The boundaries are then set as the middle point of two adjacent centres. Their second method divides the expected output range into a large number (200) of slots. At the time the Slotted Dynamic Class Boundary is being calculated, each slot the is assigned to the class with the most results in that slot. Both these methods are used periodically with static boundaries at first used, then one of the dynamic boundary techniques used every five generations. The method is applied to an image classification problem and both improve the performance of GP over the standard method. However the computational overhead is increased.

The problem of losing partial good solutions is particularly evident in multi-tree approaches where one tree may be poor in a solution with a generally high fitness or may be good in a solution with generally low fitness. Experiments reported in this thesis involved adding terms to the fitness function to give weight to both successful partial solutions and successful complete solutions. However the crafting of a fitness function that managed this in such a way as to encourage improvements took a considerable amount of time and effort for small gains in performance and further work on this was halted. An interesting and successful technique has been reported in (Muni *et al.* 2004) using a multitree classifier where the number of trees equals the number of classes. They introduce the idea of 'unfitnesss' as well as fitness to try to minimise the disruption to good trees (sub-solutions with high fitness) whilst maximising the opportunity for poor trees (sub-solutions with low fitness) to improve. The selection of individuals for the next generation is carried out in two parts. Firstly, the whole individual, containing all trees, is given a fitness value as usual by applying it to the test set and recording the results. Those individuals with higher fitness are then examined in order to give each of the constituent trees an unfitness value – how well each performed against their allotted task. The trees with high unfitness in individuals with high fitness are then more likely to be selected to take part in genetic operations of crossover and mutation. Crossover is adapted so the crossover point on each parent is chosen and the subtrees below that point are swapped, but so also are the remaining whole trees of the individual. The crossover of subtrees can only happen between trees classifying the same class. (In the experiments on the MRS data presented here there seemed to be little difference between restricting crossover to subtrees classifying the same class and allowing crossover to occur anywhere in the individual, in contrast to Muni's work.) Mutation is likewise applied probabilistically more often to unfit rather than fit trees of an individual.

This approach seems to show the way to solving the problem of variation between the fitness of trees of an individual. It allows a way of improving specific parts of an individual whilst shielding the successful parts from the destructive effect of mutation and crossover. Smart and Zhang (Smart and Zhang 2005) use a method of storing the best solution for each task as 'experts' through the run to achieve the same effect of capturing good partial solutions as they appear during the course of

the run. Although they also use multiple trees to solve a multiple class problem, they, unlike Muni and the work described here, do not specify which task each tree should undertake. Rather, they allow each tree to find a solution to a two-class problem. Fitness calculations involve ranking each individual as to their performance on each binary classification task, and then applying their fitness based on their best ranking. This ranking rather than absolute fitness allows for solutions to be found for all sub-problems.

In the work presented in this chapter no conflict resolution was required or used. The number of classes was known in advance and no samples could be from more than one class. Therefore, functions were counted as better when they correctly identified all samples into the correct class. Experiments were run using methods such as those described above from Smart and Zhang where more trees than classes were provided and trees competed with each other in an individual as well as between individuals to produce the best classification scheme. The results of these experiments were generally not as good as other methods and have not been reported here in detail. The use of the two tree method partitions the data in two ways with the final solution relying on the results of both trees. This example illustrates the fact that one partitioning makes sense – tumour versus tissue, whereas the other has no obvious logic – a tumour and a tissue, versus a different tumour and a different tissue. The results show that this method can produce results that are meaningful in a one-versus-the-rest way – kidney can be isolated for instance.

The issues of conflict resolution and post-processing of data are two parts of the same problem. If there were no classification errors there would be little need to adjust the functions evolved. In some data sets and GP setups conflict resolution is important. Kishore (Kishore *et al.* 2001) uses both weight and heuristic rules to fit misclassified samples into classes. Muni makes use of the population of solutions by taking the most fit individual and using a logical OR function on the trees of that individual with each tree of every other individual to produce a better solution. This is interesting as it again gives the opportunity for good partial solutions to be found and utilised.

Preprocessing of data can involve reducing the size of input vectors as with PCA, or normalising data into a specific range for ease of processing or uniformity of sample ranges. Other pre-processing often involves a hand-crafted partial solution to the classification problem to make it easier to solve. This may be necessary with intractable problems but does remove the advantages of using an AI or machine learning technique to solve the problem. Kishore (Kishore *et al.* 2000; 2001) uses a technique called Feature Space Partitioning. He compares the effect of a series of binary classification tasks, which use the whole of the search space to find a solution with Feature Space Partitioning which simplifies the task of distinguishing between multiple classes by partitioning the space of all possible solutions into smaller spaces that contain examples from fewer classes. The advantage of this method is that the task is made simpler. The disadvantage of this is that it involves both preprocessing of the data into feature spaces and post-processing of the solution by applying conflict resolution tools to allow for the cases where a solution attaches more than one class label or no class label to a sample. It appears to be a sophisticated method of partially solving the problem before presenting it to a GP run. Smart and Young in the work discussed above use Communal Binary Composition to ensure multiple class classification can take place in one run. They allow the trees in an individual to perform on any of the binary classification sub-problems of the whole task. These binary tasks are not of the form one-versus-the-rest but a specific class versus another specific class. The fitness of an individual is based on their highest ranking in all possible binary tasks. The reported results show the technique is useful in an image data task. The preprocessing allows the data to be presented in a way to facilitate such binary tasks.

Other work (Silva and Tseng 2005) decomposes a multiclass problem into a series of binary tasks and then aggregates the resulting functions in order to classifiy between several groups.

Simpler solutions to classification and other tasks are not only easier to understand but tend to generalise better, as they reduce the chance of overfitting data to the training set. There are advantages to using the GP process to reward smaller individuals with the same fitness over larger ones. Parsimony measures are usually added to the fitness function to achieve this. In the work presented here, the

parsimony measure added to the complexity of fitness function and was difficult to scale effectively – too much weight on the parsimony term and GP tended to small but bad solutions; too little and the run was ineffective. There has been much work in the GP field on reducing 'bloat' and removing 'introns' – both descriptions of parts of a GP tree which add nothing to its functionality but contribute to its size. None of the recent papers dealing with multiclass classification have raised this as an effective technique for simplifying functions. None use Automatically Defined Functions (ADFs) either.

The problem with multiclass tasks is the computation overhead they involve. It has been reported by (Zhang *et al.* 2003) that the computational effort required to do a single multiclass run is higher than that required to do the equivalent task as a series of binary classifications. (Mcintyre and Heywood 2005) use a co-evolutionary approach to reduce the computational overhead in situations with a large set of data. They evolve both a classifier and a trainer which is a subset of data which compete to maximise solution fitness. They also implement niche-enabling techniques often found in GA to aid the evolution of better solutions.

It is interesting to note that much of the work on multiclass classification utilises a high proportion of mutation as well as crossover in order to drive towards a successful solution. This is in contrast to canonical GP which holds that crossover is the main genetic operator and mutation is to be used sparingly. It does allow for small adjustments to be made to a function without losing the main structure. Konstam, in his work on classifying Egyptian skulls (Konstam 1998), uses GP to find the structure of a solution and then uses GA to fine tune the solution by mutating the constants.

4.6 Conclusion

A conclusion could be drawn that the extra complexity in the results of multi-class classification means that although the technique can work it will not replace binary classification as a technique of choice. It would be feasible to use the multi-class

technique first, then to use a binary classification method to classify samples from those classes that have proved difficult.

Once the results are complex it becomes difficult to see what part individual features are playing in the classification. Without the advantage of this feature extraction, the use of GP over NN is not proven. The ability of NN to classify is well-known; it is the fact that the features and weights given to those features are difficult to extract from NN solutions that leads to exploration of other methods. If the results of these are as difficult to interpret as those from NN then there is no advantage to using GP.

The use of GP to classify multiple classes has not been proven here to become a method of choice. The results do show that there is scope for further work on this technique as the benefits would be so useful. Moreover, the inspection of the more successful solutions found in a series of runs can give information about which metabolites could be important in such a classification. For example, in the two tree experiment with the lactate-free dataset, vectors H, N and S feature regularly with a positive loading whereas on the frequent occasions where vector B is involved, it always take a negative form. The successful solutions have also exploited the arithmetic functions to provide solutions with the value 1.0, either by the function x/x or x/(y-y) where the protected division returns 1.0 when given a division by zero. Unlike the *in vivo* data, it is difficult to make a clear interpretation of these functions for feature extraction purposes. However, the combined effect of some solutions could highlight common metabolites.

Chapter 5

Genetic Optimisation of NMR Pulse Shapes

5.1 Introduction

Nuclear Magnetic Resonance techniques have a wide range of applications from the identification of chemical compounds (with NMR spectroscopy) to the diagnosis of cancer (with MR imaging). Improvements in techniques to acquire NMR data can lead to simplifications in the analysis of that data. Apart from the choice of sample and details of hardware, the main feature determining the information obtained from the NMR measurement is the choice of pulse sequence. In general, an NMR pulse sequence comprises a series of radiofrequency pulses, applied magnetic field gradients, delays and data acquisition events. However, a simple case of NMR spectroscopy requires only a single RF pulse followed by a period of data acquisition. Considerable effort and expertise has been used in designing and implementing NMR pulse sequences for specific purposes. The present study investigates the possibility of using genetic techniques to evolve one or more NMR pulse sequence elements and, ultimately, entire pulse sequences. If this is possible, it will allow collection of data to be tailored to allow for more straightforward analysis.

GP methods have previously been used with data from NMR systems including classification and feature selection of tumour data collected by NMR spectroscopy (Gray *et al.* 1998) and image enhancement (Poli and Cagnoni 1997). GA has been used in spectroscopic infrared imaging (Van Den Broek *et al.* 1997) and to generate pulses for NMR spectroscopy in a study which used a subjective fitness function, interactively judging fitness (Freeman and Wu 1987). In addition, evolutionary methods have shown to be useful in the shaping of laser pulses (Assion *et al.* 1998).

Evolutionary computation has been used with hardware, although much of the work has focussed on evolving hardware such as field programmable gate arrays (FPGA) (Thompson 1996) (De Vega *et al.* 2004) and logic circuits (Ali *et al.* 2004) or evolving sensors for use in robotics (Balakrishnan and Honavar 1996). The use of evolutionary computation methods rather than hand-crafted or methods using mathematical models to enhance or optimise signals have been explored in NMR in quantum computing (Rethinam *et al.* 2004), optimisation of NMR coils (Andris and Frollo 2002) and in radiotherapy treatment planning (Haas 1999).

NMR spectroscopy involves one or more radiofrequency pulses to detect the presence and quantity of different substances in a sample or region of interest. The output is displayed as a spectrum with different substances giving rise to peaks at different frequencies. It is often the case that some peaks (such as water) may dominate the spectrum and it may be desirable to suppress these or enhance others. This is especially a problem for *in vivo* NMR spectroscopy where the water concentration (about 40 M) is substantially greater than that of all biochemical compounds of interest. Where there is a very large peak in a spectrum, it can be difficult to use the information in the rest of the spectrum as it is so relatively flat. Elevated lactic acid concentrations are a common feature of several diseases (such as in stroke and solid tumours) but even in such cases a concentration of 1-10 mM is typical, i.e. about 10000-fold lower than that of water. The selective suppression or excitation of NMR signals can be achieved using shaped pulses with frequency selective properties. The pulse shape is described by a sequence of values of phase, amplitude and duration. Figure 5.1 shows the effect of suppressing the water signal in a spectrum.

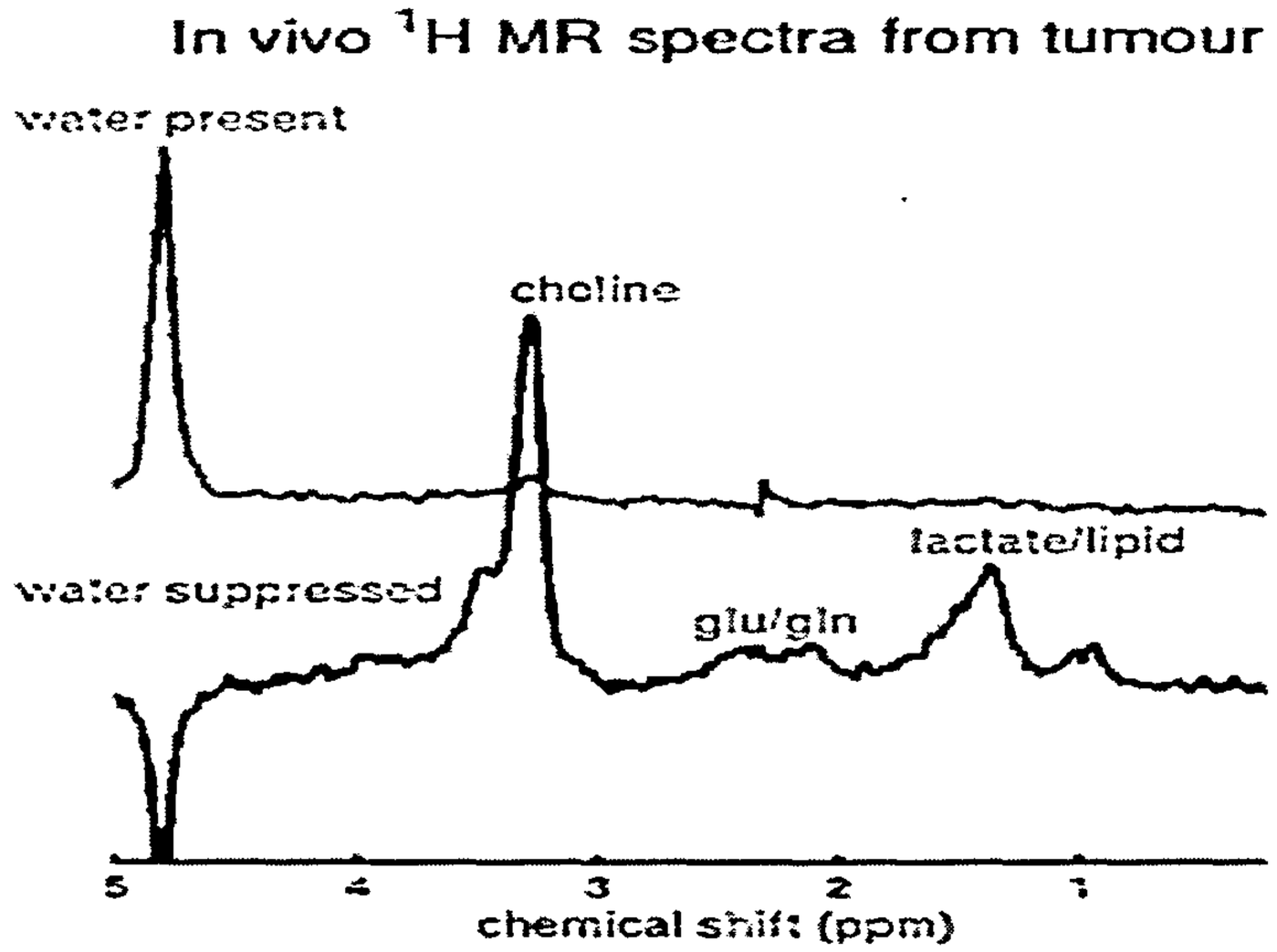


Figure 5.1 The effect of suppressing water on a spectrum

The design of such pulse shapes can be done on a theoretical basis (Geen and Freeman 1991). However, for many purposes, only approximate solutions are obtained. In addition, hardware errors or deviations are not taken into account. It may be the case that new pulse shapes are variations on existing successful ones and that it may be difficult to find novel sequences. Evolutionary techniques, such as GP, offer a way of developing pulse shapes which explore a wider range of possible solutions.

