THE ASSIGNMENT OF PROTEIN NMR SPECTRA USING A GENETIC ALGORITHM

Thesis Submitted for the degree of

Doctor of Philosophy

at the University of Leicester

by

Bartlet Gilbert Ailey BSc (Swansea) MSc (Cardiff)

Biological NMR Centre

Biochemistry Department

University of Leicester

September 1997

To my Father

ABSTRACT

THE ASSIGNMENT OF PROTEIN NMR SPECTRA USING A GENETIC ALGORITHM

BARTLETT G AILEY

NMR spectroscopy is one of the two methods for determining the structures of proteins. The production of a structure using NMR has a number of phases; with the assignment phase being one of the most time consuming. Any automation, even partial, of the assignment process would be of enormous benefit. This thesis describes five modules (2D-SAM, 3D-SAM, BAM-1, BAM-2 and SCAM) that use a genetic algorithm (GA) to assign protein NMR spectra. The 2D-SAM and 3D-SAM are Sequential Assignment Modules. They take the relevant spin system identification and sequentially assign either a 2 dimensional homonuclear or 3 dimensional heteronuclear NOESY spectra. The 2D-SAM is effective with small proteins which generate high quality spectra while the 3D-SAM is effective with larger isotopically labelled proteins. The BAM-1 and BAM-2 are Backbone Assignment Modules. The BAM-1 takes several triple resonance spectra and assigns the peaks to relevant nuclei creating peak systems. The BAM-2 takes the peak systems and sequentially assigns them. The SCAM is a prototype Side Chain Assignment Module; it is designed to take either a HCCH C^{13} TOCSY or COSY spectrum and assign its peaks to certain types of amino acid. The BAM-1, BAM-2, and SCAM were designed to work in sequence to assign a whole protein. Although each module is designed to assign a specific type of spectrum or spectra they are all based around the same GA core. This core uses a crowding factor, phenotypic domain specific genetic operators and a novel age concept to improve its performance. When evaluated the performance of each module (average correct assignment) was 2D-SAM 100%, 3D-SAM 71%, BAM-1 96% and BAM-2 75%.

ACKNOWLEDGEMENTS

I would like to acknowledge the advice, encouragement and freedom given to me by my supervisor Gordon Roberts

I would like to thank the other members of my PhD committee: Arthur Rowe and Nick Measor for the time they gave to me and the project.

I would like to thank the Biotechnology and Biological Sciences Research Council for funding my PhD.

I would also like to thank the members of the Biological NMR Centre in Leicester for their help throughout my PhD, particularly:

Igor Barsukov
Amrik Basran
Jaswir Basran
Marco Cassarotto
Pillip Dawson
Bernadette Hawkins
Peter Kilby
Lu-yun Lian
"Spud" Kong Hong Sze
Gillian Walford
Jan White

Bissan

My Mother, for everything.

TABLE OF CONTENTS

1.0 Introduction..1

2.0 The Assignment of Protein NMR Spectra..5

2.1 Producing A Solution Structure 5

2.1.1 Sample Preparation..5

2.1.2 NMR Experiments...6

2.1.3 Interpretation of Protein NMR Spectra 8

2.1.4 Determination of Three Dimensional Structure of a Protein9

2.2 The Use of Two Dimensional Homonuclear ^{1}H NOESY Spectra to perform Sequential Assignment..10

2.2.1 Assignment Rules for NOESY Spectra..............................10

2.2.2 Assignment Ambiguities 11

2.3 The Use of Three Dimensional Heteronuclear ^{15}N NOESY Spectra to Perform Sequential Assignment..13

2.3.1 Assignment Rules..14

2.3.2 Assignment Ambiguities 14

2.4 The Use of Triple Resonance Spectroscopy to Perform Backbone Assignment..15

2.4.1 Assignment Rules..15

2.4.2 Assignment Ambiguities 16

2.5 Performing Spin System Assignment Using HCCH ^{13}C TOCSY and COSY Spectroscopy..16

2.5.1 Assignment Rules..17

2.5.2 Assignment Ambiguities 17

3.0 Evolutionary Strategy...19

3.1 A Simple Genetic Algorithm...19

3.1.1 Population...19

3.1.2 Mutation 20

3.1.4 Fitness..22

3.1.5 Reproduction...23

3.1.6 Running a Genetic Algorithm 24

3.2 Genetic Algorithm Theory 25

3.2.1 Schema...25

3.2.3 Schema Theorem 26

3.2.4 Search Space...27

4.0 Design of the Genetic Algorithm Core 30

4.1 GA Core Design Principles 30
4.2 Individual Object 35
4.2.1 Individual Class Definition 35
4.2.2 Individual Class Functions 35
4.3 Population Class 36
4.3.1 Population Class Definition 36
4.4 Main Population Class 36
4.4.1 Main Population Definition 37
4.4.2 Main Population Class Functions 37
4.5 Sub Population Class 40
4.5.1 Sub Population Class definition 40
4.5.2 Sub Population Class Functions 40
4.6 Problem Domain Dependent Elements 43
4.6.1 Coding 44
4.6.2 Fitness Function 44
4.6.3 Genetic Operators 45
4.7 GA Core Object Definitions 45
4.7.1 Chromosome Class Attributes 45
4.7.2 Individual Class Attributes 46
4.7.3 Population Class Attributes 48
4.7.4 Main Population Class Attributes 48
4.7.5 Sub Population Class Attributes 49

5.0 Two Dimensional Sequential Assignment Module 51
5.1 Design of the 2D-SAM 51
5.1.1 Coding of 2D-SAM 52
5.1.2 Fitness Function of2D-SAM 55
5.1.3 Genetic Operators of 2D-SAM 58
5.2 Evaluation of the 2D-SAM 61
5.2.1 Testing of the 2D-SAM 61
5.2.2 Conclusions 64
5.3 2D-SAM Object Definitions 67
5.3.1 Peak Class 67
5.3.2 Peal List Class 68
5.3.3 Spin System Class 69
5.3.4 Connection Class 69
5.3.5 Connections Class 71

6.0 Three Dimensional Sequential Assignment Module 72
6.1 Design of the 3D-SAM 73

6.1.1 Coding 3D-SAM 73
6.1.2 Fitness function For 3D-SAM 74
6.2 Evaluation of the 3D-SAM 75
6.2.1 Testing of the 3D-SAM 75
6.2.2 Conclusions 76

7.0 Backbone Assignment Module 1 78
7.1 Design of the BAM-1 78
7.1.1 Coding 79
7.1.2 Fitness Function 80
7.1.3 Genetic Operators 81
7.2 Evaluation of The BAM-1 84
7.2.1 Testing of the BAM-1 85
7.2.2 Conclusions 87
7.3 BAM-1 Object Definitions 88
7.3.1 Peak Class 88
7.3.2 Peak Lists Class 89
7.3.3 Alignment Class 90
7.3.4 Peak System Class 91
7.3.5 Peak-list Class 91

8.0 Backbone Assignment Module 2 93
8.1 Design of the BAM-2 93
8.1.1 Coding 93
8.1.2 Fitness Function 95
8.1.3 Genetic Operators 97
8.2 Evaluation of the BAM-2 99
8.2.1 Testing of the BAM-2 99
8.2.2 Conclusions 102
8.3 BAM-2 Object Definitions 104
8.3.1 Alignment and Peak Systems Classes 104
8.3.2 Peak System Class 104
8.3.3 Assignment Class 105

9.0 Future Work 106
9.1 Side Chain Assignment Module 106
9.1.1 Coding of the SCAM 106
9.1.2 Fitness Function of the SCAM 107
9.1.3 Genetic Operators of the SCAM 110
9.2 Future Work on 2D-SAM, 3D-SAM, BAM-1 and 112
BAM-2 112

9.3 User Interface 112
9.4 SCAM Object Definitions 113
9.4.1 Amino Acid Class 113
9.4.2 Spin System Class 115

Bibliography 116

CD-ROM .... attached to back cover

CD-ROM

The CD-ROM is formatted to the ISO-9660 standard with Rock-Ridge extensions.

The CD-ROM contains two files:

```
BG_Ailey_PhD_Thesis_Leicester_1997.tar.Z
BG_Ailey_PhD_Thesis_Leicester_1997.tar.gz
```

The files are both compressed tar archive files. The Z file is compressed using the standard UNIX compression utility compress. While the gz file was compressed using the gzip compression utility. To uncompress the files use the uncompress utility or the gunzip utility respectively. The tar archive file is extracted using the same tar utility that was used to compress it. Examples of extraction commands are listed below.

```
uncompress BG_Ailey_PhD_Thesis_Leicester_1997.tar.Z | tar -xvf -
```

```
gunzip BG_Ailey_PhD_Thesis_Leicester_1997.tar.gz | tar -xvf -
```

The tar file is 29 Mb and Contains the source code (Common Lisp) for each module, the results for each module and the experimental data each module was run on.

1.0 INTRODUCTION

Proteins are essential for life, even the simplest of living organisms[1] possess them. They form the most diverse class of biological macromolecules in both their structure and function. Proteins have three major functions: catalytic, mechanical and regulatory. In the form of enzymes[2] proteins catalyse nearly all biological reactions. They are also responsible for both the structure and movement of living organisms on both the intra and extra cellular level. These functions, and others, within an organism are regulated and controlled either directly or indirectly by other proteins. Although these are the main functions of proteins there are a number of others. The function of a protein is dependent upon its structure. Therefore the great diversity of protein function is dependent on an equal diversity of protein structure.

Proteins are formed by the combination of smaller biological molecules called amino acids. There are twenty naturally occurring amino acids and although each is different they all share certain properties (Figure 1.1). It is their shared property of being able to link to two other amino acids that enable them to form proteins; a protein is a chain of amino acids (Figure 1.2). The sequence and length of an amino acid chain determines the structure of a protein. The amino acid chain of a protein can be anything from a few tens to thousands of amino acids in length. The number of possible proteins formed for each chain length is 20 to the power of the chain length (Table 1.1). It is this astronomical number of possible amino acid sequences combined with the variety of amino acid biochemistry that gives proteins their structural diversity.

Proteins are classified as having 4 levels of structure. The primary structure of a protein is its amino acid sequence. The secondary structure of a protein is the local conformation of the amino acid chain. There are three main types of secondary structure: α-helix, β-sheet and random coil. In the α-helix secondary structure (red area of Figure 1.3 and Figure 1.4) the amino acid chain forms a helix. In the β-sheet form (yellow area of Figure 1.3 and Figure 1.4) two or more regions of the amino acid chain of the protein align with each other to form a plane or sheet. In the random coil (blue and white area of Figure 1.3 and Figure 1.4) there is effectively no secondary structure, the amino acid chain is in an irregular configuration. The sequence of amino acids in a protein determines its secondary structure and a protein can have all three types of

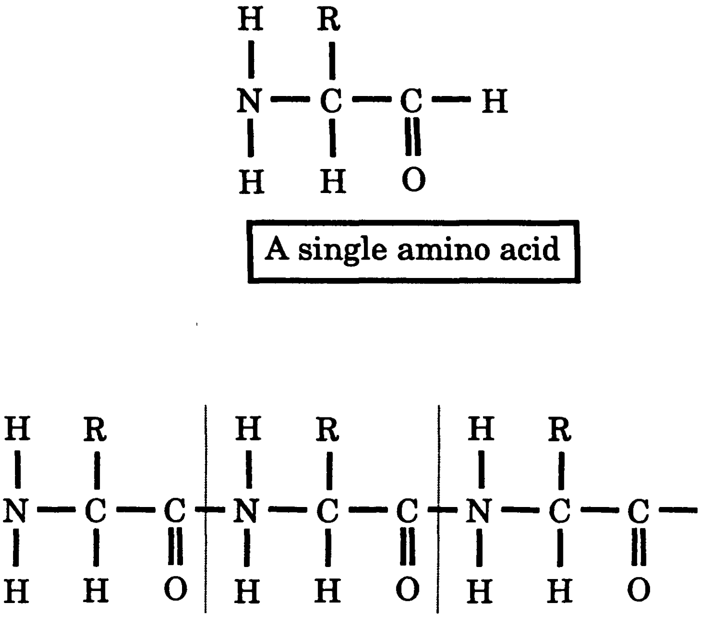
[1] Viruses are the simplest form of life; so simple it is debatable as to whether they are alive.

[2] Biological catalysts.

Table 1.1 Increase in the number of amino acid sequences with the increse in amino acid chain length.

Amino Acid Chain Length	Number of Possible Proteins
1	20
10	1.024×10^{13}
100	1.268×10^{130}

Figure 1. 1 Amino acid chain strucutre.



A single amino acid

Three amino acids linked together

All amino acids conform to this structure, they vary in the nature of the R.