There are advantages to running candidate solutions directly on an NMR system rather than in simulation. Hardware features can be incorporated into the fitness measure without needing to be explicitly programmed. This means that novel solutions that overcome hardware limitations may be discovered.

The work detailed in this paper uses both GP and GA to generate pulse sequence elements and automatically evaluate them on an NMR system without human intervention. GP was initially used where the generation of pulse files occurred as a side-effect of executing the function generated. The structure of a GA chromosome as a vector fitted more closely with the structure of pulse files required in this application and so the experiments were also run with GA.

5.2 Methods

5.2.1 NMR Setup

The NMR system involves a 4.7 Tesla magnet and a Varian NMR spectroscopy/imaging system running VNMR (version 6.1) software (Varian, Palo Alto, CA, USA). The sample used in the experiments is 255 mM trimethylsilyl-2,2,3,3-tetradeutero-propionate sodium (TSP) dissolved in a 1:2 mixture of water and dimethylsulphoxide (DMSO). This was contained in a 30µl spherical bulb, placed in the centre of a 13mm diameter two-turn coil tuned to the ^{1}H NMR resonance frequency (200 MHz). This sample gives three signals in the ^{1}H NMR spectrum: the DMSO and TSP signals are 370 Hz and 900 Hz upfield, respectively, of the water signal.

Matlab (The Mathworks, Natick, MA, USA) is used for receiving and copying the files and for displaying results.

5.2.2 GP Parameters

The GP system used is lil-gp (Zongker and Punch). The functions available to the program are addition, subtraction, multiplication, protected division (returning 1.0 to a division by zero) and `writepf`. This constrains its two arguments to the ranges –360.0 to +360.0 and 0 to 1023.0 and writes them to a text file as phase and amplitude values. The duration value is set to be 1.0 for each pulse sequence element in all experiments. `writepf` also returns the value of its first argument. The terminal set consists of random real numbers in the range –1.0 to +1.0. The fitness function is as follows;

$$\text{Standardised fitness} = 1-((\text{nmr\_fit} +(3* \text{no\_lines}))/10000) \quad (1)$$

nmr_fit is the value returned by the NMR system and is a ratio between two of the peaks, one to be suppressed, the other detected. The value of 10000 is an arbitrary one (and thus standardised fitness is not used as a test of success). no_lines is the number of lines in the pulse file generated by the individual. This term is included to encourage the development of longer pulse files. The factor of three was included so

that the term is relatively significant in early generations where raw fitness values are small and becomes relatively less significant as raw fitness values increase.

At each generation of the GP run each individual is evaluated, producing a pulse shape in text format (referred to as a pulse file). The pulse file is made available to the NMR system which then activates the pulse-acquire sequence with a pulse width of 200-2000 μs (fixed for any given GP run). The offset frequency of the pulse is set to be the same as the water resonance frequency. The resulting spectrum is received by the NMR system. Calculations of peak heights and ratios take place there and the resulting value (nmr_fit) is returned to the GP program to allow for fitness calculation. The evaluation of each new pulse occurs every five to eight seconds. A diagram of the system is shown in Figure 5.2.

The aim of the first experiment was to maximise the ratio between DMSO and water. The second was to maximise the ratio of TSP to water. In both of these cases the task was therefore to suppress the signal (water) at the pulse carrier frequency compared to off-resonance signals (DMSO or TSP). The third experiment was to maximise the ratio of TSP to (DMSO + water).

All experiments were run with a population of 50 over 50 generations. The replication and mutation rates were each set to 0.1 and the crossover to 0.8. Initial depth of trees was between two and six with a maximum depth of 20. Each experiment was run three times with different random number seeds.

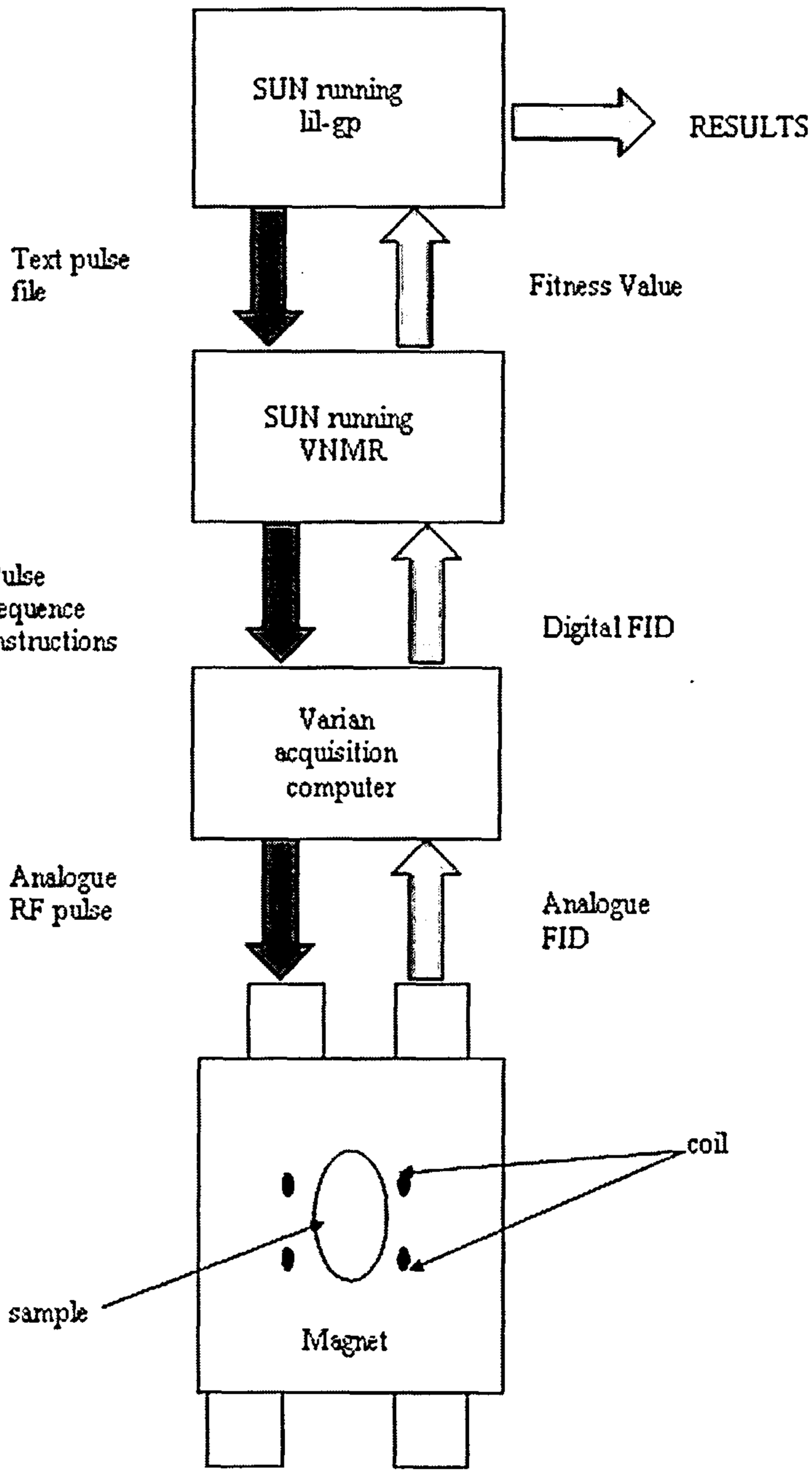


Figure 5.2 The setup for the pulse experiments.

Results of the second experiment were compared to several known pulse shapes and to sets of 2500 random pulse shapes (i.e. the same number of pulse shapes were tested in the random runs as in the GP runs).

5.2.3 GA Parameters

The GA system used was the Genetic Algorithms Optimisation Toolbox for use with Matlab (Houck *et al.* 1995). A Matlab implementation was chosen because some aspects of the system for running pulse file generations with GP had already been written in that language, it allowed for real-valued chromosomes and also because the graphical reporting tools allow for easy comparison of results.

Matlab is an interpreted language which means it could run slower than the compiled version of lil-gp (written in C). However, the matrix data types built in to Matlab allow the code to be simpler and shorter than with other languages and this could help with speed. Moreover, the major time component of the experiment is that required by the NMR system, with built-in pauses between evaluation to allow a return to equilibrium of the nuclear magnetism (spin states), so the extra overhead of interpretation is not a significant feature. Also, the GA chromosome produced is a (2xn, 1) vector which needs to be reshaped into a (n,2) file whereas the GP individual produced is a function which has to be executed for the values to be placed into a file. The GA processing for this stage is therefore both simpler and faster than that of GP.

The GAOT system was set up to provide a chromosome containing n by 2 elements of real values, n being the length of the file, the two values per n are for phase and amplitude. As with the GP experiments duration was assumed to be 1 for each element and this value was added to the file formed by the chromosome just before evaluation. The values of n were set to be 5, 10 and 20. Most experiments took place with n = 10.

The values allowed in the phase and amplitude elements were constrained to [-360, 360], [0,1024]. Matlab has an inbuilt function *bounds* which allows for this.

GAOT is supplied with a choice of mutation, crossover and selection functions. The standard experimental set up for this application used one mutation function and one crossover function. *SimpleMutation* allows a single element of the chromosome to be replaced by a randomly generated value. *SimpleCrossover* is a single point crossover applied at the same point to both parents with the values after the cut point

swapped between the parents.

The selection functions available included a normalised geometric selection, based on ranking individuals on their fitnesses and assigning probabilities of selection based on that fitness in relation to the fitness of the rest of the individuals in that generation. Tournament and roulette selection functions are also included. In most runs tournament selection with a tournament size of 5 was used. This followed the use of the selection functions in the GP experiments. The tournament size was tested with values between 3 and 9 and did not seem to be a significant factor in the success or otherwise of runs in this experiment.

GAOT was set up to allow evaluation of all individuals in each generation in batch mode. This had several advantages. It minimised the delay in transferring files from the evolutionary system to the NMR controlling program and the transfer of fitness values back again. As the VNMR system can be configured to run a set number of pulses in sequence, there are fewer problems associated with repeatedly starting and stopping the VNMR program. The parameters of the GA experiments are summarised in Table 5.1

The fitness function for the experiments was the same as with the GP experiments

Fitness = (height of TSP peak / height of water peak + height of DMSO peak)) * 1000 (2)

The GAOT program was initially configured with an elitism strategy (Michalewicz 1996). This ensures that the best individual found so far cannot be lost from the population by always replacing the individual with the worst fitness with the one with the best fitness, once all evaluations had taken place. The elitist strategy could have the effect of reducing diversity in the population and in this application the small population size allied with a relatively large tournament size ensured the best solution has a reasonable chance of being selected, so the elitist strategy was not used. Each set of parameters was run five times with different seeds.

Individual size	n * 2 (where n = 5,10 or 20)
Population size	20
No. of generations	50
Mutation function	simple mutation (one element change)
Mutation rate	4-200 mutations per generation
Crossover function	simple (1 point) crossover
Crossover rate	4-200 crossovers per generation
Selection function	Tournament or NormGeomSelect
Size of tournament (or Selection pressure)	3 – 9 (0.08 -0.48)
Fitness function	(height TSP peak/(height of water peak + height of DMSO peak)) *1000

Table 5.1 GA parameters

5.3 Results

5.3.1 GP Results

The results from the first set of experiments showed an increase in mean fitness of the population as well as an increase in the fitness of the best. The best individuals had fitness values more than 100-fold better than the mean fitness value seen in the initial (random) generation. Figure 5.3a shows the increase in mean fitness in one run.

The spectra produced from experiments can be compared with that produced by a square pulse. The square pulse shape is produced by values [0.0, 1023.0, 1.0] describing phase, amplitude and relative duration, respectively. The resulting spectrum is shown in Figure 5.3b.

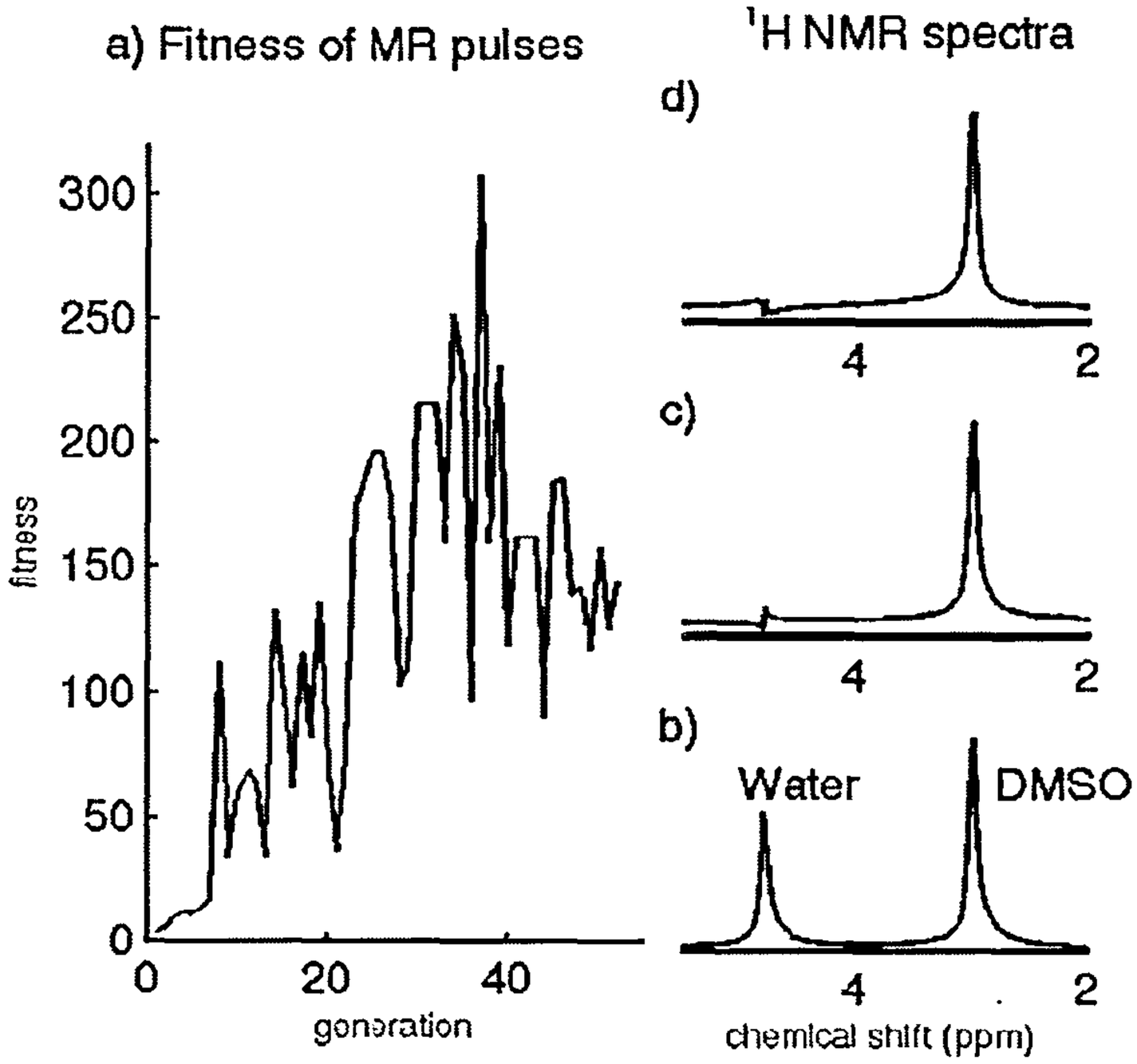


Figure 5.3 Results of the experiment to maximise the ratio between DMSO and water. a) mean fitness changes during GP run ; ^{1}H NMR spectra with b) square pulse c) 1-$\bar{1}$ like pulse from GP run and d) best pulse from GP run.

Running experiments on the NMR system directly has the effect that identical runs will not always produce identical results, although the differences are small. For this reason all the results in this paper are shown with the baseline fitness (i.e. the fitness score for a one-element square pulse shape) set to one. This calculation was performed at the end of GP runs and allows a more meaningful comparison between runs and between experiments. An individual in generation five produced the pulse file [179.94, 100.0, 1.0; 0.0, 95.39, 1.0]. The spectrum from this is shown in Figure 5.3c. This is very close to simple 1-$\bar{1}$ pulses (i.e. pulses with two elements of equal amplitude and 180^0 phase difference) which are well known for their solvent suppression properties. The 1-$\bar{1}$ pulse is the simplest of the family of binomial selective excitation pulses and pulses approximating another binomial pulse (1-3-3-1) have also been produced in GP runs. The 1-3-3-1 binomial pulse has four elements with the second and third elements having three times the amplitude of the first and fourth elements during the pulse duration. The first and third elements have the same phase as each other, the second and fourth are 180^0 out of phase to that.

Other binomial pulses follow the same pattern. Binomial pulses with more elements have a smoother effect than simpler ones.

Later generations contained individuals producing higher fitness. The best in one run, from generation 26, had a fitness of two orders of magnitude higher than the base line value. The spectrum produced from this individual is shown in Figure 3d.

The second set of experiments (maximising the ratio of TSP to water) was also successful, again showing improvements in fitness in the region of two orders of magnitude. Table 5.2 compares the results from the GP runs to pulses obtained by other methods. The pulses used were a non-selective square pulse, three binomial pulses and a BURP pulse. The BURP pulse was calculated using 'Pandora's Box' software included in the Varian NMR programme based on the method described in (Geen and Freeman 1991). The pulse was defined as having a bandwidth of 2250Hz (corresponding to 2000 μs pulse width) and the off-resonance excitation optimised (to –1900Hz) to give the maximum fitness value. Figure 5.4 shows the spectra from these pulses and Figure 5.5 shows the increase in maximum fitness of individuals through the generations.

Pulse shape	Fitness
a) Square (24 μs)	1
b) 1-$\bar{1}$ binomial (2000 μs)	3
c) 1-3-3-1 binomial (2000 μs)	41
d) 1-5-10-10-5-1 binomial (2000 μs)	335
e) BURP (2000 μs)	19
f) GP (2000 μs)	91
Best random pulses (2000 μs)	5

Table 5.2 Comparison of GP with other pulse shapes for suppressing the water peak in relation to TSP (a-f correspond to the NMR spectra in Figure 5.4)

It can be seen that the best pulse obtained by GP performed better than the 1-$\bar{1}$ and 1-3-3-1 binomial pulses and the BURP pulse. The 1-5-10-10-5-1 gave the best

performance although from Figure 2 it can be seen that water suppression is virtually as effective with the GP pulse (f) as with the 1-5-10-10-5-1 (d). The latter pulse is also very effective in suppressing the DMSO signal but this factor was not used for the fitness calculation in these experiments. The BURP pulse performed relatively poorly. The comparison is not entirely fair as one of its features (avoiding phase distortions) is not evaluated here because signal phase is not taken into account (magnitude spectra are used). The best pulses found by random search of the same number of pulses tested in the GP runs showed only a 5-fold improvement in fitness compared to non-selective pulses. A comparison of fitness between random and evolutionary search is shown in Figure 5.6.

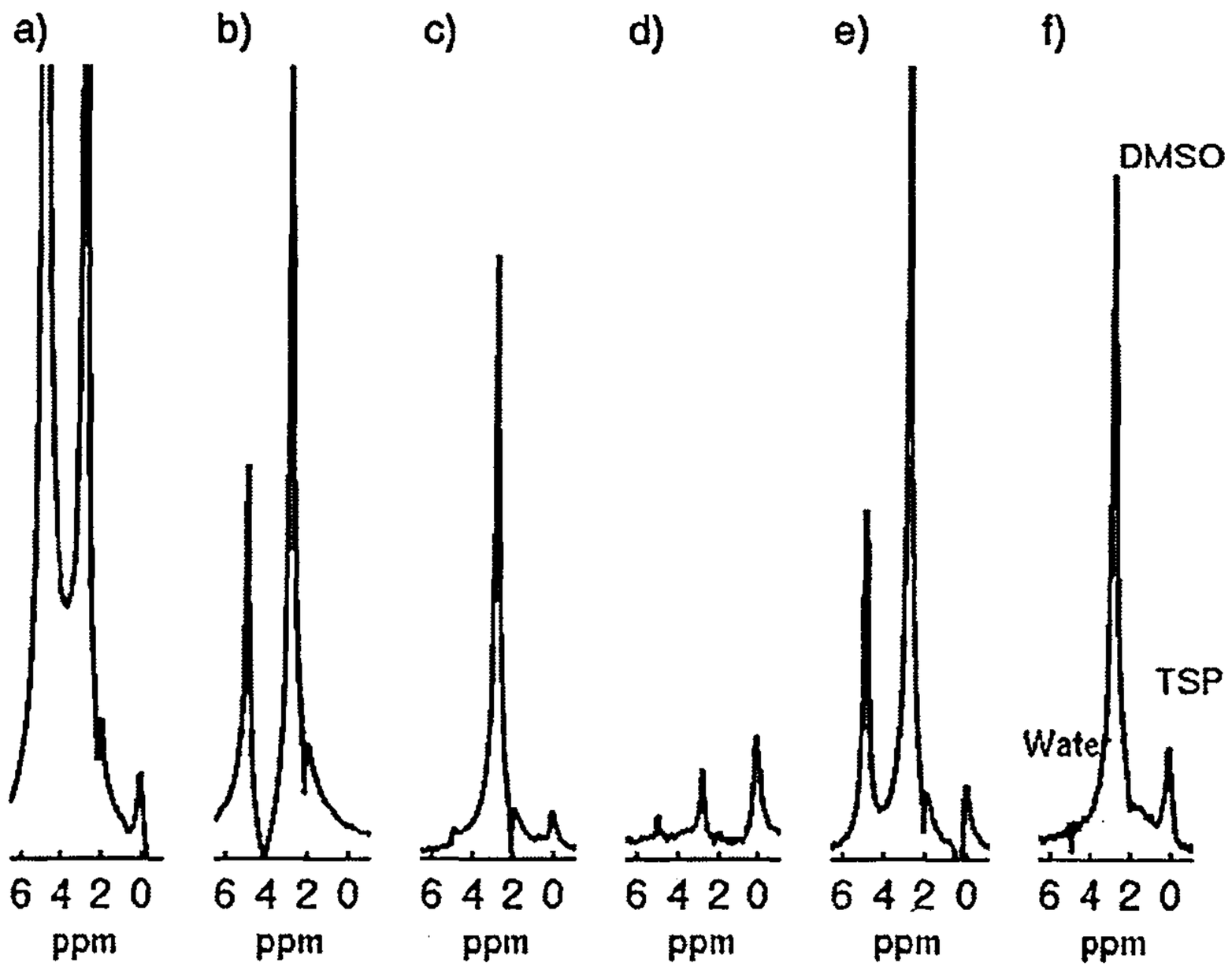


Figure. 5.4 Improvements in ratio between TSP and water. a) square pulse, b) 1-ī binomial, c) 1-3-3-1 binomial, d)1-5-10-10-5-1 binomial, e) BURP, f) GP

The third experiment is the hardest and improvements over 50 generations are smaller. Improvements of fitness in the order of five times as high as baseline have been found. These pulses tend to be relatively good at suppressing the water peak and relatively poor at suppressing the DMSO peak.

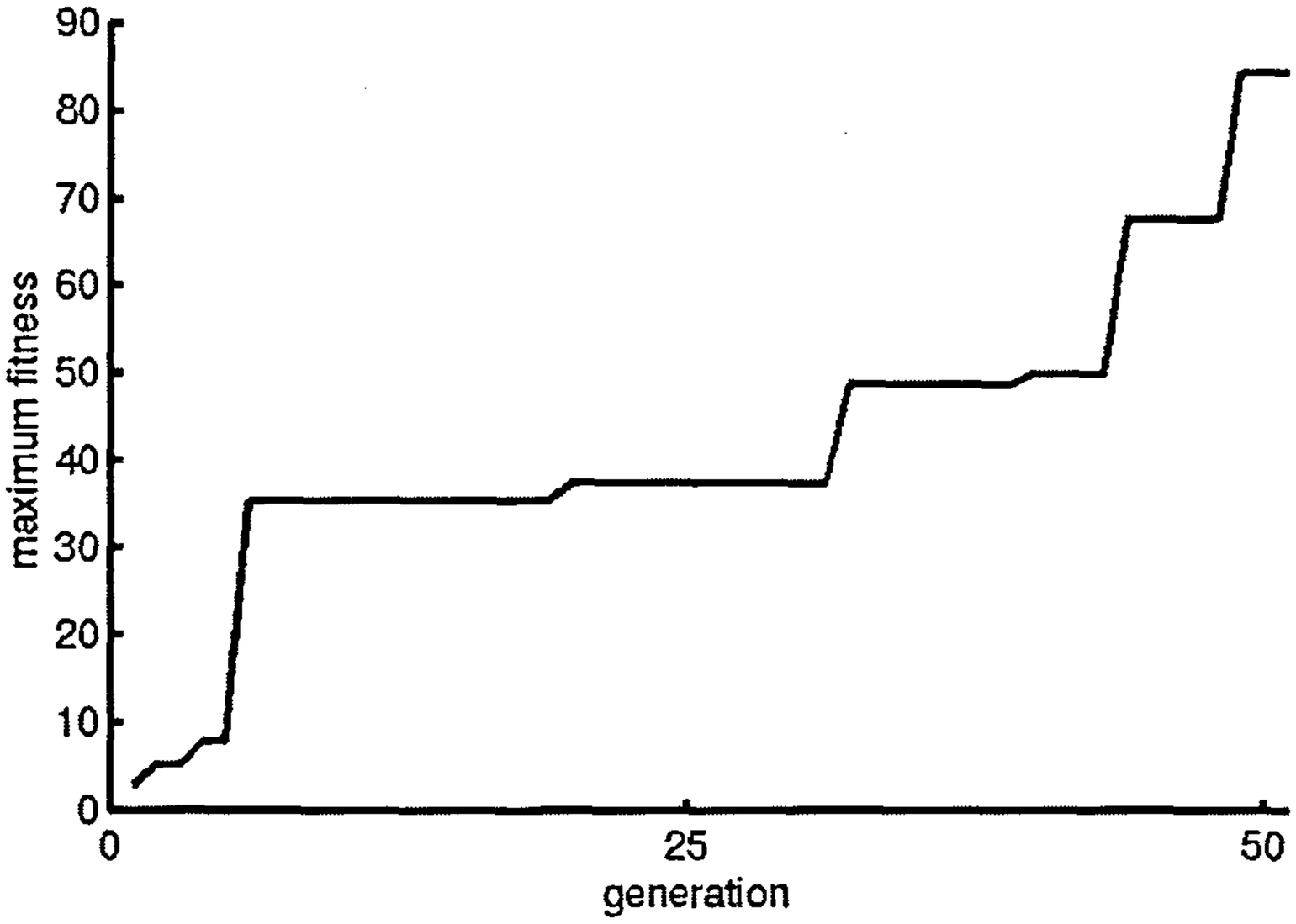


Figure 5.5 Improvement in Fitness during a GP Run maximising the Ratio of TSP to Water

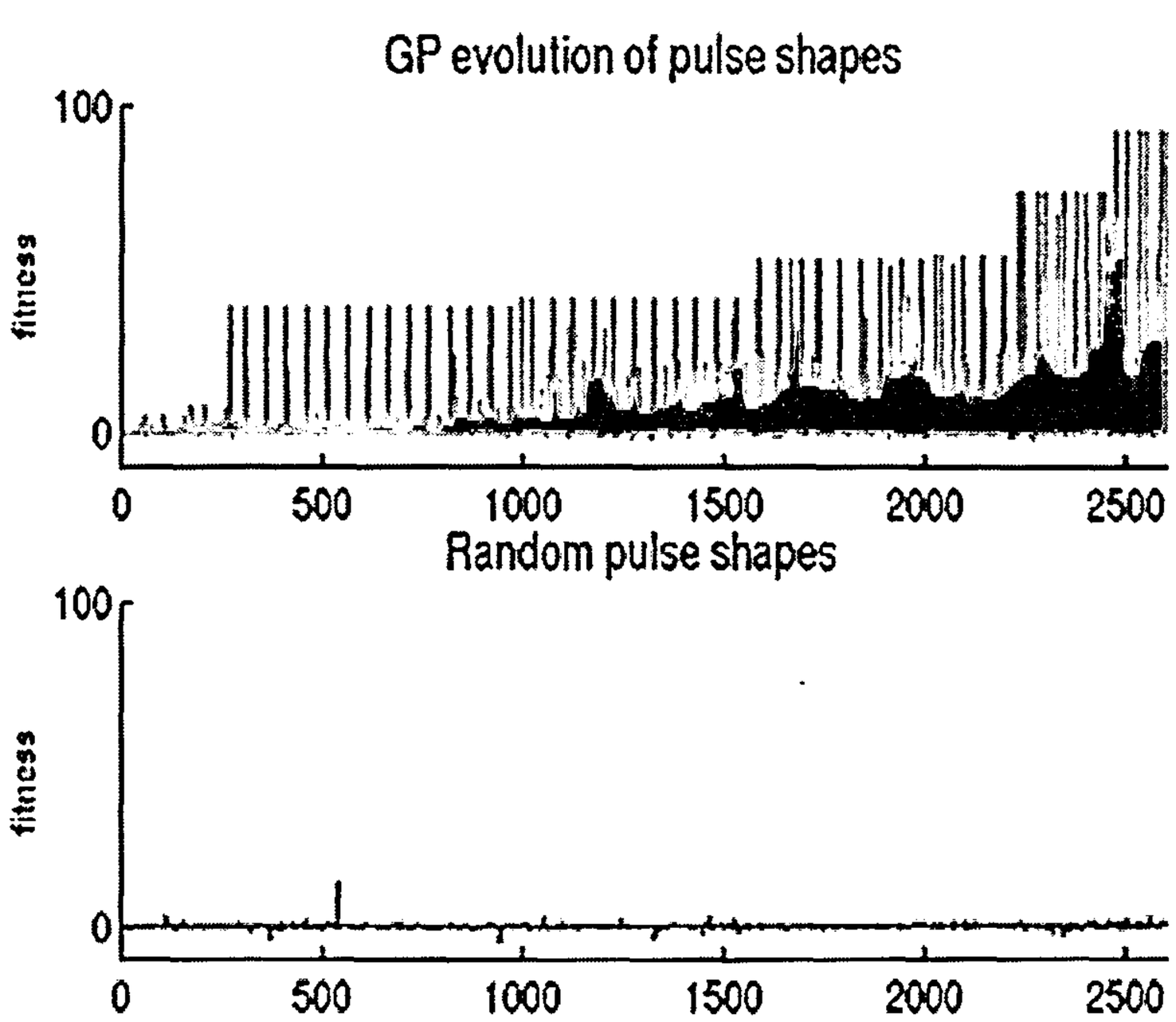


Figure 5.6 Improvement in maximum fitness in each generation for pulses optimising TSP signal compared to water signal.

5.3.2 GA Results

Results from the use of GA were similar to those achieved by the use of GP. However, the running of the experiment proved to be much simpler. GA produced good solutions in nearly all runs. The program appears to be robust with respect to parameters used, with only extreme values (such as very high mutation or crossover rates) forcing the program to fail to produce a good result.