Figure 1. 2 The amino acid chain of a protein

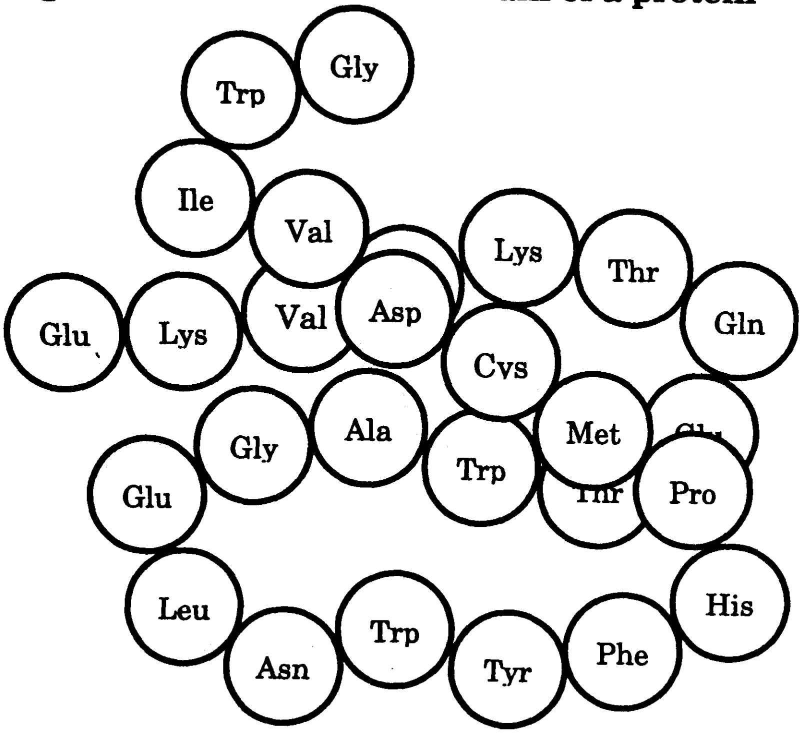


Figure 1.3 1BTA (PDB) α–Helical secondary strucutre

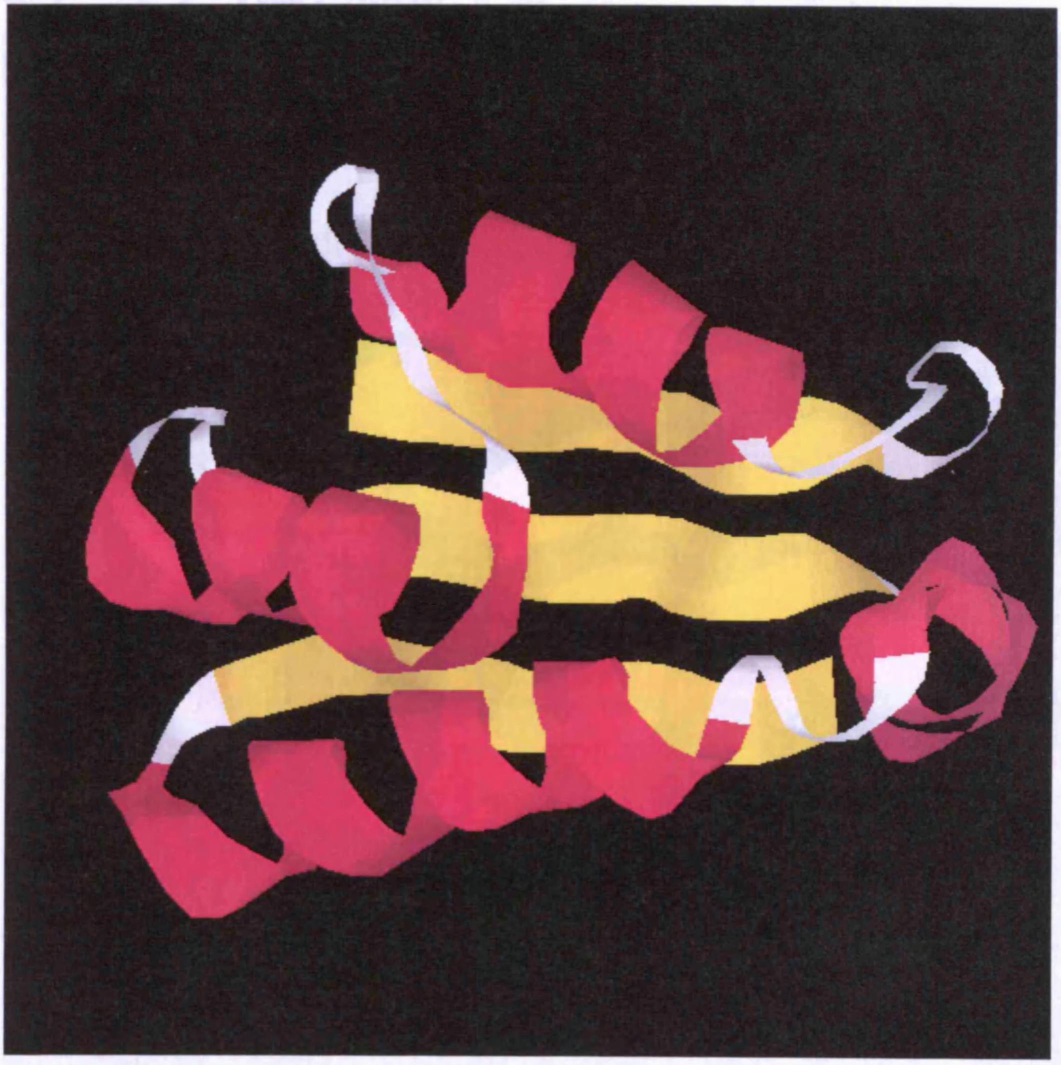
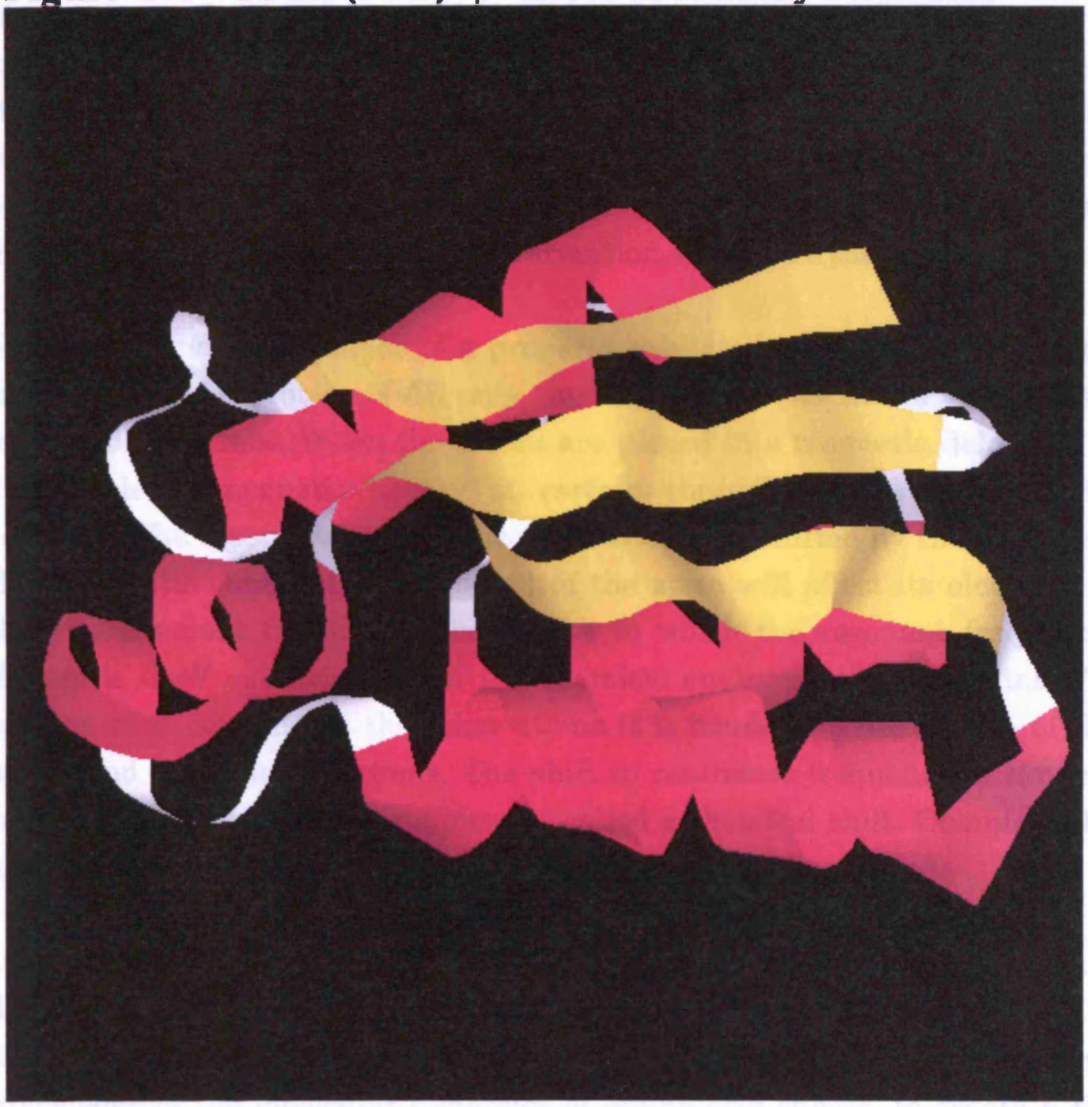


Figure 1.4 1BTA (PDB) β–Sheet secondary strucutre



secondary structure (Figure 1.3 and Figure 1.4). The tertiary structure of a protein is the global conformation of the amino acid chain. A protein has quaternary structure when it formed by two or more amino acid chains, the combination and interaction of the chains form its quaternary structure

Determining the primary structure can be done chemically but the secondary, tertiary and quaternary have to be determined using physical techniques. To determine the three dimensional structure of a protein at atomic resolution there are only two techniques: x-ray crystallography and NMR[3] (1). X-ray crystallography uses the x-ray diffraction pattern from the crystals of a protein to determine its three dimensional structure. In NMR the magnetic resonance of certain nuclei in the protein are used to generate NMR spectra; the information in the spectra is then used to determine its three dimensional structure. In the resolution of the protein structure and the size of protein to which the technique can be applied x-ray crystallography is the superior technique. NMR is superior in that it delivers the structure of the protein under near in vivo conditions and information about its dynamics.

Nuclear magnetic resonance is a property exhibited by certain atomic nuclei. In the context of protein NMR only the ^{1}H, ^{13}C and ^{15}N nuclei exhibit useful magnetic resonance. When the nuclei are placed in a magnetic field they will absorb electro-magnetic energy at certain characteristic frequencies, they resonate. The resonant frequency can be altered or shifted by the electrons of the atom. The chemical environment of the atom will affect its electrons and thus its resonant frequency. The degree to which the resonant frequency is shifted is itself characteristic of the chemical environment of the atom. This environment depends on the other atoms it is bonded to, the nature of those bonds and other nearby atoms. The shift in resonance frequency of a nucleus caused by its chemical environment is called a chemical shift. Chemical shifts are small and consequently measured in parts per million (ppm).

In protein NMR (2, 3, 4 & Chapter 2) sophisticated techniques are used to determine the resonant frequencies of most of the nuclei[4] in the protein and the distance between them. This information is then used, in combination with knowledge of the molecular dynamics of amino acid chains, to create a three dimensional structure of the protein. The process of determining the three dimensional structure of a protein has a number of different stages (Figure

[3] Nuclear Magnetic Resonance.

[4] The nuclei that exhibit useful magnetic resonance.

2.1). The most time consuming and least automated of these is the assignment phase in which the resonances observed in an NMR spectrum are assigned to one or more nuclei.

There have been a number of attempts to automate the assignment of protein NMR spectra. Most of these have concentrated on a rigid examination of NMR spectra to extract information to construct possible assignments followed by an exhaustive comparison of these possible assignments to choose the best (5, 6, 7, 8, 9, 10, 11 & 12). The applications generated had some success with very small easily assigned proteins, but problems arose when larger proteins were attempted. The data from larger proteins is nearly always more ambiguous (Section 2.2.2, 2.3.2, 2.4.2 & 2.5.2) and the number of possible assignments increases geometrically. The first problem will exacerbate the second. The rigidity of the programs could not cope with the ambiguity while the exhaustive approach was susceptible to combinatorial explosion. An example of a combinatorial explosion is the astronomical increase in the possible sequences of amino acids for a relatively modest increase in length of an amino acid chain (Table 1.1).

To solve the problems of ambiguity and combinatorial explosion two approaches have been tried: NMR and computational. The NMR approach has been to take advantage of new NMR techniques. These techniques use three dimensional and hetero nuclear experiments (Sections 2.3, 2.4 & 2.5). These techniques, used in combination, have significantly improved both the quality and the nature of the information contained in an NMR spectrum. In certain cases the change in the nature of the information has even reduced the number of possible assignments (Section 2.4). These techniques have allowed NMR to determine the structure of larger and more difficult proteins; the quality of the information, even with larger proteins, is an improvement over that supplied before.

The computational approach has been to use what are described as AI techniques, for example expert systems (13 & 14) and genetic algorithms (15). The expert systems techniques supply a flexibility that allows the programs to cope with the ambiguity inherent in NMR spectra, while genetic algorithms are powerful general purpose search algorithms. A search algorithm is one that 'searches' a number of possible solutions and selecting the best or one of the best solutions. Rather than an exhaustive search of the possible solutions a GA repeatedly samples them to find a good solution. This approach enables a GA to work effectively with very large numbers of possible solutions. Both these techniques seem to offer improved performance but still could not cope with proteins of any size.

By combining both expert systems and GAs with the improved information supplied in three dimensional heteronuclear NMR experiments it should be possible to develop useful tools for the assignment of protein NMR spectra. The strategy adopted in my work was to develop a program or module to take information from either a spectrum (Chapter 5) or type of spectrum (Chapter 7) and to assign the spectrum or spectra. If necessary the data is then passed onto another module. Although the new NMR techniques have reduced the amount of ambiguity they have not eliminated it. The remaining ambiguity combined with the diversity of protein structure means that a 100% correct automated assignment for most proteins is currently not achievable. The aim of the various modules is to perform the routine part of the assignment leaving the spectroscopist to concentrate on problems specific to the protein under investigation, spectroscopists have NMR techniques for dealing with such problems. The aim of all the modules is to produce a useful tool in the assignment of protein NMR spectra, as a rule of thumb 'a useful tool' is a module that gives an 80% correct assignment.

2.0 The Assignment of Protein NMR Spectra

The NMR spectra of a protein can be used to determine its three dimensional structure in solution (Figures 1.3 & 1.4). NMR spectroscopy is unique in being able to provide a detailed structure of small proteins in solution together with information about their dynamics. The ability to provide both structural and dynamic information under approximately physiological conditions makes NMR a valuable complement to X-ray crystallography. The determination of a protein structure by NMR involves a series of steps outlined in Figure 2.1. An important step is the assignment of resonances in the spectrum to individual groups in the protein.

2.1 Producing a Solution Structure

The following is a brief description of the protocol involved in obtaining a solution structure. The protocol is listed here to allow the assignment phase to be placed in its proper context (Figure 2.1).

2.1.1 Sample Preparation

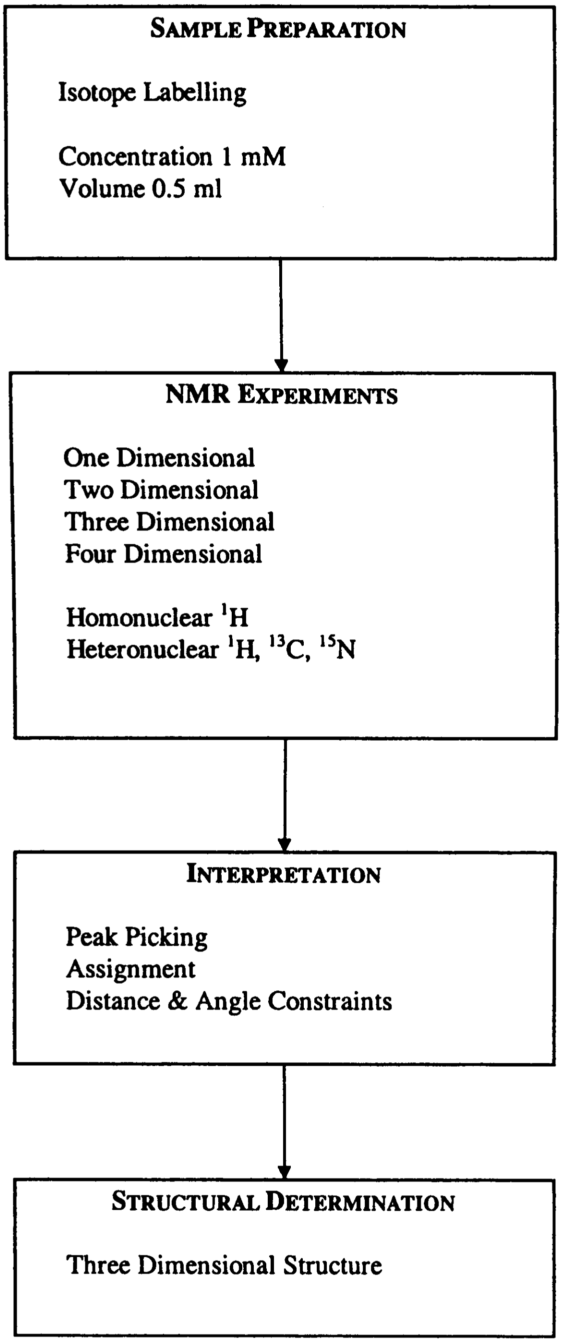
Sample preparation involves the isolation, purification and assessment of the protein. The protein is isolated from a natural source or from bacteria[5] that have been modified, using molecular biology techniques, to produce the protein. Once the protein has been isolated it is purified using various biochemical techniques. The solubility and stability of the purified protein is then assessed.