The run which produced the individual with the highest fitness was run with the parameters shown in Table 5.3. The resulting pulse file is shown in Figure 5.7.

Individual size	20 i.e. 10 rows of phase and amplitude values
Population size	20
No. of generations	50
Mutation function	simple mutation (one element change)
Mutation rate	4 mutations per generation
Crossover function	simple (1 point) crossover
Crossover rate	10 crossovers per generation
Selection function	NormGeomSelect
Selection pressure	0.18
Fitness function	(height TSP peak/(height of water peak + height of DMSO peak)) *1000
Fitness	1787.4

Table 5.3 Parameters used in the GA run producing the individual with the highest fitness

The average fitness of the first (random) generation of this run was 1.41 with a maximum of 2.8, so the improvement over the first generation was more than 600-fold.

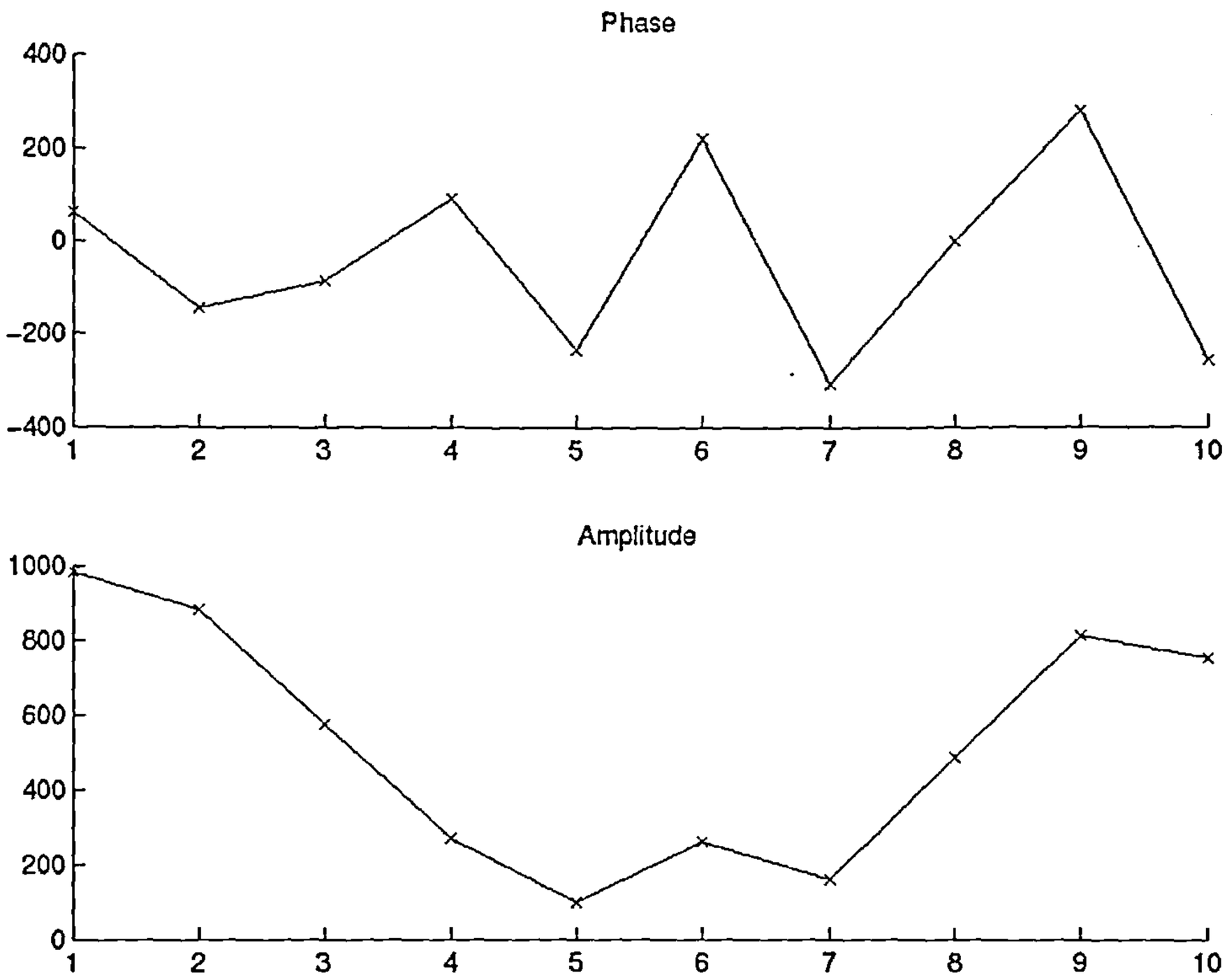


Figure 5.7 Phase and Amplitude plots of the Fittest Individual from a GA for the Experiment to Maximise the Ratio of TSP to DMSO and Water

5.4 Discussion

The discovery by GP of 1–ī pulses in the first experiment was interesting as, although not novel, they were discovered with little prior knowledge built into the system (The only assumptions made are in defining the GP fitness function and function set which are expected to be sufficient to find a solution). Later generations produced more complex pulse shapes with much higher fitness. The emergence of novel solutions argues the case for this type of method.

An advantage of techniques such as GP that work on a population of candidate solutions is that multiple solutions can be found. Although analysis of successful pulse shapes and the GP trees that produce them has not been undertaken in this work, it may be, as with (Poli and Cagnoni 1997), that there are underlying features

to these that can be exploited by further experimentation or other optimisation methods.

The advantage to running experiments on hardware is that the results are optimised for the hardware. Features and limitations of the hardware are taken into account, either explicitly, as when setting up limits to values in pulse files, or implicitly as e.g. effects of non-standard lineshapes and eddy-currents should be compensated for. The main disadvantage to running directly on hardware is time. A typical experiment with 2500 individuals takes between six and eight hours to run. Taking human intervention out of the experiment has meant that it has been possible to have larger populations and a greater number of generations than previous studies e.g. (Freeman and Wu 1987; Poli and Cagnoni 1997). However, it is still a heavy time burden. There are possible ways of speeding the process up, one of which is to operate the acquisition stage in batch mode, where all pulse shapes produced by a generation could be run in one step.

The success of the method of evolving pulse shapes allows for the possibility of co-evolving more than one pulse sequence element with the final aim to evolve a complete pulse sequence. This could then be used to aid data analysis.

5.5 Conclusion

GP and GA have been used to automatically generate NMR pulses and evaluate them directly on an NMR spectroscopy/imaging system. The GP approach has automatically evolved solvent suppression pulses. Both simple and novel pulses have been obtained with virtually no prior knowledge (using only the restrictions described in the methods).

Although it is not proposed that genetic methods are necessarily the best way of finding solvent suppression pulses, the approach does offer a number of advantages. The existence of a population of candidate solutions allows for the discovery of novel solutions and may prove to be useful in discovering common underlying features in the more successful pulses.

A second potential advantage of this approach is that RF pulses are evaluated on the NMR system itself such that hardware limitations or features are intrinsically taken into account. An additional advantage of the automatic evaluation of pulses without human intervention (compared to previous implementations of evolutionary computation (Freeman and Wu 1987; Poli and Cagnoni 1997) is that a much larger number of pulse shapes can be tested.

Although both GP and GA produced good results, the advantage of using GA is the match between the GA representation of an individual as a vector and the structure of the pulse file required for the NMR system. GA crossover and mutation operate directly on data values in the vector whereas the same operations in GP work on function applications which will change the outcomes. It does seem that the match between GA and the required pulse files make it a preferable method. If the main difference between GA and GP is the representation of the solution, then choosing one method over another can be based on representation method.

The experiments run in GA were set up so that they could be run in batch mode. In order for the VNMR system to do this it needed to be running the same number of pulses in each generation. This meant that individuals would be evaluated each generation, even if there was no change from the previous generation. Since the evaluation is a real NMR experiment, the fitness values for identical individuals are not necessarily the same. The amount of variability can be particularly high for successful pulses i.e. those where there is good suppression of water and DMSO as the denominator values in the fitness function can be very small. The improvement in fitness is genuine but the diversity of the population can appear greater than it really is. The convergence has not affected the success of the runs in these experiments.

Chapter 6

Optimisation of the Collection of NMR Signals to Aid Brain Tumour Classification

6.1 Introduction

Ideally it is best to obtain maximum information from MR data. However, in reality this is not always possible, especially for *in vivo* data as optimum data acquisition conditions (especially RF pulse profiles and echo times) vary for different chemical compounds and their physical environments. It is possible to use the pulse sequence to enhance signals from one part of the spectrum and suppress those from others in order to maximise useful information from the resulting spectrum.

It has been shown that both glutamine and alanine are important features in discriminating between spectra of meningiomas and non-meningiomas. It has also been shown that both GA and GP can evolve a water suppression pulse for an NMR system. It is then useful to examine whether evolutionary computation methods can develop a pulse sequence to enhance the signals from alanine and glutamine in as realistic a sample as possible. Following work described in the previous chapter GA was used for these experiments as the representation of individuals fits better with the output required by pulse and other files required in a pulse sequence.

One approach to be tested is the optimisation of pulse shapes used in a spin-echo sequence. The most sensitive version of a spin-echo sequence has a precise 90^0 pulse followed by a precise 180^0 pulse acting on the frequency corresponding to the metabolites that are of interest. The standard 1-ī pulses, for example, are good for optimising one part of the spectrum at the expense of another. The alanine and glutamine regions are not contiguous and are separated by NAA in the spectrum and.

therefore this problem is likely to be more complex than can easily be solved with a 1-ī pulse. In a spin-echo sequence, the second pulse in the middle of the echo period, TE, refocuses the phase of the spins. (In imaging, this could also be done using gradients rather than a second pulse.) The resulting spectrum also depends on the effect of coupled spins (i.e. other protons separated by only a few chemical bonds) and whether or not they are affected by the pulses.

An alternative approach is to optimise an entirely new pulse sequence, including the selection of pulse shape, pulse power and duration and inter-pulse delays.

6.2 Methods

6.2.1 Sample and Experimental Setup

A sample was set up containing (approximately) 100mM of each of L-alanine (ala), L-glutamine (gln), N-acetyl-aspartate (NAA) and creatine (cr) in a deuterated water solution, the metabolites mimicking the conditions in a mixed brain tissue and tumour environment. Deuterated water, D_20, has the hydrogen replaced by deuterium. Although the use of water would give a more realistic experiment, early experiments showed that the optimisation process was dominated by the water suppression element.

Cells *in vivo* have a neutral pH and the chemical shift of the metabolites is based on this. The sample used here has not been controlled for pH so the chemical shift may be slightly altered from the normal values. The residual water peak has been set to be the reference position at 4.8ppm. A spectrum from this sample is shown in Figure 6.1.

In these experiments one region for each metabolite was used. The peaks on the right-hand side on this particular spectrum (1.0 - 3.5 ppm) tend to be larger than the peaks from the same metabolites in the rest of the spectrum (as they are often from CH_2 or CH_3 groups). The latter peaks tend also to be mixed with signals from other metabolites and are also relatively close to the signal from water.

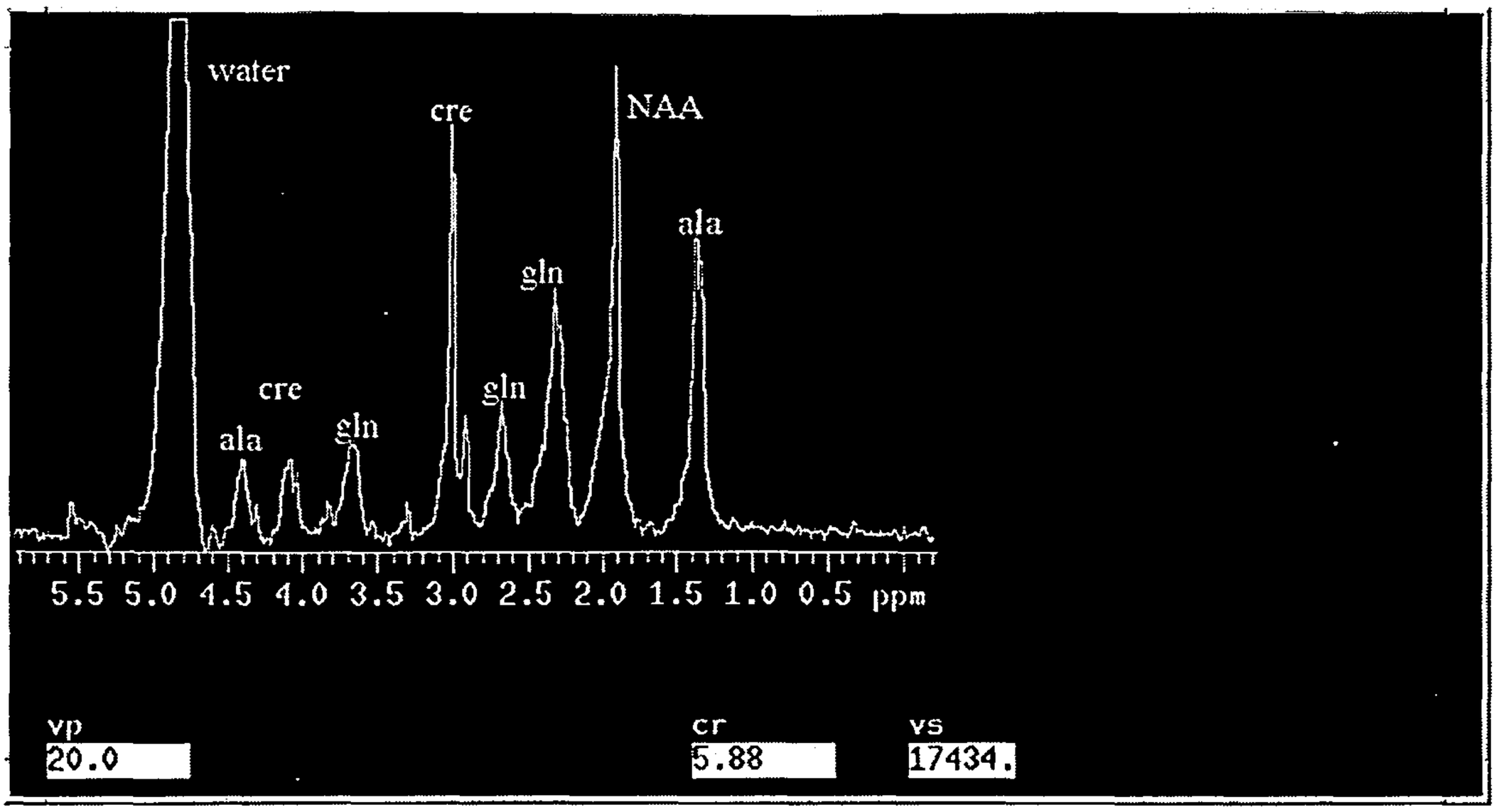


Figure 6.1 Spectrum from the experimental sample.

Experiments were set up to maximise the ratio of alanine plus glutamine to NAA plus creatine in order to gain maximum information from the alanine and glutamine signals.

Two sets of experiments were run with this sample, the first to evolve a spin-echo sequence, the second to evolve a sequence with elements selected from a predefined list of options.

With both methods, an acquisition element was automatically added to the end of the sequence. This was to ensure that each candidate solution would produce a spectrum which can be evaluated by the fitness function. As it is a constant, it is unnecessary to force the GA to evolve it. This acquisition element, plus other parameters including waits between pulse evaluations, are placed inside a macro read by the VNMR system and are therefore separated from the evolutionary process.

6.2.2 Spin-Echo Sequence

This experiment used a spin-echo sequence consisting of two pulses with two equal delay periods of half the echo time (TE/2) followed by an acquisition phase as shown in Figure 6.2. TE in this experiment was set to be 20ms. Two pulses of 20 two-

valued elements are to be evolved. The chromosome is set up to be a vector of size (40, 2) (as length is fixed, it need not be placed in the chromosome, but added prior to applying the sequence to the sample). The first 20 elements will be used as the phase and amplitude values for the first pulse, the latter 20 for the second pulse. As all phases are constrained by one range of values and all amplitudes by another there is no advantage to separating the vector into two. Crossover can take place at any point within the chromosome, as can mutation.

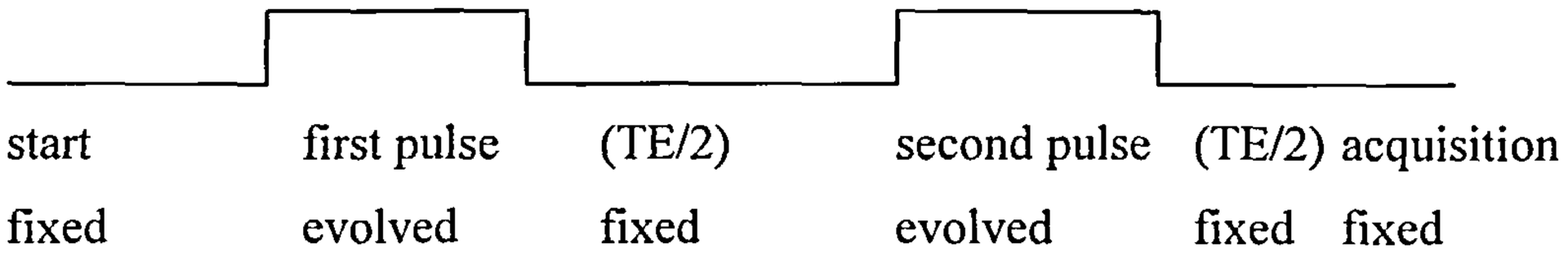


Figure 6.2 The spin-echo sequence.

The parameters for the first experiment are shown in Table 6.1

Population Size	20
Number of Generations	100
Fitness function	Formula 1 (see equation 1, p. 113)
Pulse sequence type	Spin echo
Length of pulses	2 pulses each of 20 elements
Delay between pulses	0.2seconds
Crossover type and rate	Simple 1 point 80%
Mutation type and rate	Single point 20%
Chromosome size	40 x 2 real valued elements
Number of Runs	10

Table 6.1 Parameters for the spin-echo experiments

6.2.3 Flexible Pulse Sequence

In this set of experiments the aim is to evolve a sequence without specifying that it is to be a spin-echo sequence. A set of existing pulses is made available to the system and the aim of the experiment is to create a sequence of pulses and delays that will

produce the desired result. The shape of the sequence is shown in Figure 6.3. Each chromosome, of length 13, consists of parameters for three pulses; initial delay, pulse type, length and power, plus a final delay. The element types are shown in Table 6.2 and the list of standard pulse shapes in Table 6.3. The inclusion of the 'End' pulse shape allows a sequence to be evolved that has fewer than three pulses as once a pulse shape 'End' is encountered, any subsequent pulses in the chromosome are ignored.

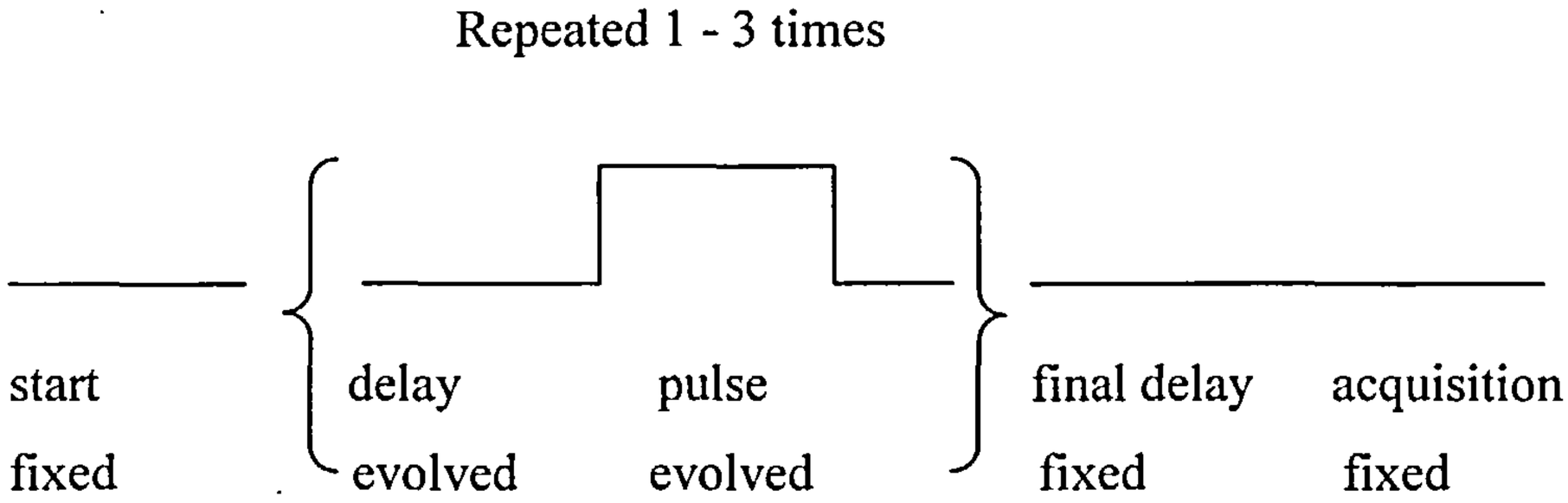


Figure 6.3 The flexible pulse sequence

The different types of value in different parts of the chromosome mean that crossover is constrained to be allowed only between elements of the same type, whilst mutation has different maximum allowable parameters for each element type.

The total time taken by each evaluation will be variable because of the differing length of total delays and number of pulses. The GA program waits for the results from all one generation's evaluations before applying the genetic operators to form the next generation.

In this experiment there will be a maximum number of pulses that can be utilised, both because the increasing complexity of multiple pulses will add considerably to execution time and because a large number of existing pulse sequences do not utilise large numbers of pulse files. In the first instance the maximum number is set to three. In (Fullmer and Miikkulainen 1992) a system to allow a varying number of neurons in a chromosome is described. In this work, the contents of each place in the chromosome are decoded (using a MOD function) to START, STOP and data bits. With this, there can be a varying number of nodes within the 800 bit

chromosome. The chromosome will be used to generate a neural network with the topology and weights determined by the chromosome. In the same way as this a variable number of pulse files and delays could be encoded into one chromosome to allow maximum flexibility to the evolutionary algorithm.

Element Number	Meaning	Type
1	Pre-pulse delay 1	0.0 - 1000.0 (milliseconds)
2	Pulse type 1	Integer 0 - 12
3	Pulse length 1	Integer 0 - 8000 (microseconds)
4	Power 1	0 - 40 (0.5 dB units)
5	Pre-pulse delay 2	0.0 - 1000.0 (milliseconds)
6	Pulse type 2	Integer 0 - 12
7	Pulse length 2	Integer 0 - 8000 (microseconds)
8	Power 2	0 - 40
9	Pre-pulse delay 3	0.0 - 1000.0 (milliseconds)
10	Pulse type 3	Integer 0 - 12
11	Pulse length 3	Integer 0 - 8000 (microseconds)
12	Power 3	0 - 40
13	Final Delay	0.0 - 1000.0 (milliseconds)

Table 6.2 Chromosome layout for the flexible pulse sequence

Pulse Number	Pulse Name	Pulse Type
0	Square	simple rectangular pulse
1	Sinc	sin(x)/x
2	Gauss	this pulse is amplitude modulated and selectively excites a bandwidth (Hz) approximately equal to 2e+6/pulse_length (μs).
3	hrm90	Hermitian. This pulse is amplitude modulated and provides a selective pulse optimised for 90-degree flips.
4	hrm180	Hermitian. This pulse is amplitude modulated and provides a selective pulse optimised for inversion and refocusing.
5	sech90	this pulse is phase modulated and converts longitudinal magnetization into transverse magnetization for spins close to resonance by adiabatic fast passage.
6	sech180	this pulse is phase modulated and provides a selective inversion pulse by adiabatic fast passage.
7	eburp1	selective excitation pulse reference: H. Geen and R. Freeman, JMR 93, 93 (1991) for 90 degree excitation only with Z initial magnetization
8	1-3-3-1	binomial
9	1-1	binomial
10	1-4-6	binomial
11	1-5-10	binomial
12	End	end marker

Table 6.3 Pulses available for use in the flexible pulse sequence experiments

6.2.4 GA Setup

GA using the GAOT within Matlab is used. This is because of the ease of linking the program to the VNMR program, which allows experiments to run directly on the hardware. The GA chromosome shape, a vector, suits the pulse shape required in earlier chapters, and although this advantage is less so with pulse sequences, where different parts of the chromosome will need to be interpreted as, for example, delays or pulses it is still easier to interpret than a GP tree-shaped individual would be.

The initial generation is randomly produced and all candidate solutions for that generation are presented to the MR system as a batch. A spectrum is produced for each and the height and integral of each peak is calculated. The height is used in the fitness function calculation

$$\text{Fitness} = \frac{X * (\text{height (alanine peak)} + \text{height (glutamine peak)})}{\text{abs (height (NAA peak)} + \text{height (creatine peak))}} \qquad (1)$$

where abs return the absolute (positive) value of its argument and X, a scaling factor, is initially set to be 100.

If the heights of the peaks are very small, using a division for the calculation can lead to very large fitnesses (even where the resulting spectrum is not good). In this case a modified fitness calculation is employed. It is useful to move from a ratio to an additive term which is much more stable.

$$\text{Fitness} = (X*(\text{height (alanine peak)}+\text{height (glutamine peak)})) - (\text{abs}(\text{height}(\text{NAA peak})+\text{height}(\text{creatine peak}))) \qquad (2)$$

where X is again initially set to 100.

6.3 Results

6.3.1 Spin-Echo Sequence

The results of the experiments are disappointing in comparison with those of the previous chapter. This experiment is much harder, with the aim to select alternate peaks from the spectrum. Improvements in fitness are of a much smaller magnitude with finesses improving between three to five times. Although the fitness improvements are low the resulting spectra do show the effects of the pulse files. The initial results of this experiment showed that both the best and the mean fitnesses did improve. Over the 10 runs the average improvement in fitness of the best chromosome was from 10 to 60 and the average improvement of the mean of the population was from 0 to 37. The mean and best in the population were still

improving at 100 generations, so a longer run may produce more fit solutions. Figure 6.4 shows the fitness of the mean of the population from a single run.

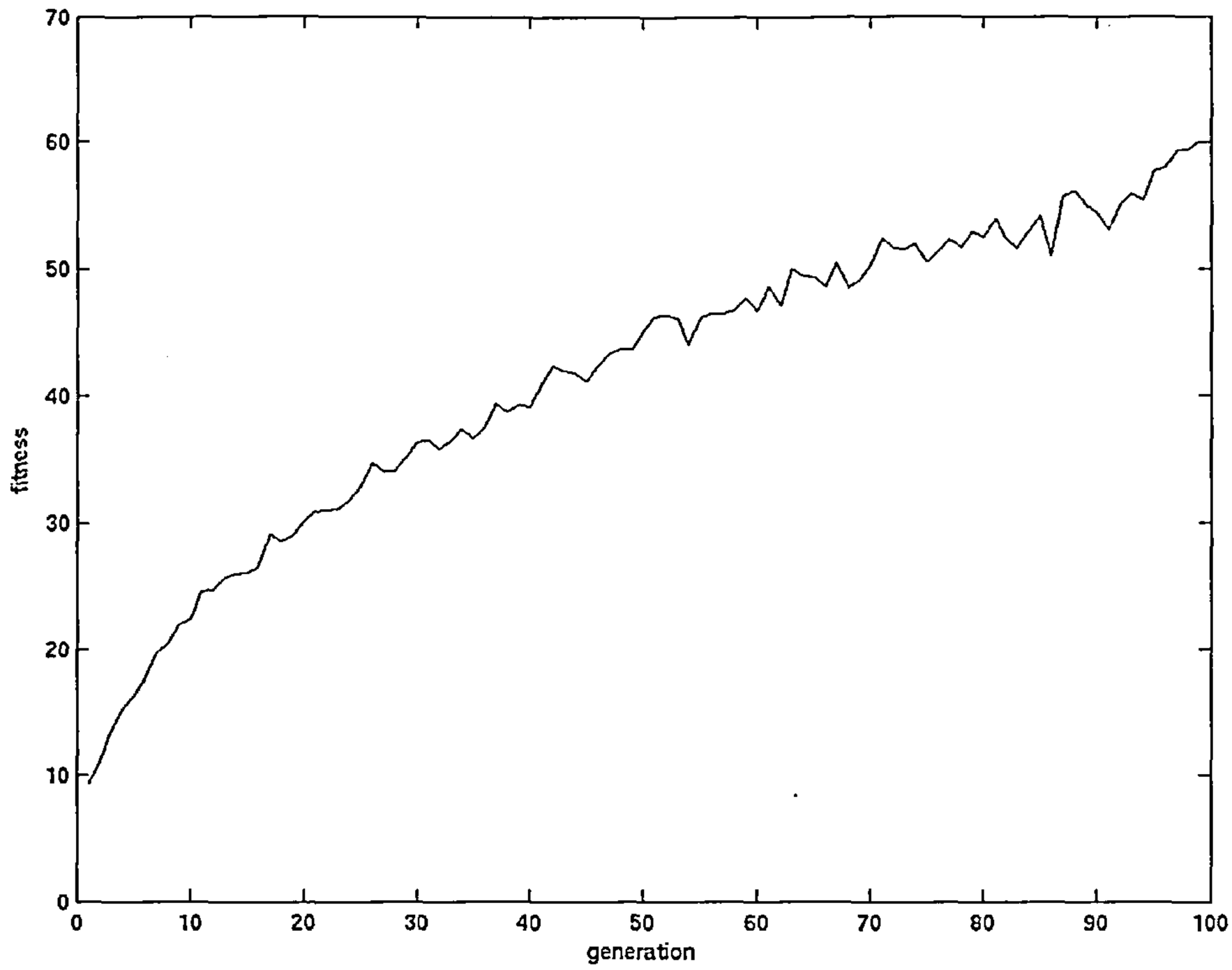
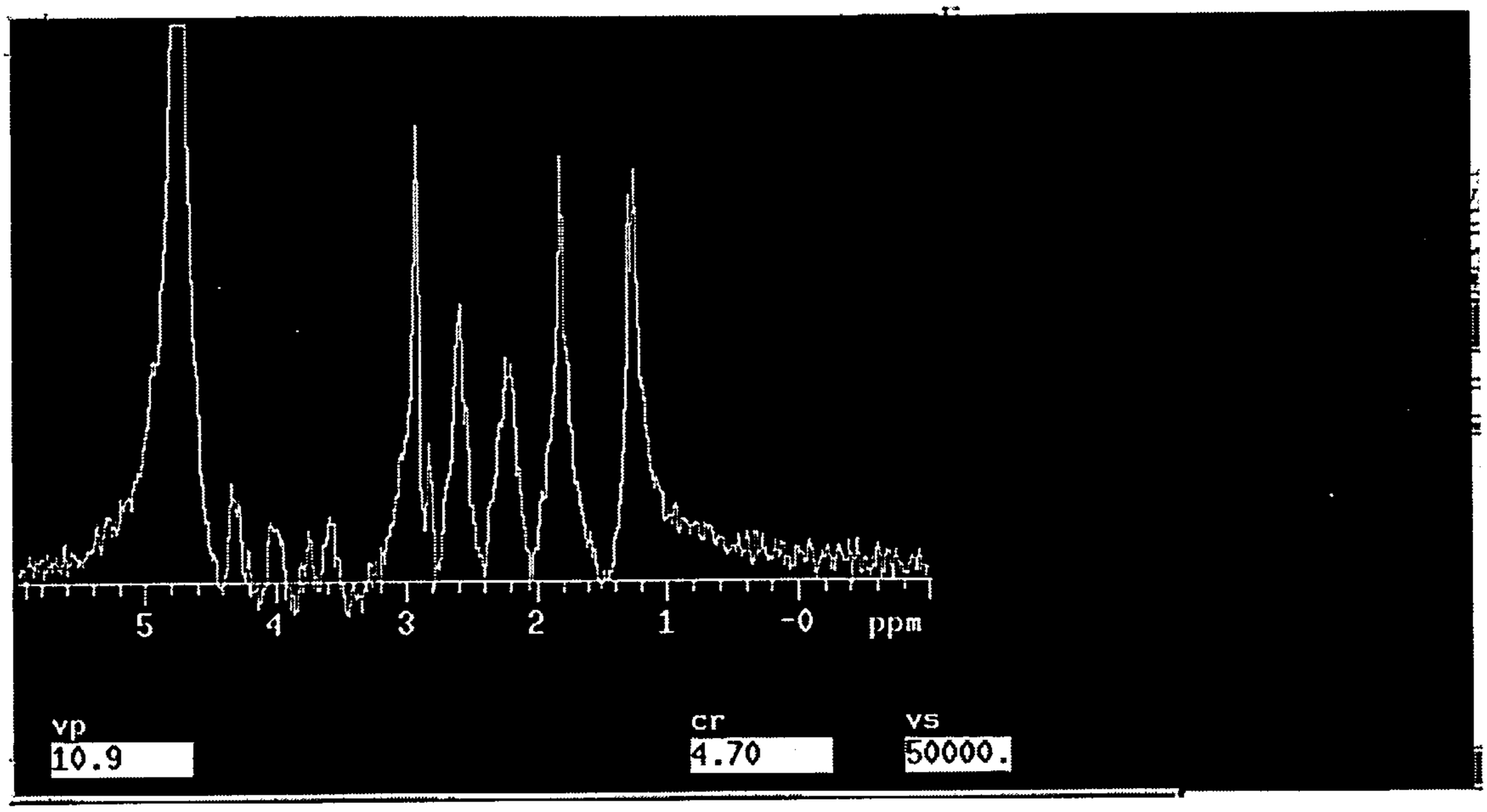