The protein must be soluble at the concentration and temperature required. If it is not soluble then the conditions must be altered: the temperature can be raised, the concentration lowered or the solution conditions changed. There are limits on these changes to the conditions. The NMR signal from the protein is proportional to its concentration for the same experimental conditions. The concentration can therefore only be lowered a certain amount. The higher the temperature the better resolved the spectra and the more soluble the protein. But raising the temperature can also reduce the stability of the protein.

The protein must be stable in the NMR tube i.e. it must retain its structure for a considerable period of time. A single NMR experiment can last for eight days. A series of experiments is often needed to gain enough information to produce a structure. Ideally the protein should exist in a single conformation. A protein can have two or more conformations which can generate more than

[5] Yeast and mammalians cells are also used.

Figure 2. 1 A protocol to produce an NMR structure of a protein.



one peak per nuclei. Different conformations can place nucleus in different environments and thus give the nuclei different chemical shifts.

For a number of experiments the isotope labelling of protein is necessary (Sections 2.3, 2.4 & 2.5). Only certain isotopes exhibit useful magnetic properties. Fortunately for biological NMR the ^{1}H isotope is one of them. Other commonly occurring isotopes in biological structures, ^{12}C, ^{14}N and ^{16}O, do not. There are carbon and nitrogen isotopes that do exhibit useful magnetic properties, ^{13}C and ^{15}N. In order to aid in the assignment process ^{12}C and ^{14}N can be replaced with ^{13}C and^{15}N. This substitution is called isotope labelling. A labelled protein is produced by using isotopically labelled nutrients in the growth medium of the cells that produce it. There are two types of labelling: selective and uniform. When selective labelling is required a labelled amino acid[6] is used in the cell culture medium. The labelled amino acid is then incorporated into the protein. In uniform labelling greater than 95% of the isotope or isotopes in the protein are substituted. All the references to isotope labelling in this thesis refer to uniform labelling unless otherwise stated.

Once all these factors have been considered the task of assigning spectra may begin. Generally the process of assigning protein NMR spectra can take anything from a few months to several years.

2.1.2 NMR Experiments

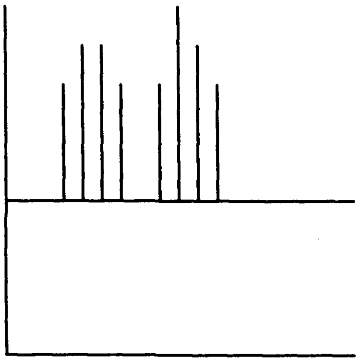
There are numerous NMR experiments that can be performed on proteins. There are several characteristics associated with each experiment that can be used to classify them. These characteristics include: the frequency dimensions of the spectra produced, the type of nuclei involved and the type of magnetisation transfer.

The most basic of NMR experiments is the one dimensional experiment (Figure 2.2) The spectrum is a resonance intensity plotted against frequency. This is informative for small molecules where only a small number of peaks are present. In larger molecules, such as proteins, there are problems. Large molecules produce a large number of peaks in their spectra. These peaks can overlap making assignment very difficult. To resolve this problem experiments were developed where a second frequency dimension was introduced. The extra dimension spaces the peaks out into another dimension, reducing overlap. This creates a two dimensional (2D) spectrum that resembles a

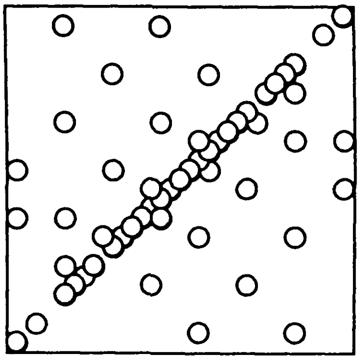
[6] labelled with either or both ^{15}N and ^{13}C.

Figure 2.2 1,2 & 3 Dimensional NMR Experiments.

1 Dimensional Spectrum



2 Dimensional Spectrum



3 Dimensional Spectrum

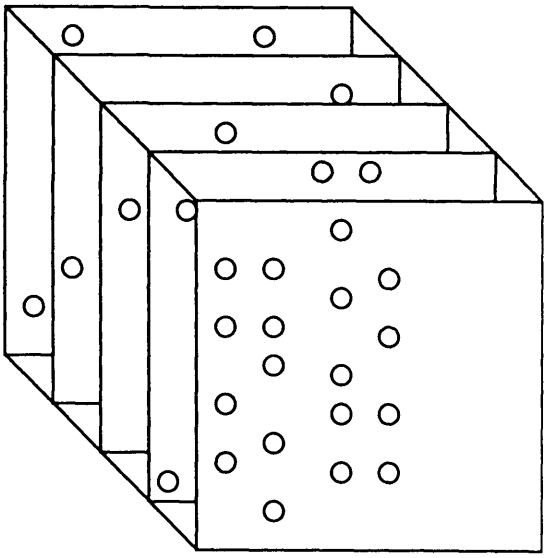


Figure 2.3 The relationship between the position of a cross peak and the resonant frequency of the nuclei that generated it.

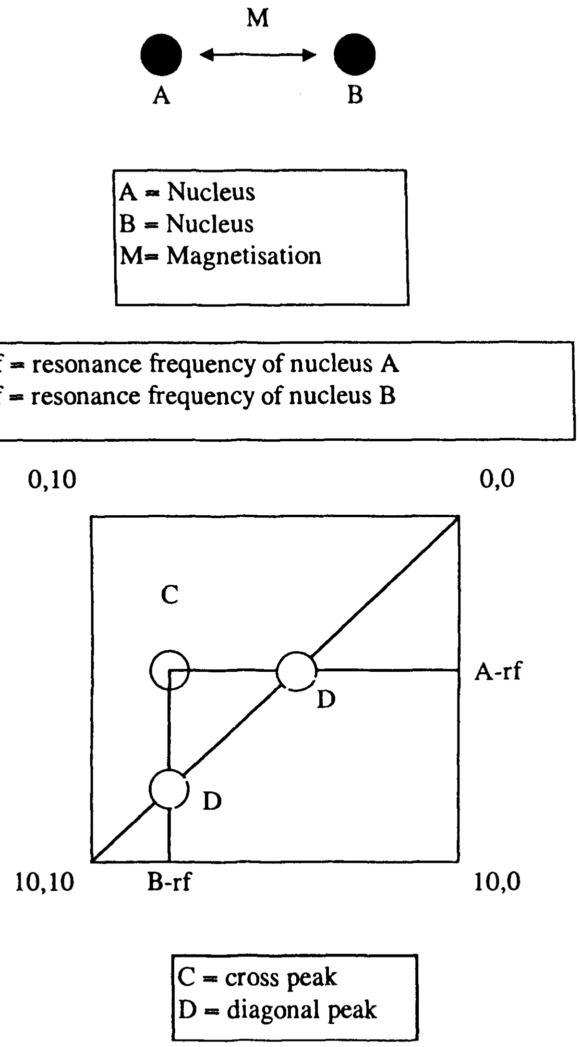


Figure 2. 4 A 'though bond' transfer of magnetisation.

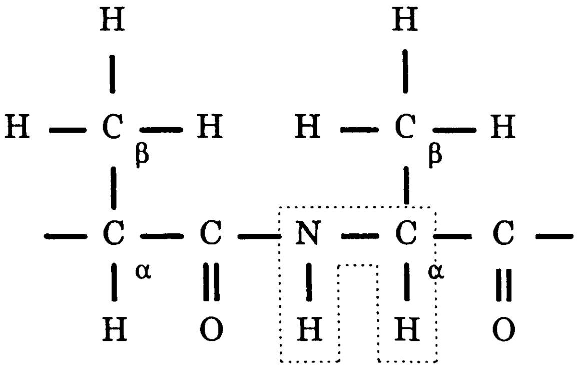
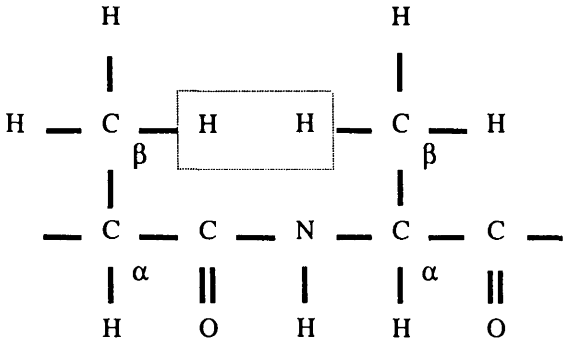


Figure 2. 5 A 'Through Space Transfer' of magnetisation.



contour map. The position of a peak is determined by its two frequencies[7] and the intensity determines its area (Figure 2.2). The 2D spectra also included information about the connections between peaks (Section 2.2). These connections are the most important feature of multi-dimensional NMR spectra and are crucial to the assignment. The two dimensional experiments were adequate for small proteins but for larger ones another increase in the number of dimensions was required. Three and four dimensional (3D and 4D) experiments increase the frequencies associated with each resonance or peak to three and four respectively. A 3D spectrum (Figure 2.2) is normally viewed as a series of 2D spectra in the form of a cube. The third dimension gives the position of a 2D spectrum within the cube. 4D spectra are viewed as a series of 3D spectra i.e. a series of cubes. Currently most NMR experiments on proteins are two and three dimensional.

NMR experiments can use several isotopes; naturally occurring ^{1}H and artificially introduced ^{13}C and ^{15}N. If the experiment uses only one isotope then the experiment is a homonuclear[8] experiment (Section 2.2). When more than one isotope is used the experiment is a heteronuclear one. Heteronuclear experiments have advantages over homonuclear experiments, they can be more efficient and additional isotopes can give additional information (Sections 2.3, 2.4 & 2.5).

The final method of characterisation of NMR experiments uses transfer of magnetisation. The way that a multidimensional experiment works is by transferring magnetisation between two or more nuclei and recording the resonant frequency of some or all the nuclei. The position of an NMR peak in a spectrum is determined by its frequency or frequencies (Figure 2.3). These frequencies are the resonance frequency of nuclei between which magnetisation has been transferred. The magnetisation can be transferred either 'through bond' (Figure 2.4) or 'through space' (Figure 2.5). Through bond refers to the type of experiment where the magnetisation is transferred through the electrons of the covalent chemical bonds that hold the nuclei together. A through space experiment transfers the magnetisation between

[7] The frequencies are referred to in an NMR spectrum as the chemical shift. In an isolated environment nuclei of the same type will resonate at the same frequency. However when in a molecule the chemical environment will cause the frequency to be shifted from what it would be in isolation, i.e. the chemical shift. The chemical shift of a nucleus is given relative to the resonant frequency of the nuclei in a reference molecule, and is measured in parts per million, ppm.

[8] In the context of protein NMR homonuclear nearly always means the 1H isotope is used

nuclei that are physically close and which may or may not be linked by a chemical bond. Through space experiments use the Nuclear Overhauser Effect (NOE) and the experiments are called Nuclear Overhauser Effect Spectroscopy or NOESY experiments.

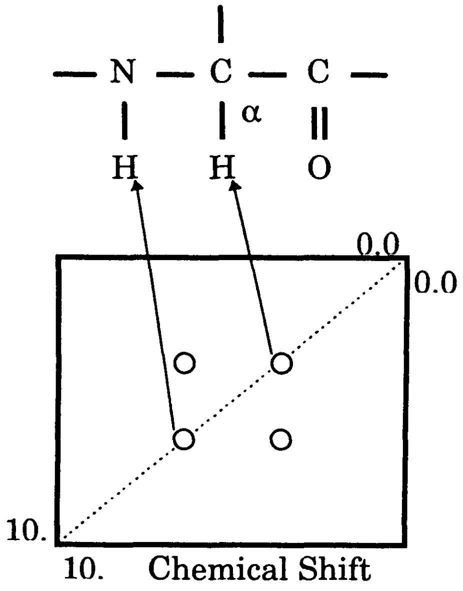
2.1.3 Interpretation of Protein NMR Spectra

Interpretation is the extraction from the spectra of a protein of all the information required to determine its solution structure. Interpretation can be divided into several stages: peak picking, assignment and the determination of angle and distance constraints.

Peak picking is the selection of genuine peaks from an NMR spectrum, determining the centre of each peak and calculating its intensity. Ideally this would be a simple process but there are problems due to noise peaks and peak overlap. Two characteristics are used to determine whether a peak is genuine or a noise peak: the peak shape and the peak intensity. In a 2D spectrum peaks have a characteristic appearance, any deviation from this is suspect. Secondly the more intense a peak the less likely it is to be a noise peak. By using these factors an experienced spectroscopist can identify genuine peaks with reasonable reliability. There can be several thousand peaks in an NMR spectrum. This can lead to some of the peaks occupying the same place, i.e. they overlap. Depending on the degree of overlap two peaks can appear to be one peak or when the existence of more than one peak can be distinguished determining their exact positions is difficult. Automated peak picking programs have been developed, with varying success. The approaches vary from algorithms that use the geometry and shape (16 &17) of a peak to neural networks that can be trained to recognise genuine peaks (18). Most automated peak picking programs are used as an initial filter. The peaks picked by the program are highlighted in the spectrum and then examined systematically by the spectroscopist to confirm that they are genuine peaks.

The assignment of an NMR spectrum is the assignment of a peak to a one or more nuclei in the protein (Figure 2.6); peaks are generated by one or more nuclei. Traditionally (19 & 20) the assignment is done in two stages: spin system identification and sequential assignment, alternatives have been suggested (21). A spin system in protein NMR is essentially an amino acid. Spin system identification is the identification of a characteristic pattern of peaks in an NMR spectrum as being generated by the nuclei of an amino acid or a type of amino acid, e.g. the pattern of peaks was generated by glycine nuclei. Sequential assignment is the assignment of all the spin systems

Figure 2. 6 Assignment of a protein NMR spectrum peaks to specific nuclei.



identified to a specific amino acids in the protein, e.g. the peaks were generated by the glycine at position twenty three. Once the spectra of the protein have been assigned the determination of the distance and angle constraints can begin.

The distance constraints state that a certain nucleus is within a certain range of distances from another nucleus. The distance constraints come from a through space NOESY experiment. Any ^1H nuclei less than ~ 5 Å[9] apart can produce a peak in such a spectrum. In a NOESY spectrum the intensity (I) of a peak is approximately inversely proportional to the distance (D) between the nuclei raised to the sixth power[10], see Equation 2.1. The intensity is only approximately proportional to the distance between the nuclei due to the inherent variability of NMR experiments.

$$I \propto \frac{1}{D^6}$$

Equation 2.1

The relative distance between any two nuclei linked by an NOE cross peak can be calculated using Equation 2.1. Certain nuclei in certain amino acids are always a certain distance apart, e.g. the nuclei in aromatic rings. Therefore absolute distances between nuclei can be calculated. An angle constraint states that the specified nuclei are a certain angle to each other.

2.1.4 Determination of Three Dimensional Structure of a Protein

The conversion of the distance constraints to a possible three dimensional structure is normally performed by a computer program. The program uses knowledge of the conformations an amino acid chain can assume in conjunction with the distance and angle constraints to generate possible structures. An amino acid chain can assume a myriad of different structures. A constraint, either distance or angle, reduces the number of possible structures. The larger the number of constraints the fewer possible structures there are. If enough constraints are used only one possible structure remains. That is not to say the time-average position of every nucleus is known, proteins are dynamic

[9] 1 Å = 1 x 10^{-10} m.

[10] The sixth power occurs only in isolated pairs of nuclei. When more than two nuclei are involved the effect of all the nuclei on each other must be calculated to generate distance to intensity relationship.

structures, in particular in regions of random coiling[11].

The process of obtaining a solution structure is not the simple linear process outlined here, rather it is an iterative process. The results from one step are often fed back into a previous step. Having placed the assignment of protein spectra in context the next sections describe the assignment process in more detail. The NMR experiments described in the subsequent sections are those used by the various GA modules (Chapters 5, 6, 7, 8 & 9).

2.2 The Use of Two Dimensional Homonuclear ^{1}H NOESY Spectra to perform Sequential Assignment

In the traditional approach (22) NOESY experiments are used to perform the sequential assignment, for example a pattern of peaks are assigned to a glycine amino acid. In sequential spin system assignment spin systems are assigned to specific residues in the amino acid sequence, for example a pattern of peaks are assigned to glycine 23. Through space NOESY experiments are used in sequential assignment because no through bond ^{1}H-^{1}H experiment can link spin systems[12]. The NOE through space transfer of magnetisation can transfer magnetisation between nuclei of adjacent spin system. Whatever the local conformation there will always be nuclei within 5Å of each other, the maximum distance that the NOE will transfer magnetisation. The NOESY spectrum or spectra are used to identify residues that are adjacent to each other. This is repeated until there are sequences of spin systems. When unique sequences appear the spin systems of the sequence can be assigned to the relevant positions. If you have a sequence of spin systems, e.g. Gly-Val-Ser-Leu that occurs only once in the amino acid sequence, e.g. residues 14 to 17, then the sequence is assigned Gly 14-Val 15- Ser 16-Leu 17 (Figure 2.7).

2.2.1 Assignment Rules for NOESY Spectra

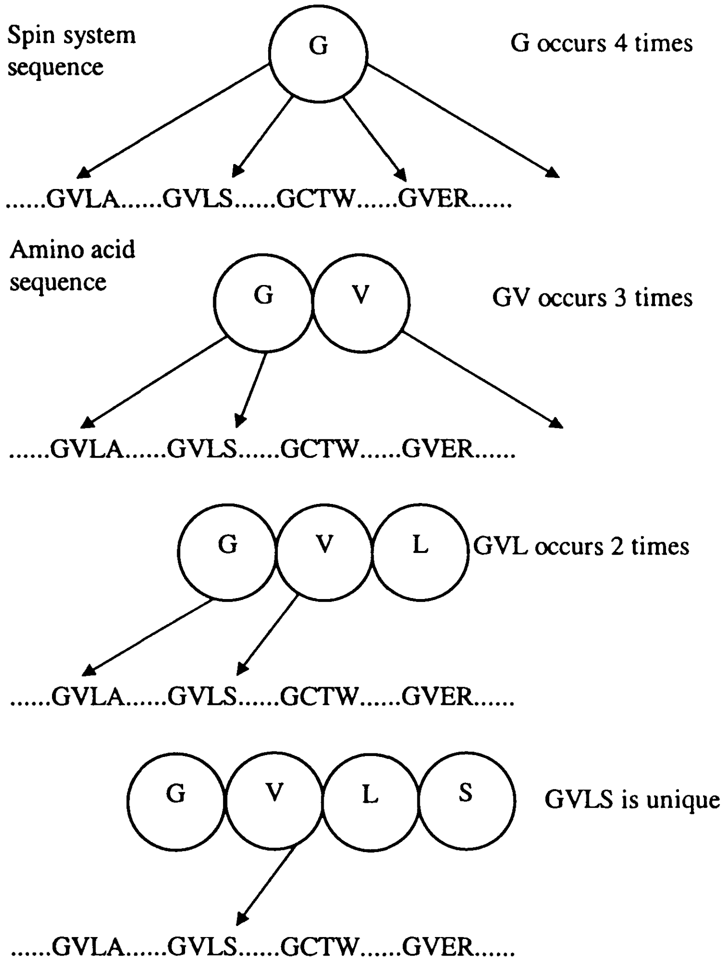
The spectrum is two dimensional (Figure 2.2). There are two types of peaks in the spectrum: diagonal peaks and cross peaks. Diagonal peaks are peaks that have the same chemical shift in both dimensions and are generated by one nucleus. Cross peaks have different chemical shifts and are generated by an NOE interaction between two nuclei. The chemical shifts of a cross peak come from the chemical shifts of the two interacting nuclei.

To find the sequential interactions of a spin system using a NOESY spectrum

[11] One of the three types of secondary structure; the other two are α-helix and β-sheet.

[12] The transfer does take place but is not detectable.

Figure 2. 7 Sequential assignment of a sequence of spin systems


Spin system
sequence
G
G occurs 4 times
......GVLA......GVLS......GCTW......GVER......
Amino acid
sequence
G
V
GV occurs 3 times
......GVLA......GVLS......GCTW......GVER......
G
V
L
GVL occurs 2 times
......GVLA......GVLS......GCTW......GVER......
G
V
L
S
GVLS is unique
......GVLA......GVLS......GCTW......GVER......

the NH, $C_{\alpha}H$ and $C_{\beta}H$ nuclei are used. The diagonal peak generated by the NH nucleus of an amino acid is selected. The chemical shift of the peak, and therefore its position, is already known from the spin system assignment. Any cross peak that aligns with the NH peak of the amino acid in either the D1 or D2 dimension indicates a potential link to another amino acid or to another nucleus within the amino acid, an intra-residue cross peak. Some of the cross peaks should align with the diagonal peaks generated by the NH, $C_{\alpha}H$ or $C_{\beta}H$ nuclei of the preceding amino acid. There should be two or three cross peaks linking the amino acid of interest, amino acid i, and the succeeding amino acid, i +1. The links are $C_{\alpha}H_{(i)}$ to $NH_{(i\ +1)}$ and $C_{\beta}H_{(i)}$ to $NH_{(i\ +1)}$ (Figure 2.8). Depending on the type of the i +1 amino acid there can be a second $C_{\alpha}H_{(i)}$ to $NH_{(i\ +1)}$ or $C_{\beta}H_{(i)}$ to $NH_{(i\ +1)}$ cross peak[13]. The NH, $C_{\alpha}H$ and $C_{\beta}H$ diagonal peaks occur in characteristic regions of the spectrum. This allows the determination of possible $C_{\alpha}H$ and $C_{\beta}H$ chemical shifts of the i amino acid. Having found possible $C_{\alpha}H$ and $C_{\beta}H$ chemical shifts of the i amino acid the spin system assignment is searched for an amino acid that has the appropriate $C_{\alpha}H$ and $C_{\beta}H$ chemical shifts. When a matching amino acid is found it is identified as the i amino acid. If no match is found then an alternate set of C_{α} H and $C_{\beta}H$ chemical shifts are tried. The secondary structure will affect the distances between nuclei of different amino acids. The change in distance will cause a corresponding change in the intensity of NOE cross peaks linking nuclei of different amino acids. The variation in the distances between nuclei can be seen in Table 2.1. The data in the table shows that whatever the secondary structure of the protein is there are at least two pairs of i and i+1 nuclei with 5Å of each other.

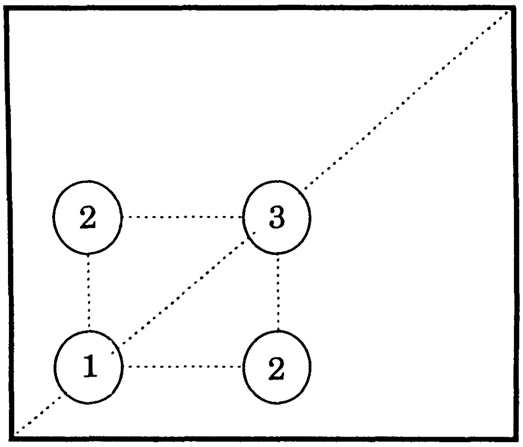
2.2.2 Assignment Ambiguities

The assignment rules outlined above are simple. The sequential assignment of a protein using its NOESY spectrum should therefore in principle be an easy process. In practice the process can be difficult. The cause of this difficulty is the ambiguity of protein NMR spectra. The ambiguity is caused by a number of factors: overlapping peaks, noise peaks, missing peaks, water line, chemical shift differences between spectra and non-sequential interactions.

- There can be several thousand peaks in a NOESY spectrum. The peaks are also distributed unevenly in the spectrum, i.e. NH $C_{\alpha}H$ cross peaks will all

[13] There is a second α cross peak in the residue is a glycine and there is a second β when the residue is a: serine, cysteine, aspartate, asparagine, leucine, lysine, arginine, glutamate, glutamine, methionine, histidine, phenylalanine, tyrosine and tryptophan.

Figure 2. 8 Sequential assignment using NOE cross peaks.



Peak 1 =-$NH_{(i+ i)}$ Diagonal Peak
Peak 3 = $C_{\alpha}H_{(i)}$ Diagonal Peak
Peak 2 = $C_{\alpha}H_{(i)}$ - $NH_{(i + 1)}$ Cross Peak

Table 2. 1 Bond lengths in various types of secondary structure.

Parameter	α-Helix	310-Helix	β-Anti parallel	β-parallel
αN (i, i)	2.6	2.6	2.8	2.8
αN (i, i+1)	3.5	3.4	2.2	2.2
αN (i, i+2)	4.4	3.8		
αN (i, i+3)	3.4	3.3		
αN (i, i+4)	4.2	>4.5		
NN (i, i+1)	2.8	2.6		
NN (i, i+2)	4.2	4.1		
βN (i, i+1)	2.5-4.1	2.9-4.4	3.2-4.5	3.7-4.7
αβ (i, i+3)	2.5-4.1	3.1-5.1		

be located in a characteristic region of the spectrum. The large number of peaks and their uneven distribution causes peaks to overlap. Overlapping peaks can have a number of effects depending on the degree of overlap. In the worst case two or more overlapping peaks can be perceived as only one peak. The effect of this is that one or more peaks appear to missing. The combined peak is not located precisely where any of its constituent peaks would be. Peak overlap can still cause difficulties even when the individual overlapping peaks can be seen. The overlap can disturb the shape of the peaks causing the apparent centre of the peaks to be altered. The change in the peaks centres and thus chemical shift causes ambiguities when the peaks are aligned (see 2.2.1).

- Noise peaks are experimental artefacts that occur in an NMR spectrum as additional peaks. These peaks can cause ambiguities by apparently linking resonances that are not linked.

- Missing peaks are peaks that in theory should appear in a spectrum but do not. This causes ambiguity in that spin systems that should be linked by an NOE peak are not.

- When the solvent used in the sample is H_2O an intense line appears in the NMR spectrum. The line is generated by the signal from ^1H nuclei in the H_2O. Any peaks on or near the intense line, the "water line", will be obscured by it.

- The chemical shift of a nucleus may vary slightly from spectrum to spectrum. The chemical shifts will commonly vary by about +/- 0.03 ppm, although it can exceed that value. The chemical shifts of the spin system assignment are obtained from different spectra to the NOE spectra. There can be a slight difference between a chemical shift of a nucleus recorded in the spin system assignment and the one found in the NOE spectrum. Put simply the cross peak linking two spin systems will not be in exactly the place anticipated.

- A non-sequential interaction is the interaction between an amino acid and one that is not adjacent to it yet is close to it in space. These interactions give the distance constraints used in the calculation of possible structures (see 2.1.3) but can also cause ambiguity during sequential assignment as the sequential and non-sequential links of an amino acid can easily be confused. There are certain specific non-sequential interactions that will appear depending on the secondary structure of the protein, see Table 2.1. When the secondary structure is α-helical the distance between the i and i + 3 residues and the i and i + 1 residues is approximately the same; giving cross peaks of equal intensity which can lead to ambiguity. When the secondary structure is β-sheet and the strands are anti-parallel the inter-strand distances between residues are approximately the same as the distance between the i and i + 1 residues. Distinguishing between the two can be difficult if it is possible at all. This is another source of ambiguity.

All the above factors combine to turn sequential assignment from a simple process to a complex one. To deal with these problems the spectroscopist will often run several NOESY experiments with varying conditions. Under different conditions noise peaks may disappear and missing peaks may appear. The chemical shifts of the peaks may move; separating overlapped peaks and moving peaks from under the water line. To allow for ambiguity a possible assignment is considered until an inconsistency is revealed as the assignment proceeds. When a possible assignment has an inconsistency the spectroscopist backtracks to the point of conflict and tries a different assignment until a consistent assignment is produced. This does not mean the assignment is complete: there can be adjacent amino acids in the assignment that are not linked by NOE peaks, gaps in the assignment and spin systems and NOE peaks not assigned. The completeness of an assignment is dependent upon the quality of the spectra which in turn is primarily dependent of the size and secondary structure of the protein (see 2.1.3).

2.3 The Use of Three Dimensional Heteronuclear ^{15}N NOESY Spectra to Perform Sequential Assignment

The interpretation of the spectra of larger proteins (23)· more than 100 amino acids, or those with large regions of α-helical secondary structure can be extremely difficult if not impossible. With larger proteins the increase in the number of peaks and the reduction in resolution can cause problems. The reduction in resolution is due to line broadening. Proteins with large regions of α-helical secondary structure will have reduced dispersion in the NH-CαH region of their spectra which causes increased peak overlap in this critical region. To sequentially assign these proteins 3D heteronuclear NOESY experiments are used. The 3D approach adds an extra spectral dimension which reduces peak overlap. The transfer of magnetisation between ^{15}N nuclei and ^{1}H nuclei is more efficient than between ^{1}H nuclei. The efficiency of magnetisation transfer, by scalar interactions, between nuclei is dependent on the coupling constant of the two nuclei. The $J_{NH\text{-}C\alpha H}$ coupling constant is ~ 3-10

Hz. The $J_{N\text{-}H}$ coupling constant by comparison is ~ 94 Hz, an order magnitude greater. This improves the signal to noise ratio. The 3D heteronuclear NOESY spectra are easier to interpret; they can each be considered to be a series of simplified 2D homonuclear NOESY spectra each at different ^{15}N chemical shifts.