Figure 6.4 The average improvement of the best chromosome from the spin-echo sequence over 10 runs

6.3.2 Flexible pulse sequence

The results from this method also show a small improvement in the fitness of individuals. Figures 6.5 and 6.6 show two runs of the experiment. In each case the spectra produced by a pulse file at the beginning of a run and from a fitter individual at the end of the same run are shown. In both cases the fitness function subtracts the area of the NAA and creatine peaks from the area of the alanine and glutamine peaks. An example of a pulse sequence generated is shown in Figure 6.7.

a)



b)

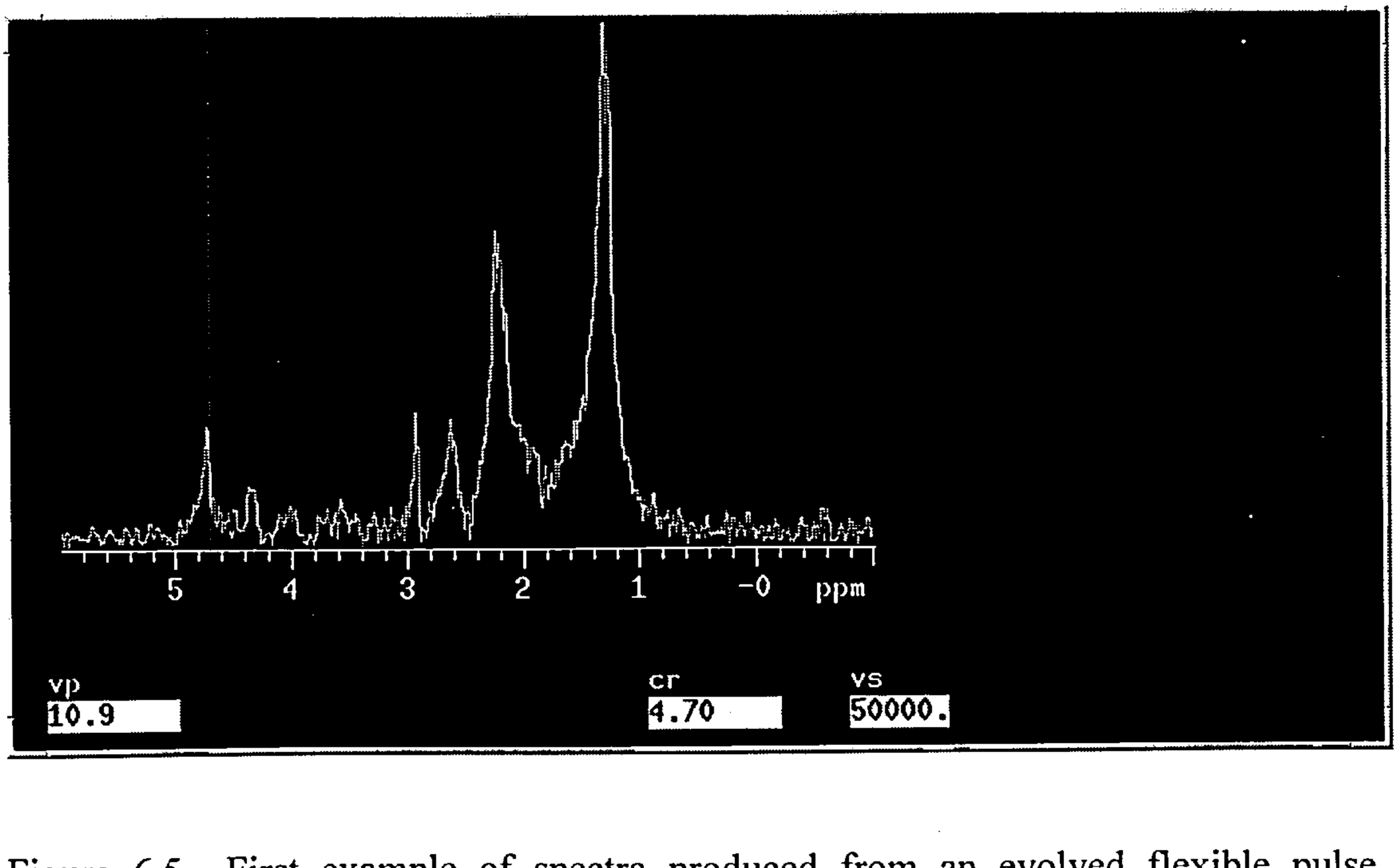
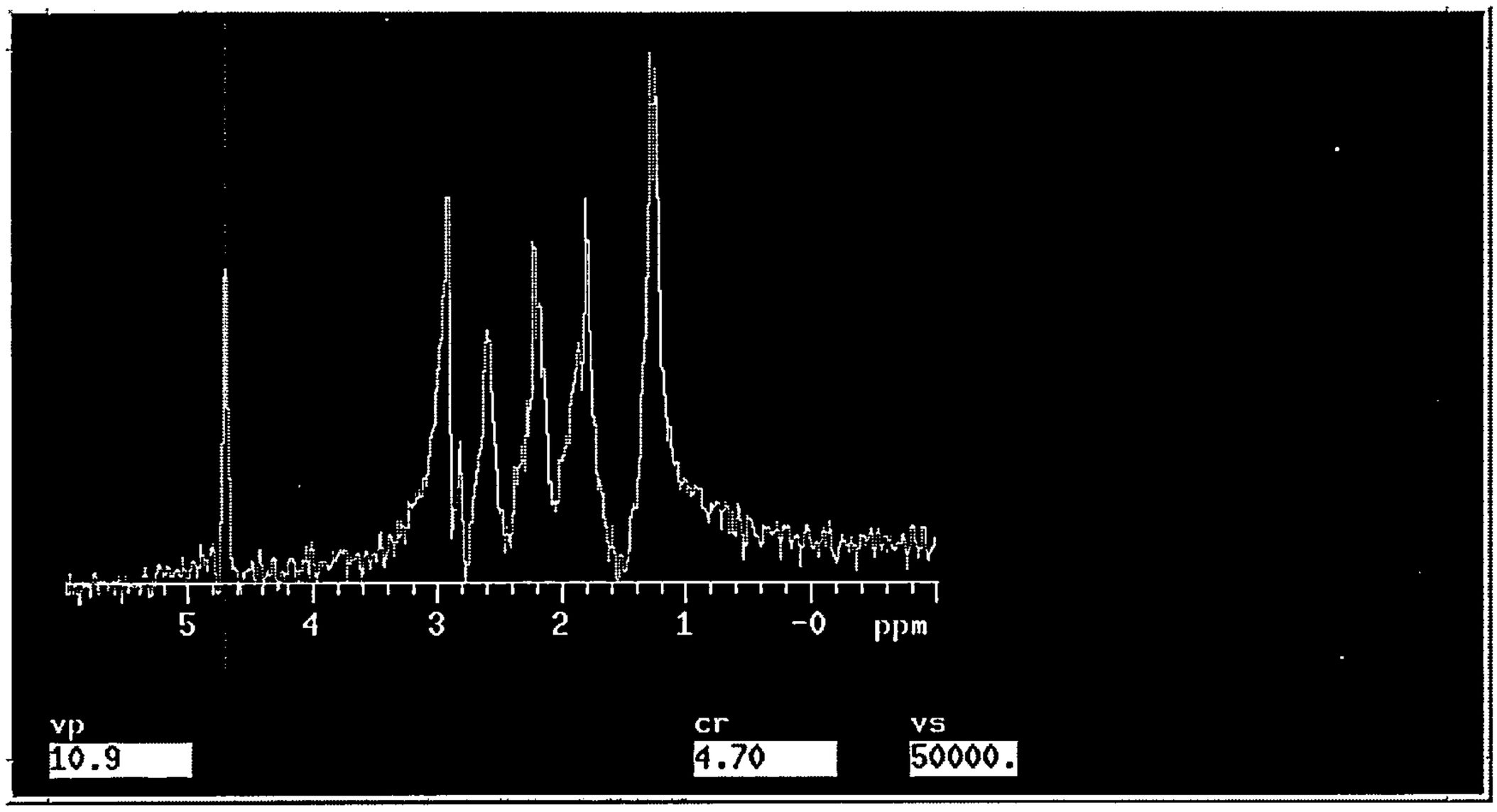


Figure 6.5 First example of spectra produced from an evolved flexible pulse sequence. a) an individual from the first generation b) an individual from the 43rd generation. The water peak at 4.8 has been coincidently suppressed in comparison with a). The peaks at 2.0 and 3.0ppm (NAA and creatine) have been reduced whilst that at 1.2ppm (alanine) is still strong. Glutamine at 2.5ppm is not particularly well enhanced.

a)



b)

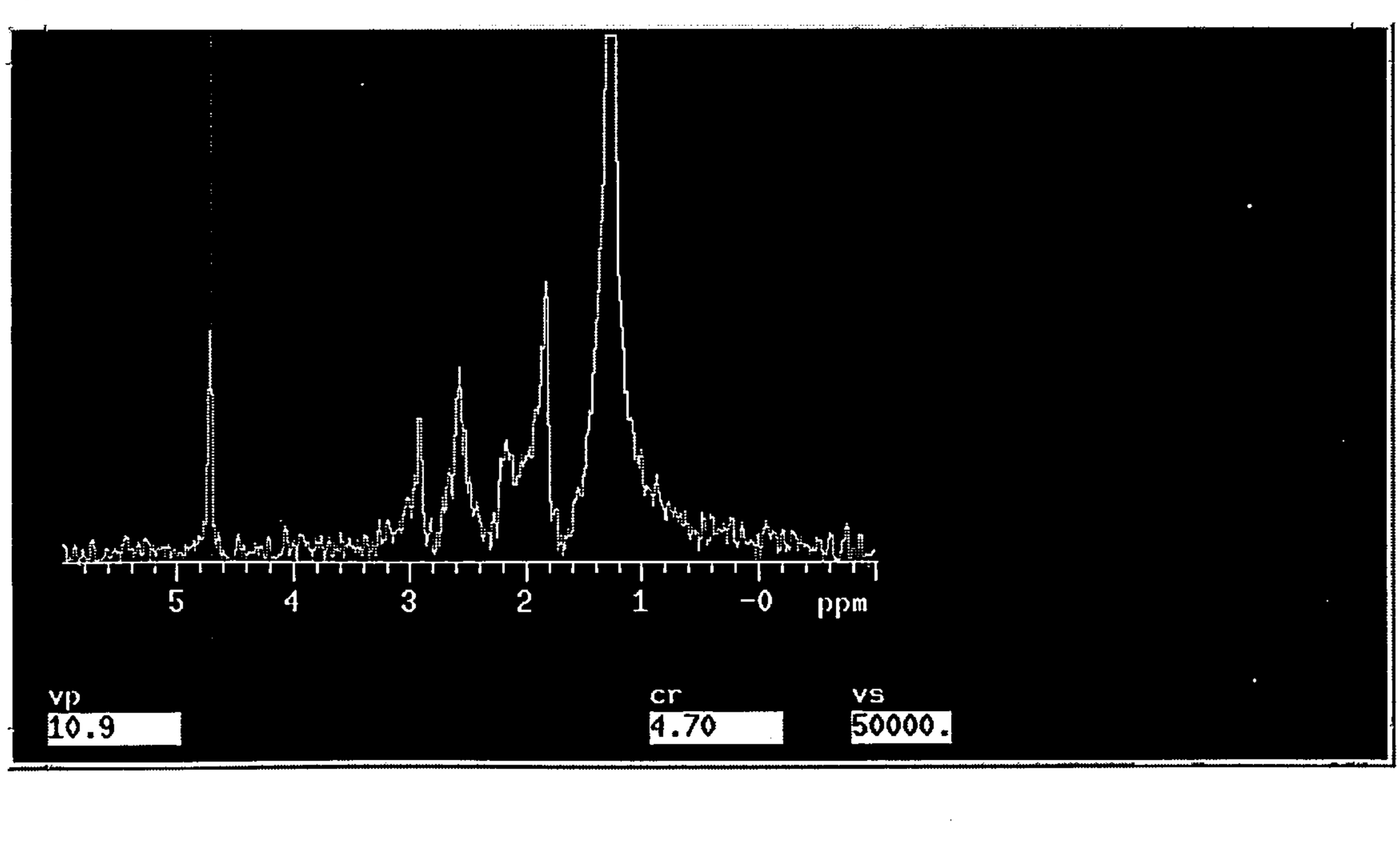


Figure 6.6 Second example of spectra produced from an evolved flexible pulse sequence. a) is from the first generation and b) from the 49th. Again the water peak is diminished, along with creatine and NAA, but again the glutamine peak has not been enhanced, although the alanine has.