2.3.1 Assignment Rules

The peaks in the spectra have three chemical shifts (Figure 2.9): one ^{15}N and two ^{1}H. The ^{15}N and one of the ^{1}H chemical shifts come from the N and NH of the amino acid. The other ^{1}H shift can come from a ^{1}H nucleus in either the same or a different amino acid. The peaks of an amino acid will have the same ^{15}NH and N^{1}H chemical shifts and therefore appear as a line of peaks in a spectrum. To perform the sequential assignment an amino acid is selected. The relevant line of peaks is found. The ^{15}NH and N^{1}H will be known from the spin system assignment. The ^{1}H chemical shifts of the intra-residue cross peaks will be known from the spin system assignment. Using this information the intra and inter residue peaks can be identified. Another spin system is then searched for that has an intra-residue peak with the same ^{1}H chemical shift as one of the inter-residue peaks (Figure 2.10). Two amino acids with matching inter-residue and intra-residue peaks are linked

2.3.2 Assignment Ambiguities

The assignment ambiguities are the same as for the 2D NOESY experiment with certain exceptions. The ambiguity caused by overlapping peaks, noise peaks, missing peaks, chemical shift differences between spectra and non-sequential interactions still exist. The water line no longer presents any difficulties and diagonal peaks can now be distinguished.

The additional dimension spaces the peaks out and reduces peak overlap. For overlap to occur in a 3D ^{15}N NOESY spectrum the peaks must have the same three chemical shifts. The pattern of peaks within a spin system makes the sequential assignment process much easier, i.e. the peaks have the same ^{15}NH and N^{1}H chemical shifts forming a line in the spectrum. However the increase in expense and the time taken to perform 3D experiments means they are primarily used for larger and/or difficult proteins. As NMR spectrometers develop larger and larger proteins are studied. To complete the sequential assignment of these very large proteins even the improvement of 3D ^{15}N NOESY is not enough. Other experiments using a completely different approach must be used.

Figure 2. 9 Nuclei to chemical shift relationship in a 3D ^{15}N NOESY spectrum.

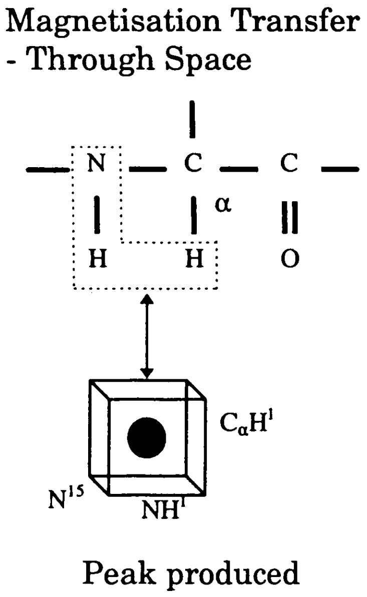
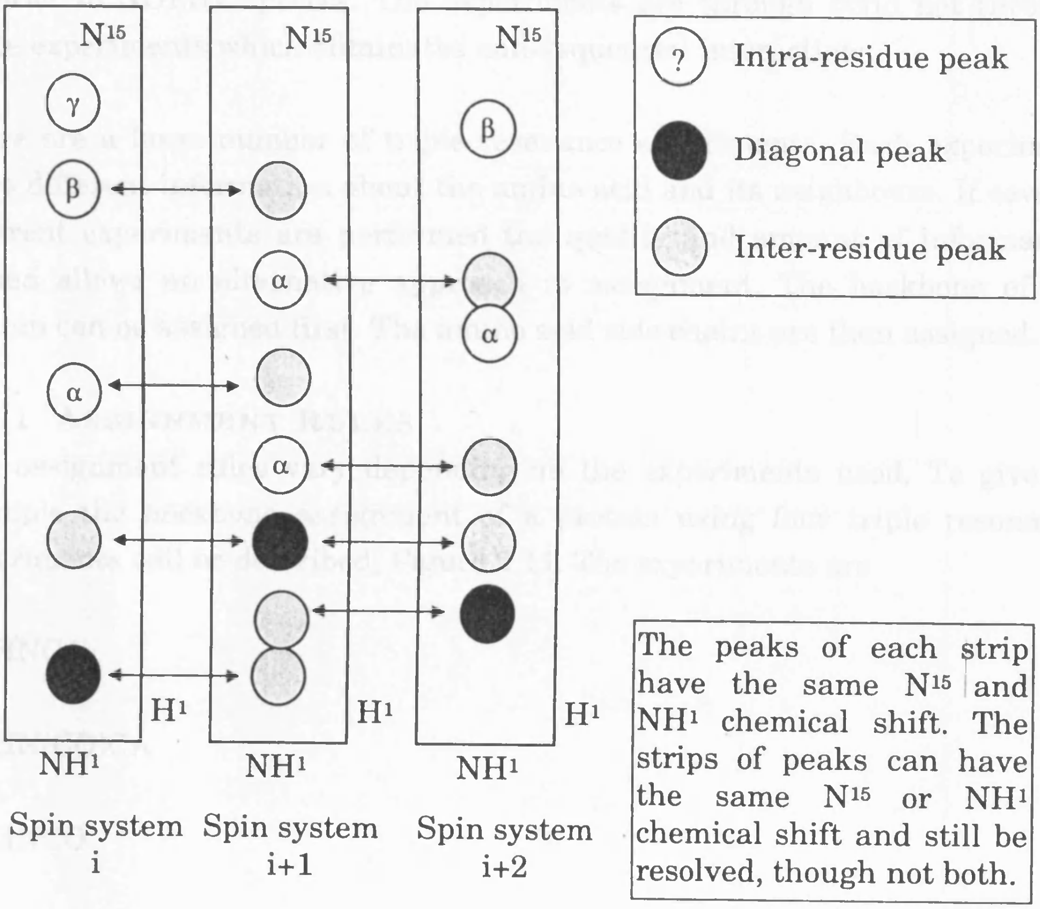


Figure 2. 10 Spin system linkage using a 3D ^{15}N NOESY spectrum.



The peaks of each strip have the same N^{15} and NH^{1} chemical shift. The strips of peaks can have the same N^{15} or NH^{1} chemical shift and still be resolved, though not both.

2.4 The Use of Triple Resonance Spectroscopy to Perform Backbone Assignment

Triple resonance means the use of three different types of nuclei. In protein backbone assignment the ^{1}H, ^{13}C and ^{15}N nuclei are used. Protein backbone triple resonance experiments are through bond experiments. The experiments enable the magnetisation transferred from one amino acid to another, through bond, to be observed. The spectrum produced by a protein backbone triple resonance experiment is relatively simple. There are one or two peaks in the spectrum for each amino acid in the protein. The peaks will have some of the chemical shifts of one amino acid and some of its neighbours. The simplicity of the spectra and the quality of the information obtained from them make them superior to NOESY spectra. The experiments are through bond not through space experiments which eliminates non-sequential interactions.

There are a large number of triple resonance experiments. Each experiment gives different information about the amino acid and its neighbours. If several different experiments are performed the quality and amount of information gained allows an alternative approach to assignment. The backbone of the protein can be assigned first. The amino acid side chains are then assigned.

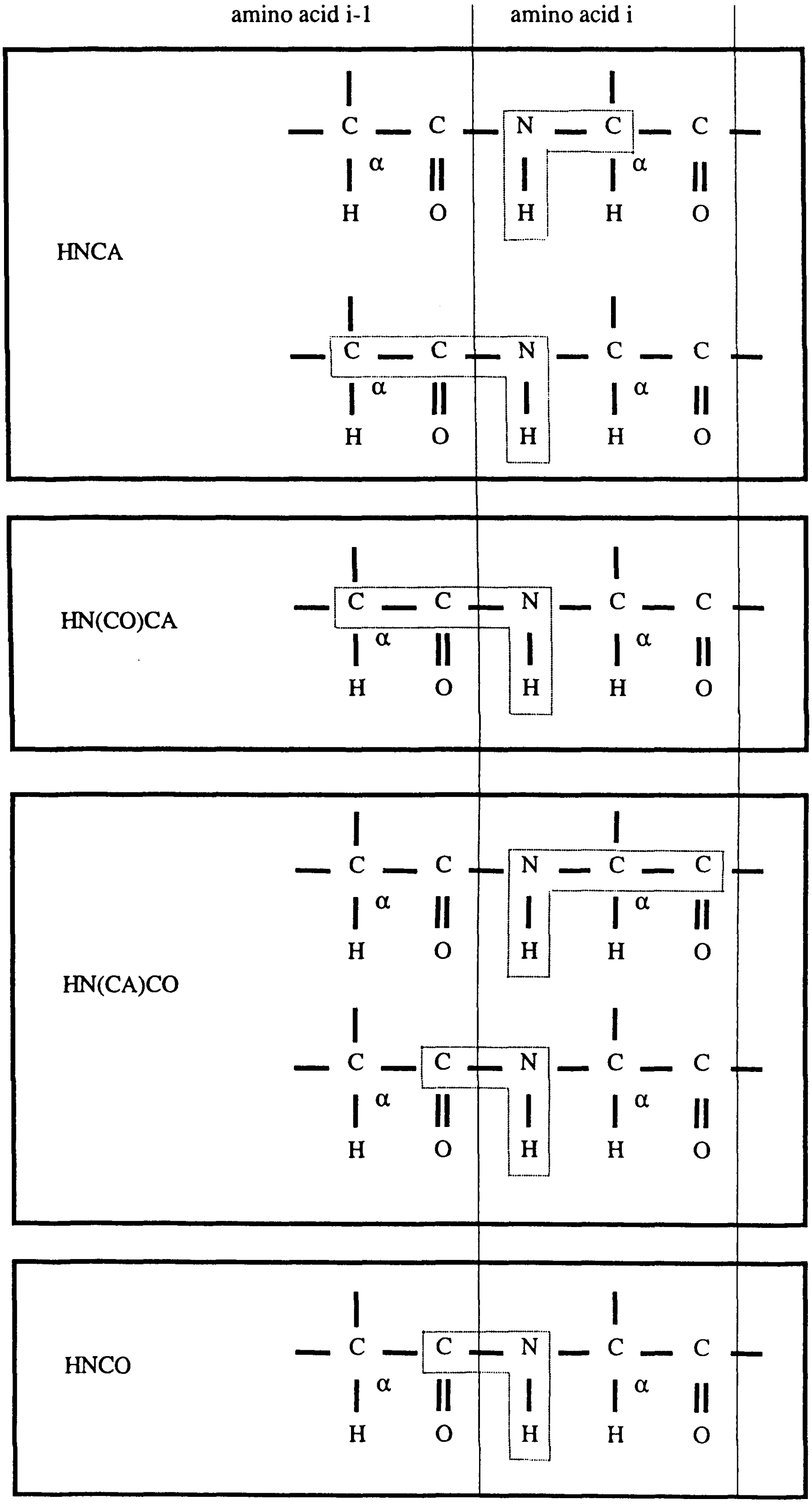
2.4.1 Assignment Rules

The assignment rules vary depending on the experiments used. To give an example the backbone assignment of a protein using four triple resonance experiments will be described, Figure 2.11. The experiments are

- HNCA
- HN(CO)CA
- HNCO
- HN(CA)CO

The names are descriptive of the transfer of magnetisation in the experiment and the information it reveals. The HN(CO)CA experiment transfers magnetisation from the N^{1}H(i) to the ^{15}N(i) to the ^{13}CO(i - 1) to the ^{13}Cα(i - 1) (Figure 2.11). The experiment gives a spectrum with a peak for each amino acid that has the chemical shifts N^{1}H(i), ^{15}N(i) and ^{13}Cα(i - 1). The ^{13}CO(i - 1) chemical shift does not appear in the spectrum; this is indicated by the brackets around the CO in the experiment name. The HNCA and HN(CA)CO

Figure 2. 11 Transfer of magnetisation in four triple resonance experiments.



experiments produce two peaks per amino acid. The main peaks are the intra-residue peaks described by the experiments name and are listed first in Figure 2.12. The secondary peaks are inter-residue peaks, more likely to be missing peaks and their intensity is often significantly less than the main peaks. The secondary peaks of the HNCA and HN(CA)CO experiments are effectively duplicates of the peaks of the HN(CO)CA and HNCO experiments respectively.

The assignment process starts by finding all the peaks from the same amino acid. This is done by finding all the peaks that have the same $N^1H(i)$ and $^{15}N(i)$ chemical shifts. The $^{13}C\alpha$ and ^{13}CO chemical shift of the amino acid, amino acid i, will be the ^{13}C chemical shift of the main peaks of the HNCA and HN(CA)CO experiments respectively. The $^{13}C\alpha$ chemical shift of the preceding amino acid, amino acid i - 1, will be the ^{13}C chemical shift on the peaks of the HN(CO)CA experiment and the secondary peaks of the HNCA experiment. The ^{13}CO chemical shift of the preceding amino acid, amino acid i - 1, will be the ^{13}C chemical shift of the peaks of the HNCO and the secondary peaks of the HN(CA)CO experiments. Knowing the $^{13}C\alpha$ and ^{13}CO chemical shift of the amino acid and the $^{13}C\alpha$ and ^{13}CO chemical shift of the preceding amino acid it is possible to link adjacent amino acids. When this process is carried out systematically it is possible to construct a backbone assignment or a sequential assignment.

2.4.2 Assignment Ambiguities

There are the standard assignment ambiguities due to: overlapping peaks, missing peaks, noise peaks and variations in chemical shifts between spectra. The peak overlap will be reduced due to the simplicity of the spectra. There is an ambiguity in the assignment due to the two different peaks produced in the HNCA and HN(CA)CO spectra. When there is a full complement of peaks for each amino acid then there is no ambiguity. When the HN(CO)CA peak of an amino acid is missing from the spectrum then which of the two peaks generated by the amino acid in the HNCA spectrum is inter-residue and which is intra-residue is ambiguous. The same is true for the HNCO and HN(CA)CO experiments. The intensity of the secondary peaks is often less than that of the main peaks but this is not always true.

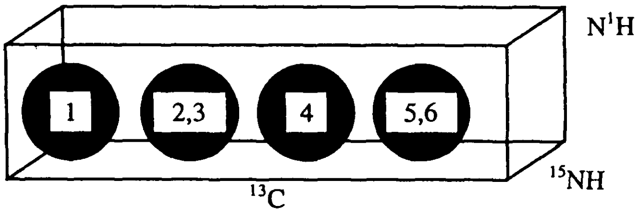
2.5 Performing Spin System Assignment Using HCCH ^{13}C TOCSY and COSY Spectroscopy

HCCH ^{13}C total correlation spectroscopy (TOCSY) and correlation spectroscopy (COSY) are through bond 3D heteronuclear experiments. The protein must be ^{13}C isotope labelled. The magnetisation is transferred from a ^1H to a ^{13}C to

Figure 2. 12 Alignment of the peaks to form a peak system from four triple resonance spectra..

An ideal spin system will give six peaks. Each peak will have three chemical shifts.

HNCA spectra gives peaks 1 and 2.
NH(CO)CA spectra gives peak 3.
NH(CA)CO spectra gives peaks 4 and 5.
HNCO spectra gives peak 6.



Positions of the six peaks generated by an amino acid and its neighbours if the four spectra were combined, note that the six peaks all share the same N^{1}H and ^{15}NH chemical shifts.

Alignments of Peaks

Peaks 1-6 align N^{1}H and ^{15}N.
Peaks 2-3 align ^{13}Ca
Peaks 5-6 align ^{13}CO
Intensity peak 1 > intensity of peak 2, probably.
Intensity peak 4 > intensity of peak 5, probably.

another ^{13}C and to another ^{1}H (Figure 2.13). The peaks in either experiment will have peaks with three dimensions, two ^{1}H and one ^{13}C. There will be a diagonal peak for each ^{1}H, a peak where the two ^{1}H chemical shifts are the same. There are cross peaks that represent connections between two protons. A cross peak will have the same ^{13}C chemical shift and ^{1}H chemical shift as one diagonal peak and the same ^{1}H shift as the other diagonal peak. In a COSY experiment the detectable magnetisation transfer between ^{1}H and ^{13}C nuclei will only extend over three chemical bonds. Only the neighbour nuclei will give peaks in the spectrum. In a TOCSY experiment the magnetisation transfer between ^{1}H and ^{13}C nuclei will extend over the whole amino acid side chain. Each diagonal peak in a spin system will have a cross peak connection to all the other diagonal peaks in the spin system.

2.5.1 Assignment Rules

There are two steps to performing a spin system assignment: constructing spin systems and then identifying them. The first step is to find a group of peaks that align with each other. The pattern of peaks in a spectrum is that of a series of strips. For each of these strips of peaks the ^{13}C and one of the ^{1}H chemical shifts will be the same. The strips are combined to perform potential spin systems by aligning a cross peak of one strip with the diagonal peak of another (Figure 2.14). Having constructed a potential spin system there is the task of identifying it. The pattern of peaks in the spin system can be unique or it can belong to one of a group of amino acids. The peaks of a spin system typically occur in characteristic regions of the spectrum, e.g. the $^{13}C\alpha$ of glycine will have a chemical shift of ~ -24 ppm[14] while the $^{13}C\alpha$ of serine is ~ -11 ppm. By a combination of characteristic peak patterns and chemical shifts it should be possible to identify 18 out of the 20 amino acids.

2.5.2 Assignment Ambiguities

There are the standard assignment ambiguities. An additional ambiguity is that the chemical shifts that are typical of an amino acid are often not unique to it. Other problems arise when the chemical shift of a nucleus is not at the characteristic position; the nucleus is in an unusual environment and is therefore shifted to an unusual degree or direction. A combination of these ambiguities can mean that it is impossible to definitely identify all the spin systems and it is only once the sequential assignment is done a definitive assignment is produced.

[14] The reference material is TMS.

Figure 2. 13 Magnetisation Transfer in a 3D HCCH ^{13}C NMR experiment.

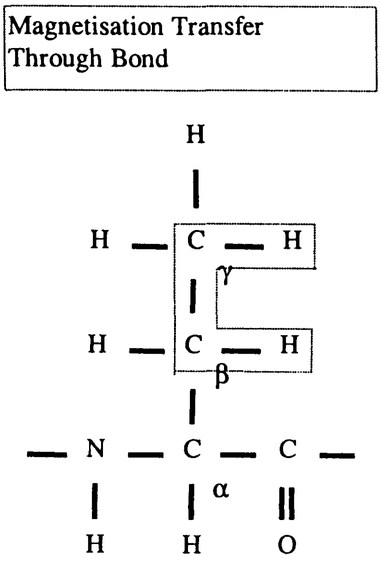
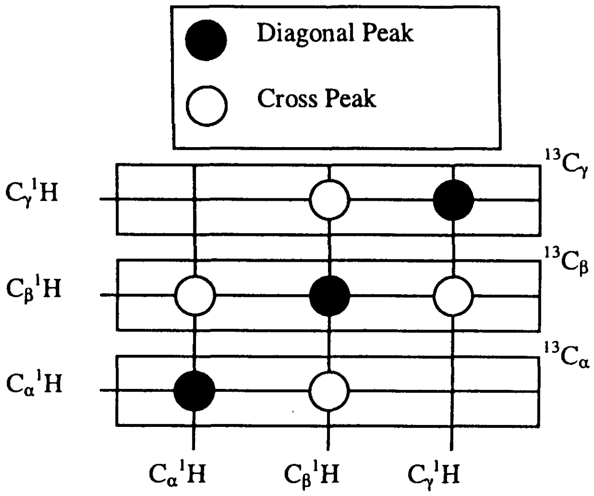


Figure 2. 14 Constructing a spin system in a 3D HCCH ^{13}C spectrum.



As stated the processes outlined in this chapter have been described in a sequential fashion. In practice the process is iterative; often different stages will be partially or completely repeated using the information gained from one of the subsequent stages. The assignment stage is completely dependent upon the quality of the spectra obtained. Good spectra result in complete assignment of the protein while poor spectra can result in partial or incorrect assignment.

3.0 Evolutionary Strategy

An evolutionary strategy is one that uses the principles of biological evolution. A computer program that uses such a strategy is called a genetic algorithm (GA) (24, 25, 26, 27, 28 & 29). GAs are search or optimisation algorithms. These algorithms search among a large number of alternatives for an optimum or near optimum solution. A GA works by creating a group of artificial structures that represent possible solutions to a problem. Four procedures are then performed on the structures.

1. The structures are randomly modified.

2. The possible solution each structure represents is determined.

3. The quality of the solutions are evaluated.

4. A new group of structures is created. The chance of a structure being in the new group is proportional to the quality of the solution it represents.

The procedures are repeated until some criterion is met: either a set number of cycles through the processes one to four or a solution achieving a predetermined quality. The random modification creates new structures. The new structures can represent either better or worse solutions than the original structure. The quality of the solution each structure represents is determined. The quality of each structure determines its chances of being in the new group of structures. By making quality proportional to survival during the running of the GA the good structures increase in number while the bad structures decrease in number. The process evolves better solutions to the problem.

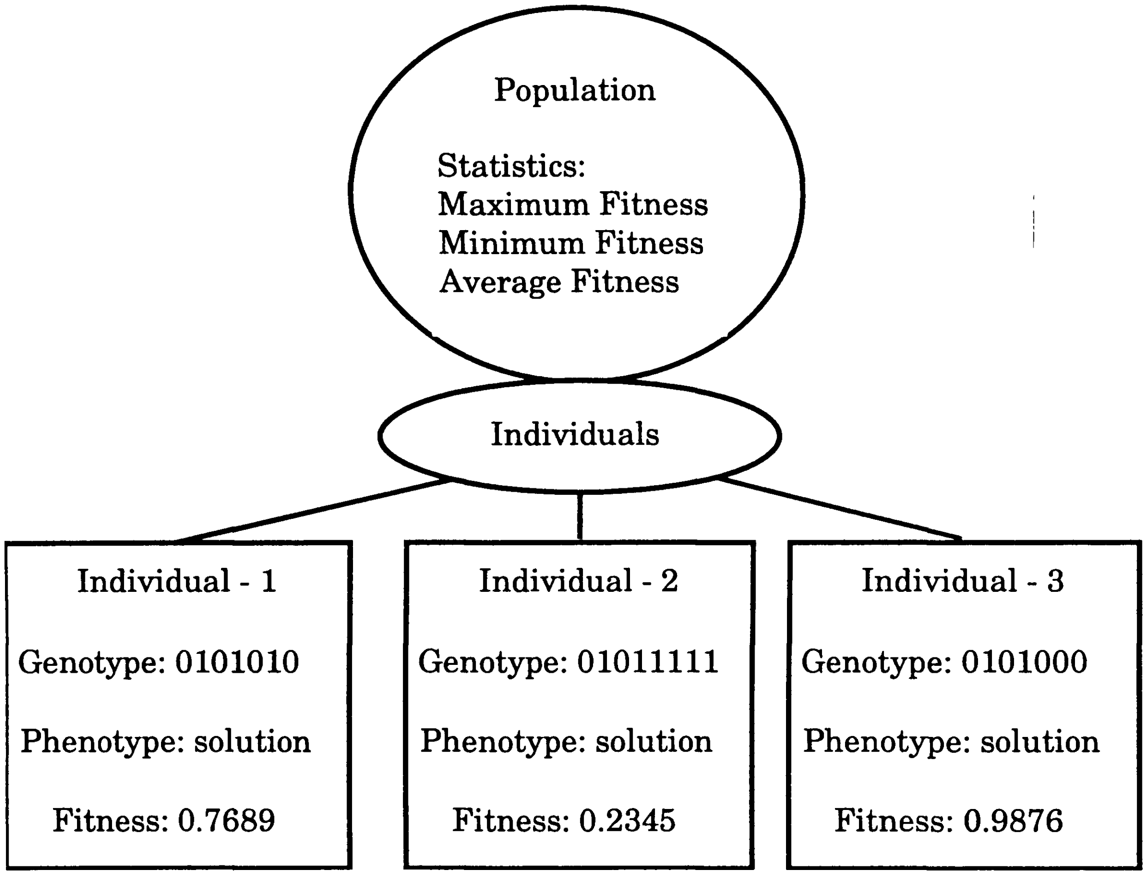
3.1 A Simple Genetic Algorithm

GAs use the terminology of biology in addition to its concepts; although the terminology does not always have the same meaning. The group of structures is referred to as a population and the operations are called mutation, expression, determination of fitness and reproduction. When all the operations have been performed the GA is said to have evolved one generation.

3.1.1 Population

The population is composed of individuals that correspond to possible solutions (Figure 3.1). Each individual has a "chromosome" and a "fitness". The chromosome will be a bit string, an array of binary digits, that encodes a possible solution. The sequence of binary digits is the "genotype" of the

Figure 3.1 Genetic algorithm data structures.



individual. Genotype is a biological term referring to the genetic constitution of an organism. The possible solution an individual represents is its “phenotype”; it is produced by the “expression” of the genotype. Phenotype is a biological term referring to the anatomy and physiology of an organism. It is produced by the interaction of an organisms genotype and its environment. Once a possible solution has been generated its quality or “fitness” can be determined. Once determined, the fitness of the individual will be recorded as a number. The population will also normally record certain statistics about itself, i.e. total fitness of population, maximum fitness, average fitness and minimum fitness.

3.1.2 MUTATION

“Mutation” is the random alteration of the chromosome of an individual. The mutation creates new genotypes and allows the discovery of better solutions. The mutation is performed by genetic operators. There are two types of genetic operators (Figure 3.2 & 3.3):

- Mutation genetic operator (Figure 3.2). This should be more properly called a point mutation operator. Point mutation refers to a change that converts one allele to another. The operator selects a point on an individual's chromosome at random and alters the bit at that point. If the bit is a 1 it becomes a 0 and vice versa. The point mutation operator is the simplest of the genetic operators. Nearly all GAs use a mutation operator, to create new genotypes by a minor modification of the existing one. There can be a problem when the change of one bit can cause a disproportionate change in the phenotype. The problem is dependent on the expression or coding system used (see 3.1.3).

- Crossover genetic operator (Figure 3.3). The crossover genetic operator randomly selects two individuals. The same point on the two chromosomes is selected at random and all the bits from that point to the end of the chromosome are swapped between the two chromosomes. The crossover operator is different from the other operators for two reasons. In the first instance it changes two individuals not one. In the second it produces the new genotypes by combining the existing ones; as opposed to modifying them. A crossover operator is found in all GAs and is one of the most important reasons for their success.

The combination of genetic operators used in a GA vary. A simple GA will have a crossover and mutation operator. There are other genetic operators but they are either specialised or problem specific (Sections 4.6.3, 5.13, 6.1.3, 7.1.3 &

Figure 3.2 Point mutation operator.

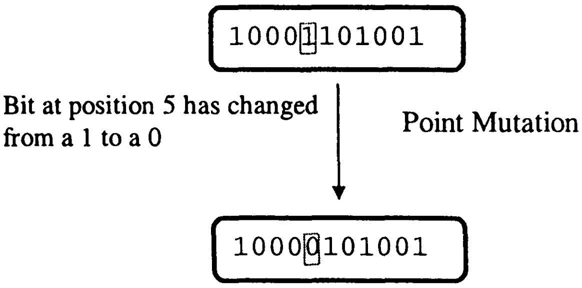
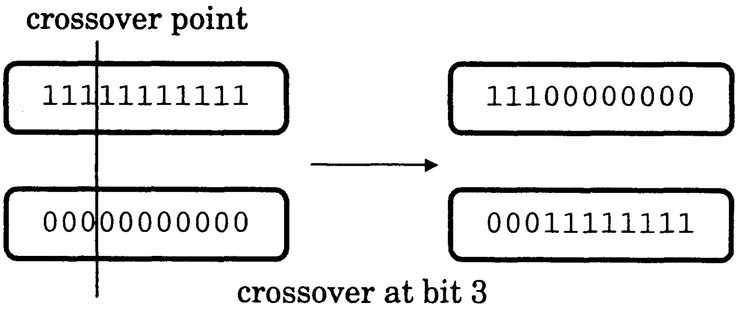


Figure 3. 3 Single point crossover operator.



8.1.3). Once the genetic operators to be used have been decided upon, their frequency of use must be determined. There has been research into the empirical setting of the frequency of genetic operators (30). Currently the practice is to start with values that have proved to be effective in the past and adjust them experimentally for each GA.

3.1.3 Expression

Expression is the conversion of the genotype, the bit string, to the phenotype, a possible solution. Expression is a biological term; coding and mapping are other terms used to describe the process. The expression is problem specific[15]. How the operation is performed is dependent on the problem the GA is trying to solve. To describe how an expression operator is designed an example problem is required; such as the sequential assignment problem outlined in section 2.2.

The aim of sequential assignment is to create a sequence of spin systems linked together by the appropriate cross peaks. The sequence of spin systems will be the same length as the amino acid sequence of the protein under investigation. Each spin system will be linked to the next by up to four cross peaks, NH-NH, α-NH[16], β1-NH and β2-NH (Figure 3.4). In a GA designed to produce a sequential assignment method of converting a bit string, the genotype, to a sequential assignment, the phenotype, has to be found. Figure 3.5 is an example of such a method. The bit string is split into sections that encode for an element of the sequential assignment. To encode a spin system and its NOE links to the next spin system requires five such sections of bit string. The sections of bit string are converted to integers. The integers give the position of the element it encodes for in an array of such elements. The first integer selects a spin system from the array of spin systems. The second, third, fourth and fifth integers select the NH-NH, α-NH, β1-NH and β2-NH cross peaks respectively from the array of cross peaks. The process performed on the entire bit string will create a sequential assignment.

The expression or mapping is one of the two most critical factors determining GA performance. An efficient conversion from genotype to phenotype will have an impact both on the time taken to produce a solution and the quality of that

[15] This is normally called domain specific but the term domain has a different meaning in protein biochemistry. To avoid conflicting meanings the word domain will not be used.

[16] When the second spin system is a glycine there will a second NH-α.

Figure 3.4 NOE cross peak linkage of two spin systems.