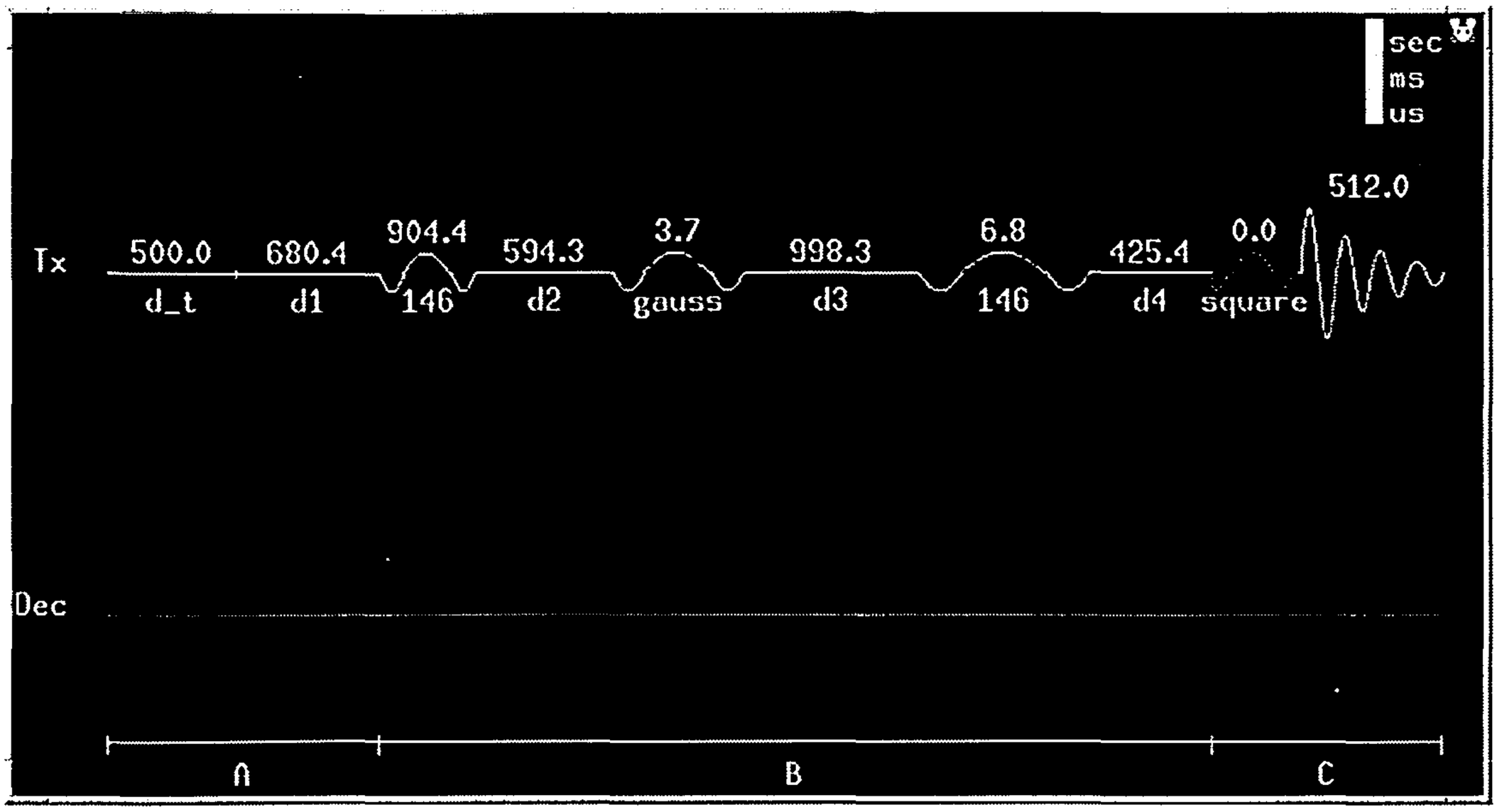


Figure 6.7 A sequence evolved using the flexible sequence method

6.4 Conclusion

The task is a complex one and GA in its most straightforward implementation struggles with it, given the nature of the problem where alternate peaks are to be minimised. However, GA does make some improvements.

Both methods had difficulty with the task. The spin-echo sequence is probably too restrictive a format for this task. The flexible sequence allows for more variation in the style of result and this may prove to be more fruitful in further experiments.

The advantage of running candidate solutions on hardware is that this is where successful pulses and pulse sequences would be run in a clinical setting. A more traditional approach would involve modelling of the system. Although the GA approach has had limited success on this occasion, the potential advantages of using it include the ability to capture non-linear and unpredictable pulses on real MR hardware and capturing novel solutions

A disadvantage of running experiments on hardware is that experiments need to be scheduled around competing demands for an expensive resource (the MR system)

and when there are problems with the MR system (as there were during the course of these experiments) then experiments are forced to be rescheduled or abandoned.

Chapter 7

Conclusions

7.1 Conclusions

The work presented here has focused on two ways that evolutionary computing can aid with medical uses of NMR; classifying data acquired from NMR spectroscopy to aid with cancer diagnosis and using evolutionary techniques to obtain pulse files and sequences.

In the area of classification, GP has proved that it can cope with two classes, and that the advantage over neural networks is that the features that aid the classification can be extracted relatively easily, especially where the solutions are simple. In the brain tumour diagnosis, the features that proved to be central to the classification included glutamine and alanine.

Principal component analysis was used to reduce the number of data points from the 400 involved in a digitised spectrum to twenty vectors. These were then rotated using varimax. The resulting principal component factors proved, in several cases, to contain signals mainly from a single metabolite. This made the interpretation of results much clearer from a biochemical perspective.

The use of medical data generally suffers from a problem of few samples on which to carry out classification. However, even with a dataset of 75 samples it was possible to divide it into separate training and test sets. Care needs to be taken with small data sets that the training my lead to over-fitting of the data to the samples given, which leads to poor generalisation ability. Although it is difficult to tell

whether the data has been over fitted, it is the case that those metabolites identified as having most discriminating capacity have been also identified by other research.

Genetic methods, both GA and GP, have been shown to be capable of dividing data into more than two groups. The use of a series of one-versus-the-rest binary classification tasks can obviously give solutions to such a task, however, it seems neater to try and solve the problem in a more parallel way. The methods of using one output for all classes, versus one output per class (with GP this would be done using one tree, or one tree per class) both have their advantages and disadvantages. If there is one tree, with the result being interpreted based on a range of values per class, then there can be no assignment of a sample to no class or more than one class. This does not mean that the answer cannot be wrong, but an area of confusion can be eliminated. The disadvantage of this method is that, in order to solve the whole problem in one step, the solutions can become unnecessarily complex.

The use of one tree per class, with a single point as a boundary, leads to simpler solutions. However, as each tree concentrates on a particular task, there is a possibility of samples being assigned to more than one class, or to none. A further step may need to be added to resolve such conflicts.

The novel solution described in this thesis is to use n tree for a maximum of 2^n classes. This allowed, for four classes, a two-way split in the data with two trees separately distinguishing between two different pairs of classes. In the tumour example, on division was entirely meaningful; tumour versus tissue, whilst the other was more contrived; a tissue and a tumour versus a different tissue and a different tumour. This method did prove successful though, with relatively simple solutions and relatively few conflicts to resolve.

The use of genetic methods to optimise pulse sequence elements, such as pulse files was intended as a method of tailoring signals from certain metabolites that had proved to be important in the classification of tumours. The next step from producing pulse files was to evolve more elements of a pulse sequence such as delays and pulse files with their power and length.

The results of this section can be considered successful as they prove that it is possible to attach an evolutionary technique to the software that runs a piece of hardware and to improve the results of such a technique. In the present case, it has proved more useful to use GA as the evolutionary technique as the representation of the individual matches much more that required by the system.

It is useful to be able to run experiments on the actual hardware as that can take into implicit consideration the nuances of the signal that would, with another method, need to be explicitly stated. Considerations of the hardware led to the use of a batch processing system to minimise the chances of an error in the running of an experiment. The batch processing also allowed for experiments to be run overnight or at weekends when the NMR system was not required by other users.

One disadvantage of using real hardware rather than a computer simulation is that the NMR system can have a fault which stops all acquisition of data.

For both the classification and pulse file generation, genetic methods have proved to be feasible but probably not the method of choice for the future. The use of GP over GA in both these situations is not shown here to be a better solution. It is indicative of the state of GP within the research community, that papers published within the last 12 months still describe what GP is, more than 10 years after the first conference on the subject.

7.2 Future Work

In the field of classification of MR data the introduction of multi-voxel MRSI signals is probably the most interesting development in respect of further work. Using both spatial and metabolic data from such data collection has been reported using statistical methods but as yet no application of EC to MRSI data has been reported. The use of binary classification, either with pre-processing of data or with a pairwise classification with post-processing to resolve conflicts and identify best solutions would be an interesting area. The aim of such classification is to produce clinically useful data and there is a need for a system to be robust and clear enough to allow it

to be used by clinicians. The identification of tissue type within a mixed tissue sample is information that would be very useful in a clinical setting and is there is need within MRSI data collection for this to be automated.

The use of EC methods to develop pulses and pulse sequences directly on the hardware has more scope for future work. The use of a sample of known quantities of specific metabolites allows for repeated experiments. The availability of an MR system is the limiting factor in such experimentation.

A specific area of interest is the use of gradients to affect the output of a pulse. Gradients were not used in the experiments reported here due to hardware problems but it would be interesting to include gradients as an extra parameter to the pulse or pulse sequence.

Other areas that may prove useful are how best to suppress the water peak, (given that pulses exist that do this successfully) and also how to enhance multiple non-adjacent peaks within a spectrum whilst minimising the signal from metabolites between the peaks of interest i.e. to make the data acquisition more metabolite specific and/or to help validate the presence of particular metabolites. In a given study, measurement of MRS signals from a range of metabolites is likely to remain important. However, it may be critical to obtain additional confirmation to confirm the identity of one (or more) of these signals.

Application of MRS for evaluation of potential drug targets and candidate drugs is of increasing importance (Workman *et al.* 2006). EC methods for optimising data acquisition (e.g. 'pulse sequence discovery') could be valuable in accelerating this part of the drug development process.

References

Ala-Korpela, M., K. K. Changani, et al. (1997). "Assessment of Quantitative Artificial Neural Network Analysis in a Metabolically Dynamic ex vivo P-31 NMR Pig Liver Study." Magnetic Resonance in Medecine **38**(5): 840 - 844.

Ala-Korpela, M., Y. Hiltunen, et al. (1996). "Artificial neural network analysis of H-1 nuclear magnetic resonance spectroscopic data from human plasma." Anticancer Res **16**(3B): 1473 - 1477.

Ala-Korpela, M., Y. Hiltunen, et al. (1995). "Quantification of biomedical NMR data using artificial neural network analysis: lipoprotein lipid profiles from ^{1}H NMR data of human plasma." NMR in Biomedicine **8**: 235 - 244.

Alam, T. M. and M. K. Alam (2005). "Chemometric Analysis of NMR spectroscopy data: A review." Annual Reports on NMR Spectroscopy **54**(54): 41 - 80.

Ali, B., A. Almaini, et al. (2004). "Evolutionary algorithms and their use in the design of sequential logic circuits." Genetic Programming and Evolvable Machines **5**(1): 11 - 29.

Aliferis, C. F., D. Hardin, et al. (2002). Machine Learning Models for Lung Cancer Classification Using Array Comparative Genomic Hybridization. Proceedings of AMIA Symposium.

Andris, P. and I. Frollo (2002). "Optimisation of NMR coils by genetic algorithms." Measurement Science Review **2**(2): 13 - 22.

Anthony, M., S. B, et al. (1994). "Pattern recognition classification of the site to nephrotoxicity based on metabolic data derived from proton nuclear magnetic resonance spectra of urine." Molecular Pharmacology **46**: 199 - 211.

Assion, A., T. Baumert, et al. (1998). "Control of chemical reactions by feedback-optimized phase-shaped femtosecond laser pulses." Science **282**: 919 - 922.

Axelson, D., I. J. Bakken, et al. (2002). "Applications of neural network analyses to in vivo H-1 magnetic resonance spectroscopy of Parkinson disease patients." Journal of Magnetic Resonance Imaging **16**(1): 13 - 20.

Baeck, T., D. Fogel, et al. (2000). Evolutionary computation 1: basic algorithms and operators, Institute of Physics.

Bakken, I., D. Axelson, et al. (2001). "Classification of *in vivo* ^{1}H MR spectra from breast tissue using artificial neural networks." Anticancer Res **21**(2B): 1481 - 1485.

Bakken, I. J., D. Axelson, et al. (1999). "Applications of neural network analyses to in vivo H-1 magnetic resonance spectroscopy of epilepsy patients." Epilepsy Research **35**(3): 245 - 252.

Balakrishnan, K. and V. Honavar (1996). On sensor evolution in robotics. Genetic Programming 1996, Stanford, USA, MIT Press.

Banzaf, W. (1998). Genetic Programming: An Introduction, Morgan Kaufmann.

Bathen, T. F., J. Krane, et al. (2000). "Quantification of plasma lipids and apolipoproteins by use of proton NMR spectroscopy, multivariate and neural network analysis." NMR Biomedicine **13**(5): 271 - 288.

Bayley, M. J., G. Jones, et al. (1998). "GENFOLD: a genetic algorithm for folding protein structures using NMR restraints." Protein Science **7**(2): 491 - 499.

Berthold, M. and D. J. Hand (1999). Intelligent Data Analysis: An Introduction, Springer-Verlag.

Bezdek, J. C., L. O. Hall, et al. (1993). "Review of MR image segmentation techniques using pattern recognition." Medical Physics **20**(4): 1033 - 1048.

Bishop, C. M. (2005). Neural networks for pattern recognition, Oxford University Press.

Blake, C. L. and C. J. Merz (1998). UCI Repository of machine learning databases, University of California, Department of Information and Computer Science.

Brameier, M. and W. Banzhaf (2001). "A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining." IEEE Transactions on Evolutionary Computation **5**(5): 17 - 26.

Branston, N. M., R. J. Maxwell, et al. (1993). "Generalisation performance using backpropogation algorithms applied to patterns derived from tumour ^{1}H NMR spectra." Journal of Microcomputer Applications **16**: 113 - 123.

Cramer, N. L. (1985). A representation for the adaptive generation of simple sequential programs. International Conference on Genetic Algorithms and their Applications, Pittsburgh.

De Falco, I., A. Della Cioppa, et al. (2002). "Discovering interesting classification rules with genetic programming." Applied Soft Computing **1**: 257 - 269.

de Vega, F. F., J. I. Hidalgo, et al. (2004). "A methodology for reconfigurable hardware design based upon evolutionary computation." Microprocessors and Microsystems **28**(7): 363 - 371.

De Vos, M., T. Laudadio, et al. (2007). "Fast nosologic imaging of the brain." Journal of Magnetic Resonance **184**: 292 - 301.

Dounias, G., H. Axer, et al. (2002). Genetic Programming for the Generation of Crisp and Fuzzy Rule Bases in Classification and Diagnosis of Medical Data. First International NAISO Congress on Neuro Fuzzy Technologies, Havana, Cuba, NAISO (Natural and Artificial Intelligence Systems Organization).

Dracopoulos, D. C. and S. Kent (1997). "Genetic programming for prediction and control." Neural Computing and Applications **6**: 214 - 228.

El-Deredy, W., S. M. Ashmore, et al. (1997). "Pretreatment prediction of the chemotherapeutic response of human glioma cell cultures using nuclear magnetic

resonance spectroscopy and artificial neural networks." Cancer Research **57**(19): 4196 - 4199.

Emmanouilidis, C., A. Hunter, et al. (1999). Multiple-criteria genetic algorithms for feature selection in neurofuzzy modeling. International Joint Conference on Neural Networks, Washington USA, IEEE.

Fogel, D. B. (1994). "An introduction to simulated evolutionary optimization." IEEE Transactions on Neural Networks **5**(1): 3 - 14.

Freeman, R. and X. Wu (1987). "Design of magnetic resonance experiments by genetic evolution." Journal of Magnetic Resonance **75**: 184 - 189.

Fullmer, B. and R. Miikkulainen (1992). Using marker-based genetic encoding of neural networks to evolve finite-state behaviour. Proceedings of the First European Conference on Artificial Life, MIT Press.