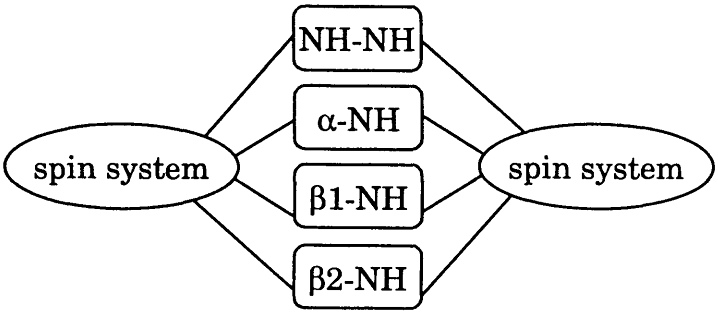
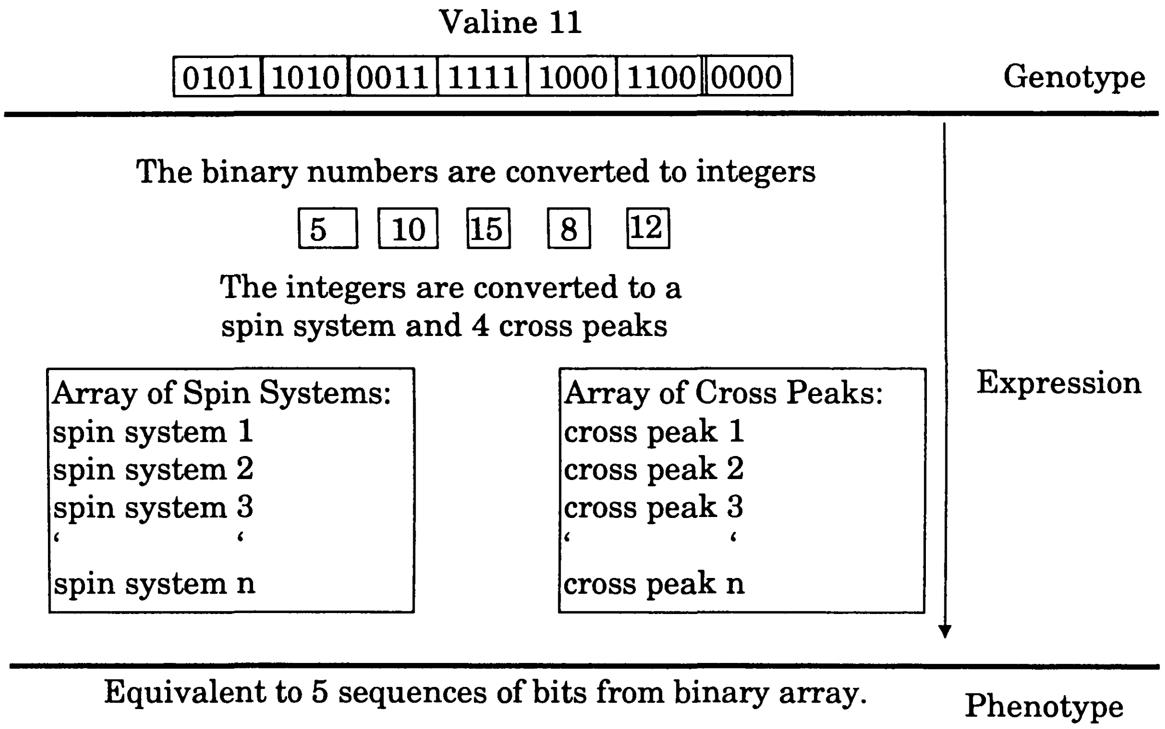


Figure 3.5 The production of a sequential assignment 'phenotype' from a binary array 'genotype'.



Valine 11

The binary numbers are converted to integers

The integers are converted to a spin system and 4 cross peaks

Array of Spin Systems:
spin system 1
spin system 2
spin system 3
' '
spin system n

Array of Cross Peaks:
cross peak 1
cross peak 2
cross peak 3
' '
cross peak n

Equivalent to 5 sequences of bits from binary array.

Spin system 5	Cross Peak 10	Cross Peak 15	Cross Peak 8	Cross Peak 12

solution.

3.1.4 FITNESS

Fitness is the ability of an organism to survive and compete for resources compared with others of its species. A fit individual will survive and reproduce while an unfit one will not. In GAs fitness is the quality of the solution an individual encodes; i.e. the quality of its phenotype. A fit individual will probably have offspring in the next generation, an unfit one will probably not. The evaluation of fitness is problem specific like expression (Section 3.1.3). To describe how fitness is evaluated an example is needed. The sequential assignment problem outlined in section 2.2 will again be used as an example.

The fitness or quality of a sequential assignment depends on three factors:

1. The completeness of the sequential assignment. The completeness is the number of spin systems assigned compared to the total number of spin systems.

2. The quality of the spin system identification.

3. The quality of the NOE links between spin systems.

The fitness can be calculated from the sum of the fitness of each spin system in the sequential assignment. The greater the number of spin systems in the sequential assignment the greater the number that contribute to its fitness, this deals with factor 1. The fitness of an single spin system will be a combination of the accuracy of its identification and the quality of the NOE links to the preceding and succeeding spin systems. Both factors are evaluated for each spin system. The evaluation will produce a number that is the quality of each factor. The two number are multiplied together to give the fitness of the spin system.

Spin system identification will often produce spin systems that have been identified as possibly being several types of amino acid. Each potential identity of the spin system will have a probability associated with it. To cope with the ambiguity of the spin system identification it is encoded as a two element list. The list will be the same length as the number of amino acid types in the protein. The first element in the list will be an amino acid and the second will be the probability that the spin system is that type of amino acid. The probability will be a number between 1.0 and 0.0. Each spin system position in a sequential assignment corresponds to an amino acid at the same position in the amino acid sequence of the protein. The appropriate probability is found for the position the spin system occupies. The probability is the fitness of the spin system identification (Section 2.1.3).

A spin system is linked to the two adjacent spin systems by up to eight NOE cross peaks; four to each neighbouring spin system (Figure 3.5). The fitness of the NOE links of a spin system is the sum of the individual NOE links. A cross peak should link two spin systems by having the chemical shifts of its centre align with a chemical shift of the spin systems, see Figure 3.6. Where the two chemical shifts of the spin systems overlap is the ideal position for the centre of the cross peak. The closer to the ideal position the greater the probability that the cross peak links the two spin systems. Beyond a certain distance from the ideal point there is little or no probability that the peak links the two spin systems. When the peak is in the ideal position it is ascribed the number 1.0. When the cross peak is beyond the certain distance it is ascribed the number 0.0. When in between, the number ascribed is inversely proportional to the distance from the ideal point.

The calculation of the quality of an NOE link when it is within the range where it is considered possible to link the two spin systems is shown in Figure 3.6 and Equation 3.1.

$$Fitness_{NOE} = 1 - \frac{D1 + D2}{2r} \qquad \text{Equation 3.1}$$

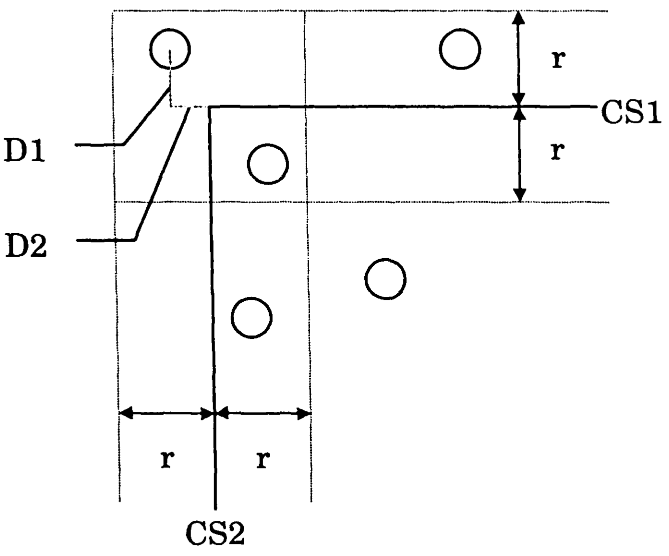
The chemical shift used will depend on the nature of the link; for example in a β1-NH link the β1 chemical shift of one spin system and the NH chemical shift of another will be used.

The fitness operator, often called the fitness function, is the second critical factor in GA performance. It will have a direct impact on the quality of the solution generated and the speed with which it is reached.

3.1.5 Reproduction

In a GA reproduction is the transfer of selected individuals from the current generation to the next generation. An individual's chance of surviving into the next generation is proportional to its fitness. An individual can have none, one or several copies of itself in the next generation depending on its fitness and random chance. The reproduction of individuals according to their fitness has

Figure 3.6 Scoring the quality of an NOE peak.



r	range within which a peak could possibly link the spin systems.
CS1	chemical shift of a spin system.
CS2	chemical shift of a spin system.
D1	distance of peak center from CS1.
D2	distance of peak center from CS2.

two effects. Fitter individuals in the population will survive and increase in number. Unfit individuals will decrease in number and eventually die out. These effects are called selection pressure. The reproduction operator is quite simple:

1. A new population is created.

2. An individual from the old population is selected.

3. The selected individual is copied from the old population to the new.

4. Steps 2 and 3 are repeated until the new population has a full complement of individuals.

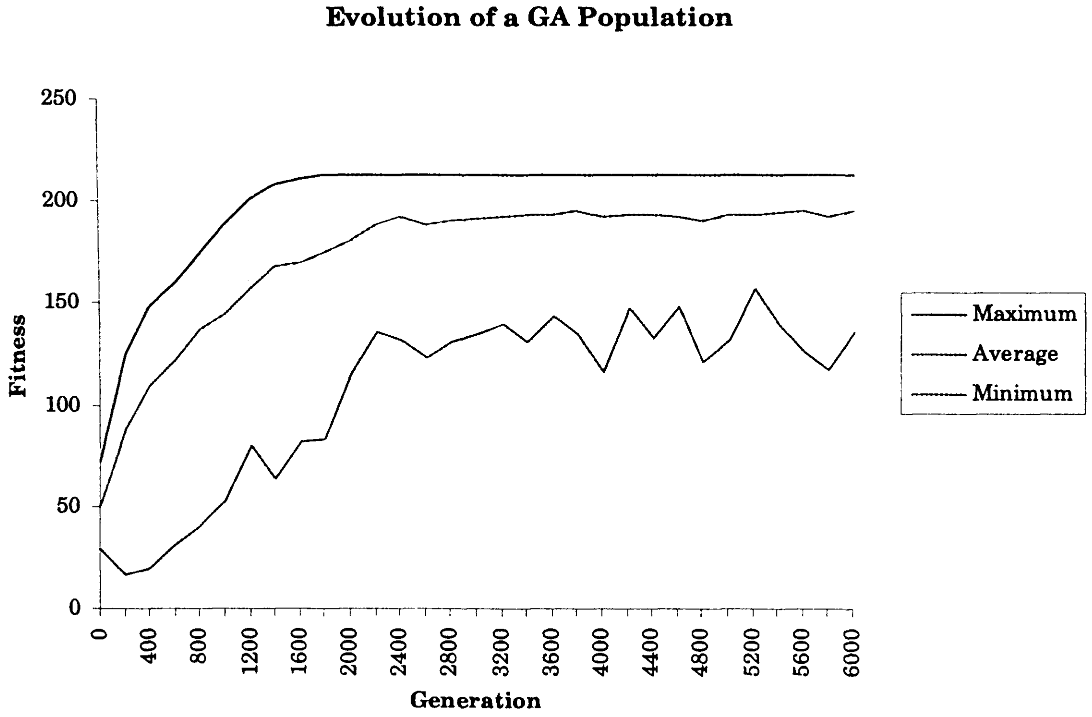
The important part of the process is the selection of the individuals to be reproduced. There are several methods, the simplest being roulette wheel selection. The roulette wheel selection is based on the game of roulette. Each individual is given a segment of a roulette wheel. The size of an individual's segment is in proportion to its fitness. To select an individual the wheel is spun. The individual that generated the segment where the ball lands is selected. In practice the total fitness of the population is calculated. A random number between 0.0 and 1.0 is generated and multiplied by the total fitness; this conceptually is the roulette ball. The individuals of the population are selected one at a time and their fitness is added to a running total. The running total is equivalent to the roulette wheel. When the running total equals or exceeds the random fraction of the total fitness the current individual is transferred to the next generation; in effect the ball lands in the segment of this individual.

3.1.6 Running a Genetic Algorithm

Once a GA has been designed the problem of setting parameters remains. The parameters will be: the population size, the number of generations the population is to be evolved and the frequency of the use of genetic operators. In practice the parameters are set by trial and error. It can take some time to optimise the running of a GA. A graph of the maximum, average and minimum fitness can be seen in Figure 3.7[17]. In a simple GA the difference between the maximum fitness of the population and average fitness of the population would be reduced at the end of the run. This is due to the fact that

[17] The graph comes from a run of the 2D-SAM GA module, Chapter 5.

Figure 3.7 A graph of the evolution of a GA.



the population tends to converge on one individual. The fittest individual will normally have multiple copies of itself in the population and numerous minor variations of it. The fittest individual will dominate the population at the end of the run

GA provide effective answers in a number of fields but to design an effective and efficient GA requires an understanding of how they work. This understanding is particularly important when designing expression and fitness functions.

3.2 GENETIC ALGORITHM THEORY

Understanding how a GA work requires the development of several ideas: schemata, building blocks and search space. Schemata are a method of describing a set of binary strings. A building block is a specific type of schemata. GAs work by processing schemata. Search space is an idea common to all search algorithms that allows the difficulty of the problem to be investigated.

3.2.1 SCHEMA

A schema is a description of a set of bit strings. As an example, take a GA where the chromosome of each individual is an 8 bit long bit string. There are 2^8 possible bit strings. To describe a set of bit strings where the first 4 bits are all 1 and the last bit a 0 the following schema is produced: 1111***0. The * indicates that either a 1 or a 0 can be at the position; it is a wild card character. The following bit strings are all described by the above schema: 11110000, 11110110, 11111110. A schema can define 1 bit string; e.g. the schema 11111010 describes only one bit string. Alternatively a schema can describe all the bit strings possible; i.e. ********.

Schemata have three properties: order, defining length and fitness. The order of a schema is the number of specified bit positions either a 1 or a 0. In the schema 1111***0 the order is 5. There are 5 specified positions and 3 wild card positions, denoted by a *, in the schema. The defining length of a schema is the distance between the two extreme specified positions in the schema. The schema **1*001* has a defining length of 4 (7-3). Schemata will vary in the contribution they make to a bit string's fitness. A very fit individual will be a member of a number of schemata that contribute to its fitness. A schema can be described as fit or unfit depending on the contribution it makes to an individual's fitness. A schema will correspond to a characteristic or characteristics in the phenotype of an individual. These characteristics can confer either fitness or unfitness.

A GA works by spreading building blocks through a population. Building blocks are very fit, low order, short defining length schemata. The factors that define a building block all contribute to its spread. Fitness increases the chance of a schema surviving into the next generation. Reproduction selects fit individuals more frequently than unfit ones. Fit individuals will be members of fit schemata. Therefore fit schemata will be reproduced more frequently than unfit ones. Low order and short defining length enhance the chance of a schema surviving the action of genetic operators. A low order reduces the chance of mutation altering a schema. A short defining length reduces the chance of crossover altering a schema.

3.2.3 Schema Theorem

The schema theorem describes the change in the number of schemata in a population from one generation to the next. There are two factors that affects the change in number: the chance of surviving into the next generation and the chance of surviving the disruptive effects of genetic operators: both mutation and crossover.