Gadian, D. G. (1995). NMR and its applications to living systems, Oxford University Press.

Geen, H. and R. Freeman (1991). "Band-selective radiofrequency pulses." Journal of Magnetic Resonance **93**: 93 - 141.

Gerstle, R. J., S. R. Aylward, et al. (2000). "The role of neural networks in improving the accuracy of MR spectroscopy for the diagnosis of head and neck squamous cell carcinoma." American Journal of Neuroradiology **21**: 1133 - 1138.

Goldberg, D. E. (1989). Genetic algorithms in search, optimisation and machine learning. Reading, Massachusetts, Addison-Wesley.

Gray, H. F., R. J. Maxwell, et al. (1998). "Genetic programming for classification and feature selection: analysis of ^{1}H nuclear magnetic resonance spectra from human brain tumour biopsies." NMR Biomedicine **11**: 217 - 224.

Gribbestad, I. S., B. Sitter, et al. (1999). "Metabolic composition in breast tumours examined by proton nuclear magnetic resonance spectroscopy." Anticancer Res **19**(3A): 1737 - 1746.

Griffiths, J. R. (1996). A multicentre trial on MR spectroscopy of human tissue biopsies. Eurospin annual 1995-1996. F. Podo. Rome, Instituto Superiore di Sanita: 295-299.

Gruber, S., A. Stadlbauer, et al. (2005). "Proton magnetic resonance spectroscopic imaging in brain tumour diagnosis." Neurosurgery Clinics of North America **16**(1): 101.

Haas, O. (1999). Radiotherapy Treatment Planning New System Approaches, Springer-Verlag.

Hagberg, G. (1998). "From magnetic resonance spectroscopy to classification of tumors. A review of pattern recognition methods." NMR in Biomedicine **11**: 148 - 156.

Hallinen, J. (2001). Feature selection and classification in the diagnosis of cervical cancer. The Practical Handbook of genetic Algorithms; Applications. L. D. Chambers, Chapman and Hall: 167 - 202.

Hiltunen, Y., J. Kaartinen, et al. (2002). "Quantification of human brain metabolites from in vivo H-1 NMR magnitude spectra using automated artificial neural network analysis." Journal of Magnetic Resonance **154**(1): 1 - 5.

Hinchliffe, M., H. Hiden, et al. (1996). Modelling Chemical Process Systems Using a Multi-Gene Genetic Programming Algorithm. Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996, Stanford University, CA, USA, Stanford Bookstore Stanford University, Stanford, California 94305-3079, USA.

Holland, J. H. (1975). Adaption in Natural and Artificial systems. Ann Arbor, USA, University of Michigan Press.

Holmes, E., A. W. Nicholls, et al. (2000). "Chemometric models for toxicity classification based on NMR spectra of biofluids." Chem Res Toxicology **13**(6): 471 - 478.

Holmes, E., A. W. Nicholls, et al. (1998). "Development of a model for classification of toxin-induced lesions using ^{1}H NMR spectroscopy of urine combined with pattern recognition." NMR in Biomedicine **11**: 235 - 244.

Hornak, J. (1997-1999). The basics of NMR http://www.cis.rit.edu/htbooks/nmr/.

Houck, C., J. Joines, et al. (1995). A genetic algorithm for function optimisation: a Matlab implementation, North Carolina State University.

Huang, Y., P. Lisboa, et al. (2003). "Tumour grading from magnetic resonance spectroscopy: a comparison of feature extraction with variable selection." Stat Med **22**(1): 147 - 164.

Inza, I., M. Merino, et al. (2001). "Feature Subset Selection by Genetic Algorithms and Estimation of Distribution Algorithms. A case study in the survival of Cirrhotic patients treated with TIPS." Artificial Intelligence in Medicine **23**(2): 187 - 205.

Kaiser, H. F. (1958). "The varimax criterion for analytic rotation in factor analysis." Psychometrika **23**: 187-200.

Kishore, J. K., L. M. Patnaik, et al. (2000). "Application of genetic programming or multicatagory pattern classification." IEEE Transaactions on Evolutionary Computation **4**(3): 242 - 258.

Kishore, J. K., L. M. Patnaik, et al. (2001). "Genetic programming based pattern classification with feature space partitioning." Information Sciences **131**: 65 - 86.

Konstam, A. (1998). Group classification using a mix of genetic programming and genetic algorithms. Symposium on Applied Computing, SAC98, Atlanta.

Koza, J. (1992). Genetic programming:on the programming of computers by means of natural selection. Cambridge, Massachusetts, MIT Press.

Koza, J. (1994). Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press.

Koza, J., D. Andre, et al. (1998). Genetic Programming III: Darwinian Invention and Problem Solving, Morgan Kaufmann.

Koza, J., M. Keane, et al. (2003). Genetic Programming IV: Routine Human-Competitative Machine Intelligence, Kluwe Academic Publishers.

Krasnogor, N., W. E. Hart, et al. (1999). Protein structure prediction with evolutionary algorithms. Proceedings of the Genetic and Evolutionary Computation Conference, Orlando, Morgan Kaufmann.

Langdon, W. B., S. J. Barrett, et al., Eds. (2004). Genetic Programming for Combining Neural Networks for Drug Discovery, Physica Verlag.

Langdon, W. B. and B. F. Buxton (2004). "Genetic programming for Mining DNA Chip data from Cancer Patients." Genetic Programming and Evolvable Machines **5**: 251 - 257.

Laudadio, T., M. Martinez-Bisbal, et al. (2007). Fast Nosological Imaging of 2DTSI Brain Data Using Canonical Correlation Analysis. International Society of Magnetic Resonance in Medicine.

Lee, Y. Y. B., Y. Huang, et al. (2000). "Robust methodology for the discrimination of brain tumours from *in vivo* magnetic resonance spectra." IEE Science, Measurement and Technology **147**(6): 309 - 314.

Lisboa, P., A. Kirby, et al. (1998). "Assessment of statistical and neural networks methods in NMR spectral classification and metabolite selection." NMR in Biomedicine **11**: 225 - 234.

Lisboa, P. J. G., A. R. Mehridehnavi, et al. (1993). The interpretation of supervised neural networks. Workshop on neural network applications and tools, IEEE Computer Society Press.

Malinowski, E. R. (1987). "Theory of distribution of error eigenvalues resulting from principal component analysis with applications to spectroscopic data." Chemometrics J **1**: 33-40.

Marmelstein, R. E. and G. B. Lamont (1998). Pattern classification using a hybrid genetic programming - decision tree approach. Genetic Programming 1998: Proceedings of the Third Annual Conference, Morgan Kaufmann.

Maxwell, R. J., I. Martinez-Perez, et al. (1998). "Pattern recognition analysis of ^{1}H NMR spectra from perchloric acid extracts of human brain tumour biopsies." Magnetic Resonance in Medicine.

McIntyre, A. R. and M. I. Heywood (2005). Toward Co-evolutionary Training of a multi-class classifier. IEEE Congress on Evolutionary Computation, Edinburgh, IEEE Press.

Michalewicz, Z. (1996). Genetic algorithms + data structures = evolution programs, Springer.

Mitchell, M. (1998). An Introduction to Genetic Algorithms, MIT Press.

Montana, D. J. (1993). Strongly typed genetic programming. Cambridge, Massachusetts, Bolt Beranek and Newman Inc.

Mountford, C. (2001). "Diagnosis and prognosis of breast cancer by magnetic resonance spectroscopy of fine-needle aspirates analysed using a statistical classification strategy." British Journal of Surgery **89**(9): 1234 - 40.

Muni, D. P., N. R. Pal, et al. (2004). "A novel approach to design classifiers using genetic programming." IEEE Transaactions on Evolutionary Computation **8**(2): 183 - 196.

Oltean, M. and C. Grosan (2003). Solving classification problems using Infix Form Genetic Programming, citeseer.ist.psu.edu/oltean03solving.html.

Özkan, M., B. M. Dawant, et al. (1993). "Neural-network-based segmentation of multi-modal medical images: a comparative and prospective study." IEEE Transactions on Medical Imaging **12**(3): 534 - 544.

Pao, Y.-H. (1989). Adaptive pattern recognition and neural networks. Reading, Massachusetts, Addison-Wesley Publishing Company.

Pearlman, D. A. (1996). "FINGAR: a new genetic algorithm-based method for fitting NMR data." Journal of Biomolecular NMR **8**: 49 - 66.

Pena-Reyes, C. A. and M. Sipper (2000). "Evolutionary computation in medicine: an overview." Artificial Intelligence in Medicine **19**: 1 - 23.

Pena-Reyes, C. A. and M. Sipper (2001). "Fuzzy CoCo: A Cooperative Coevolutionary Approach to Fuzzy Modeling." IEEE Transactions on Fuzzy Systems **5**(5): 100 - 110.

Piccolboni, A. and G. Mauri (1998). Application of evolutionary algorithms to protein folding prediction. Artificial Evolution: Third European Conference AE '97, Nimes, France, Springer-Verlag.

Pinkus, A. (1999). "Approximation theory of the MLP model in neural networks." Acta Numerica: 143 - 195.

Poli, R. (1996). Genetic programming for feature detection and image segmentation. Evolutionary Computing: AISB Workshop. T. C. Fogarty, Springer-Verlag. **LNCS 1143:** 110 - 125.

Poli, R. and S. Cagnoni (1997). Genetic programming with user-driven selection: experiments on the evolution of algorithms for image enhancement. Genetic Programming 1997, Morgan Kauffman Publishers.

Poptani, H., J. Kaartinen, et al. (1999). "Diagnostic assessment of brain tumours and non-neoplastic brain disorders in vivo using proton nuclear magnetic resonance spectroscopy and artificial neural networks." Journal of Cancer research and Clinical Oncology **125**(6): 343 - 349.

Prechelt, L. (1994). PROBEN1 - A set of Neural Network Benchmark Problems and Benchmarking Rules, University of Karlsruhe.

Prost, R. W., M. L, et al. (1997). "Detection of glutamate/glutamine resonance by ^{1}H magnetic resonance spectroscopy at 0.5 Tesla." Magn Reson Med **37**: 615-618.

Pulkkinen, J., M. Lappalainen, et al. (2002). Application of self-organising maps in automated chemical shift correction of in vivo H-1 MR spectra. Intelligent Data Engineering and Automated Learning - IDEAL, Springer Verlag.

Raymer, M. L., W. F. Punch, et al. (1997). Simultaneous feature extraction and selection using a masking genetic algorithm. ICGA-97.

Remy, C., C. Arus, et al. (1994). "In vivo, ex vivo and in vitro one- and two-dimensional nuclear magnetic resonance spectroscopy of an intracerebral glioma in rat brain: assignment of resonances." J Neurochem **62**: 166-179.

Rethinam, M. J., A. K. Javali, et al. (2004). A genetic algorithm for finding pulse sequences for NMR quantum computing. **2004**.

Reyment, R. and K. G. Jöreskog (1993). Applied factor analysis in the natural sciences. Cambridge, Cambridge University Press.

Rutter, A., H. Hugenholtz, et al. (1995). "Classification of brain tumours by ex-vivo ^{1}H NMR spectroscopy." J Neurochem **64**(4): 1655 - 1661.

Sammon, J. W. (1969). "A nonlinear mapping for data structure analysis." IEEE Transactions on Computers **C-18**(5): 401 - 409.

Schiffmann, W., M. Joost, et al. (1992). Synthesis and Performance Analysis of Multilayer Neural Network Architectures, University of Koblenz.

Schild, H. H. (1990). MRI made easy, Schering.

SDBSWeb http://www.aist.go.jp/RIODP/SDBS/05072001.

Sierra, B. and P. Larranaga (1998). "Predicting the Survival in Malignant Skin Melanoma Using Bayesian Networks Automatically Induced by Genetic Algorithms. An Empirical Comparison between different Approaches." Artificial Intelligence in Medicine **14**: 215 - 230.

Silva, S. and Y.-T. Tseng (2005). Classification of seafloor habitats using genetic programming. Genetic and Evolutionary Computation Conference (GECCO'2005).

Smart, W. and M. Zhang (2005). Using genetic programming for multiclass classification by simultaneously solving component binary classification problems. 8th European Conference on Genetic Programming, EuroGP 2005, Lausanne, Springer-Verlag.

Somorjai, R. L., A. E. Nikulin, et al. (1995). "Computerized Consensus Diagnosis: A Classification Strategy for the Robust Analysis of MR spectra. I. Application to 1H Spectra of Thyroid Neoplasms." Magnetic Resonance Medicine **33**(2): 257--263.

Stoyanova, R. and T. Brown (2002). "NMR spectral quantitation by principal componant analysis. III. A generalised procedure for determination of lineshape variations." Journal of Magnetic Resonance **154**(2): 163 - 175.

Tate, R., J. Griffiths, et al. (1998). "Towards a method for automated classification of ^{1}H MRS spectra from brain tumours." NMR in Biomedicine **11**(4/5): 177 - 191.

Teller, A. and M. Veloso (1995). "Program Evolution for Data Mining." The International Journal of Expert Systems **8** **http://www.cs.cmu.edu/afs/cs/usr/astro/public/papers/Astro-ESJ.ps**(3): 216--236.

Thompson, A. (1996). Silicon evolution. Genetic Programming 1996, MIT Press.

Thomsen, J. and B. Meyer (1989). "Pattern Recognition of the ^{1}H NMR Spectra of Sugar Alditols Using a Neural Network." Journal of Magnetic Resonamce **84**: 212 - 217.

Unger, R. and J. Moult (1993). "Genetic algorithms for protein folding simulations." Journal of Molecular Biology **231**(1): 75 - 81.

Usenius, J.-P.; S. Tuohimetsa, et al. (1996). "Automated classification of human brain tumours by neural network analysis using in-vivo ^{1}H magnetic resonance spectroscopic metabolite phenotypes." Neuroreport **7**: 1597-1600.

van den Broek, W., D. Wienke, et al. (1997). "Optimal wavelength range selection by a genetic algorithm for discrimination purposes in spectroscopic infrared imaging." Applied Spectroscopy **51**(8): 1210 - 1217.

Wang, C.-H., S.-S. Tseng, et al. (1995). "Design of a self-adaptive brain tumour diagnostic system." Journal of Information Science and Engineering **11**: 275 - 294.

Wehrens, R., C. Lucasius, et al. (1993). "Sequential Assignment of 2D-NMR Spectra of Proteins Using Genetic Algorithms." Jounal of Chemical Information and Computer Science **33**: 245 - 251.

Westbrook, C. and C. Kaut (1993, 1998). MRI in practice, Blackwell.

Workman, P., E. O. Aboagye, et al. (2006). "Minimally invasive pharmacokinetic and pharmacodynamic technologies in hypothesis-testing clinical trials of innovative therapies." Journal of the National Cancer Institute **98**(9): 580 - 598.

Worth, A. J. and D. N. Kennedy (1994). "Segmentation of magnetic resonance brain images using analogue constraint satisfaction neural networks." Image and Vision Computing **12**(6): 345 - 354.

Yang, J. and V. Honavar (1997). Feature Subset Selection Using a Genetic Algorithm. Genetic Programming 1997: Proceedings of the Second Annual Conference, Morgan Kaufmann.

Zhang, M., V. B. Ciesielski, et al. (2003). "A domain-independent window approach to multiclass object detection using genetic programming." EURASIP Journal on Applied Signal Processing **8**: 841 - 859.

Zhang, M. and W. Smart (2004). Multiclass object classification using genetic programming. EvoWorkshops 2004, Springer-Verlag.

Zongker, D. and W. F. Punch lilgp 1.0 User's Manual, MSU Genetic Algorithms and Research Application Group (GARAGe).