A schema H occurs $m(H,t)$ times in a population at generation t. The number in the next generation will be $m(H,t+1)$. The chance of surviving into the next generation S_n is defined in Equation 3.2; where $\overline{f}$ is the average fitness of the population and $f(H)$ is the average fitness of the individuals that are members of schema H.

$$S_n = \frac{f(H)}{\overline{f}} \qquad \text{Equation 3.2}$$

The chance of surviving mutation S_m is described in Equation 3.3; where p_m is the mutation rate and the order of the schema H is $o(H)$. When both p_m and $o(H)$ are small the equation can be approximated to 1- $o(H)p_m$.

$$S_m = (1 - p_m)o(H) \qquad \text{Equation 3.3}$$

The chance of surviving crossover S_c is described in Equation 3.4; where $\mathrm{p_c}$ is crossover rate, δ(H) is the defining length of schema H and l is the length of the bit string.

$$S_c \geq 1 - p_c \times \frac{\delta(H)}{l-1}$$ Equation 3.4

The S$_c$ is an inequality because the crossover might restore the schema if the two individuals are similar. The number of schemata in the next generation from *t* is shown in Equation 3.5.

$$m(H,t+1) \geq m(H,t) \bullet S_n S_m S_c$$ Equation 3.5

Expanded and simplified it gives what is known as the schema theorem Equation 3.6.

$$m(H,t+1) \geq m(H,t) \bullet \frac{f(H)}{\bar{f}} \bullet \left(1 - p_c \frac{\delta(H)}{l-1} - o(H) p_m \right)$$ Equation 3.6

The schema theorem implies that the number of building blocks will grow exponentially over time.

3.2.4 Search Space

The idea of a search space is common to all search algorithms. A search space is a space where all the possible solutions to a problem exist. Similar solutions will be near each other while different solutions will be far apart. The fitness of a solution will also be a factor in its position. The search space is often visualised as a three dimensional graph. Each solution will be a point on the graph. The X and Z dimension will be some comparison of the solution's similarity to other solutions. The Y or height dimension will the solution's fitness. The search space is a product of the problem under investigation and the expression used. The search space has two characteristics that affect the difficulty of a problem: size and shape.

The size of a search space is the number of solutions it contains. Search space size in a simple GA will depend on the length of the bit string; e.g. a bit string 8 bits long will have 2.560 x 10^2 (2^8) possibilities, a bit string 32 bits long will have 4.295 x 10^9 (2^{32}) possibilities and a bit string 256 bits long will have 1.158 x 10^{77} (2^{256}) possibilities. As the length of the bit string increases the number of combinations it has increases exponentially. For example a 256 bit string is 8 times longer than a 32 bit string but the number of possibilities has increased by 68 orders of magnitude. The size of a search space will have an impact on the time taken to reach a solution. A large search space makes an exhaustive search prohibitively expensive in terms of computer time. The size of a search space has a secondary impact on the quality of the solution chosen.

The primary factor affecting the quality of the solution chosen is the shape of the search space.

The shape of a search space is dependent on the problem being investigated and the coding used. The shape of the search space affects the solution chosen by increasing the likelihood of choosing a local optimum instead of the global optimum. A local optimum is a peak in the space that is not the highest peak in the search space. If an initial solution is near a local optimum then the local optimum can come to dominate the population. If this occurs the global optimum will not be found because to get to it intermediate individuals would have a lower fitness than the local optimum. The size of the search space can also have a certain impact on the probability of this. Initial solutions are often chosen randomly; they effectively sample the search space. How well they do this is dependent on the size of the population and search space. The larger the population the better sampled a given search space. The smaller the search space the better it will be sampled for a given size of population. After the initial sampling the relative sizes and of the search space the GA population will also affect how well the search space is explored by the GA, although its shape is the predominant factor.

In a GA there are two processes working all the time: exploration and exploitation. Exploration is the search for new solutions. The higher the frequency of genetic operators the more the GA explores the search space. If there is too little exploration the GA is likely to choose a poor solution; a local optimum. Exploitation is the convergence of the population on the best individual. The higher the selection pressure and the lower the frequency of genetic operators the more the GA exploits the best individuals in the population. Selection pressure is the ratio best individuals fitness to the average fitness of the population. Genetic operators can disrupt fit individuals. When there is too little exploitation the GA makes very little or no progress. The two processes must be balanced to choose a good solution in a reasonable amount of time. The balance is not easy to achieve as the two processes are antagonistic. The balance will vary for each GA and some times for each problem.

The GA algorithm described here is a simple or classical GA. There have been a number of refinements and adaptations to GAs to enable them to cope with different problems. The four GAs I have designed and implemented have made use of a number of developments. The GA core, the problem domain independent elements, are the same for all the GA modules. The core of the

GA modules is described in the next chapter.

4.0 Design of the Genetic Algorithm Core

All the GA modules described in subsequent chapters have the same core, the problem domain independent elements. In a traditional GA, such as the one described in chapter 3, the problem domain dependent elements are the coding and fitness function. In the various modules there is an additional problem domain dependent element, the domain specific genetic operators. These genetic operators are tailored to each problem domain to improve the performance of each module. The design of the GA core is described in this chapter while the design of the coding, fitness function and genetic operators is described in the relevant module chapters (Chapters 5,6,7 & 8).

4.1 GA Core Design Principles

The GA core is based on the simple GA described in chapter 3 but incorporating one major and several minor modifications. The major modification is the use of a crowding factor (31) and the minor modifications are the use of fitness scaling (32, 33) and the stochastic remainder sampling to select the individuals to reproduce.

A crowding factor is used to increase the diversity of a population and thus improve exploration of the search space and is modelled on a biological process. When organisms are in a crowded environment they have increased competition for resources. An example would be plants competing for light in a rain forest. When a plant is in a location where it is crowded by other plants it has to compete for the available light. If the plant was unfit by comparison with its neighbours then it would probably not survive. If the same plant was in a location were there were few or no other plants; then it probably would survive because of the reduced competition. The crowding factor mimics this effect during the evolution of a GA population. The individuals of the population exist in a search space. If most of the individuals in a population are similar they will occupy a similar position in the search space. They crowd around a location in the search space. Those individuals that are different occupy a different position in the search space. The crowding factor improves the probability of survival for isolated individuals and reduces the probability of survival for crowded individuals.

This is achieved by having two populations: a main population and a sub population. The main population contains all the individuals in the population of the GA. The sub population contains a subset of the main population. Individuals from the main population are copied into the sub population. The individuals in the sub population undergo mutation and reproduction and then replace selected individuals in the main population. The crowding factor operates in the selection of the individuals to be replaced. To select an individual for replacement several individuals are chosen from the main population. Each of the chosen individuals is compared to the individual from the sub population; the most similar one being selected for replacement. The more common the genotype of an individual, or parts thereof, the greater the probability of it being replaced.

An addition to the crowding factor is the use of the "parent" of an individual. The parent of an individual is the individual in the main population it was copied from. The parent is used as one of the individuals considered for replacement. This was done by Mahfoud (34). The paper states that when the entire main population was searched for the best match for an individual from the sub population its parent was selected 83% of the time. When candidates for replacement from the main population are selected the parent of the individual from the sub population is always one of them. When this was implemented in the 2D-SAM it reduced the performance of the module. Highly fit individuals did not increase in number as rapidly, thus reducing the exploitation of such individuals.

The crowding factor used in the various modules was a modification of the one described above. Three refinements were developed to the crowding factor. The first refinement is competition. Competition was introduced by comparing the fitness of the individual from the main population with that of the individual in the sub population that will replace it. The replacement only proceeds when the sub population individual is fitter. This refinement was introduced to prevent highly fit individuals being removed from the main population before they have an opportunity to reproduce effectively. Effective reproduction is defined as producing an offspring that has a similar fitness. Without this comparison highly fit individuals from the main population can be replaced by unfit ones. By making the individuals compete, which is in keeping with the crowding concept, the demise of highly fit individuals before they can reproduce effectively is prevented.

The refinement succeeded in preventing the loss of highly fit individuals from the main population but it also reduced its diversity. The reduced diversity was caused by a reduction in the number of individuals that are replaced. Individuals from the sub population, having undergone mutation, are more diverse than those from the main population. The refinement is essentially an extreme elitist selection (35). Instead of ensuring that the best individual survives, elitist selection, it ensures that most of the fit individuals survive. To improve the performance of the GA another concept was developed within the GA that allowed the second and third refinements. The concept is that of each individual having an age. Each individual initially has an age of zero. The age of an individual in the main population is increased by one every time it is copied to the sub population. Once an individual has been copied into the sub population the age of the copy is reset to zero. An individual in the main population increases in age every time it has an opportunity to reproduce. While individuals in the sub population which are the result of that reproduction are considered to be new individuals and therefore have no age.

The first refinement is the death age factor. One of the user defined parameters for the GA will be the death age, an integer. The age of each individual in the main population selected as a possible candidate for replacement is examined. If its age is greater than the death age parameter the individual becomes the candidate for replacement. The death age removes individuals from the population that have reproduced a certain number of times. The crowding factor can cause unfit but isolated individuals to survive for a very long time. The death age factors remove these individuals.

The second age related refinement is the old age factor. One of the user defined parameters for the GA will be the old age, an integer. Once a candidate for replacement has been selected its age is again examined. If its age is greater than the old age parameter the individual is replaced. It is now so old that it cannot compete with younger individuals. If an individual is not above the old age parameter then it is young enough to compete and the fitness comparison is performed. The death age is greater than the old age, logically. Therefore any individual selected as candidate for replacement because its age is greater than the death age parameter is replaced. The old age factor keeps the population dynamic. The factor allows new individuals from the sub population into the main population at a reasonable and controllable rate. The factor also gives a highly fit individual an excellent chance to reproduce effectively. An individual will have as many chances to reproduce as the old age parameter, unless it is replaced by a fitter individual.

Fitness scaling is a proven technique for improving the performance of a GA. There are three types of scaling: linear, sigma truncation and power. Linear scaling is used in the various GA modules. In scaling the fitness of each individual is altered to give it a scaled fitness. The maximum, average and minimum fitness of the sub population are used to scale the fitness of each individual. The maximum scaled fitness will be some multiple of (in practice twice) the average fitness. The scaled average fitness will be kept the same as the average fitness. If the maximum fitness of the population is greater than twice the average fitness those individuals with above average fitness will have a scaled fitness less than their original fitness. While those individuals with below average fitness will have a scaled fitness greater than their original fitness. When the maximum fitness of the population is less than twice the average fitness the reverse is true.

The scaling of fitness has two purposes. The first is to prevent premature convergence of the main population. As the GA evolves it converges; one genotype will come to dominate the main population. Premature convergence occurs when an exceptionally fit individual appears early in the evolution of the GA. If the individual has a fitness several times greater than the sub population average it would probably have several offspring. Each of offspring could itself generate several more and so on until the genotype of the original individual, and minor variants of it, is the only one in the population. Scaling prevents this by limiting the number of offspring to 2. An individual with average fitness will probably have one offspring in the next generation (Section 3.1.5). An individual with twice the average fitness will probably have two offspring in the next generation. Scaling helps maintain a balance between exploitation and exploration by preventing over exploitation.

The second purpose of scaling is to promote fitter individuals as the main population starts to converge. As the population converges the differences in fitness between individuals will be small. The scaling of the fitness of the individuals magnifies these small differences. For example if the fittest individual in the sub population had a fitness of 5% greater than the population average it would have a 5% greater probability of an offspring than one with average fitness. With fitness scaling the individual would have twice the average fitness and therefore twice the probability of an offspring in the next generation. It is not always possible to scale up the maximum fitness to twice the average. Those individuals with below average fitness will have their fitness reduced by scaling in the circumstances outlined above. The fitness of an individual cannot be negative and this limits the amount of scaling.

Remainder stochastic sampling is an improved method of producing a new generation of individuals. In the simple GA the roulette wheel method was used (Section 3.1.5). The method of calculating the number of children is the same as for the simple GA, except for fitness scaling. The scaled fitness of an individual is divided by the average fitness of the sub population to give the number of children it will have in the next generation. The stochastic remainder method allocates the actual number of offspring an individual will have in two stages. In the first stage all the individuals with a whole number of offspring are allocated that number of offspring in the next generation. For example, if an individual is calculated to have 1.34 offspring in the next generation then it will have 1 offspring in the next generation. In the second stage the remaining places in the population are filled using the fractional part of the calculated number of offspring. The individuals are examined one after another. In each examination the fractional part of the calculated offspring is compared to a randomly generated number between 0.0 and 1.0. If the randomly generated number is less than the fractional part of the number of offspring then the individual being examined is placed in the next generation. Using the previous example 0.34 would be compared to a randomly generated number. If the number generated is less than 0.34 then the individual would have another offspring in the next generation. The examination of the population continues until the next generation has a full complement of individuals.

The remainder stochastic sampling method ensures that every individual with a fitness greater than the population average receives at least one offspring. With roulette wheel selection this was a probability but not a certainty. This certainty is its advantage over the roulette wheel sampling method; relatively fit individuals always have offspring. The remainder or fractional part of the calculated offspring number is used in a stochastic way to determine the rest of the population. There will always be gaps in the next generation because those individuals with below average fitness will not have any offspring in the next generation after the first stage of the sampling. In the second stage each probability of an individual having offspring in the next generation is equal to a fractional part of its offspring number. The second stage is similar in its stochastic nature to the roulette wheel selection except there is a reduction in the number of individuals left to be selected. As a result the probability of a below average individual having offspring in the next generation is reduced. The method guarantees that fit individuals will have offspring while maintaining the concept of the number of offspring being proportional to the fitness of the parent.

The design of all the GA modules is an object oriented one (36). The objects that comprise the GA core are the population and individual objects. The next three sections describe the design and functions of these objects. An object, in the context of object oriented design, is an instance of a class. The class defines the nature of an object and each instance of an object is created using the class definition.

4.2 INDIVIDUAL OBJECT

The individual class forms the interface between the problem domain dependent and independent parts of the various GA modules. The class definition and the purpose of its various functions are domain independent. The actual design and implementation of the functions of the object are domain dependent. The purpose of the individual object is to encode an assignment of one or more protein NMR spectra. The individual also contains information about the assignment and about the individuals behaviour in the GA.

4.2.1 INDIVIDUAL CLASS DEFINITION

The original design of the GA core had a chromosome class. This class was included in the design of the GA core to allow variation in the implementation of the chromosome. The chromosome class provided another level of abstraction. The use of a standard chromosome implementation made the class redundant. The chromosome class is now a super class of the individual class. The individual class inherits all the attributes of the chromosome class. As the object inherits the attributes of the class chromosome the definition of the chromosome class is included in this section .

The chromosome representation chosen for all the GA modules was an integer array. The chromosome class definition given here is designed to represent the features necessary for an integer array chromosome. The class has five attributes which are listed in Section 4.7.1. The individual class describes the behaviour of an individual within the GA core. The individual class has nine attributes which are again listed in Section 4.7.

4.2.2 INDIVIDUAL CLASS FUNCTIONS

The functions of the individual class, except for the accessor functions, mainly use the attributes of the chromosome super class. These functions are some of the domain dependent elements of the GA modules.

The functions of the chromosome class are confined to constructing the chromosome object and functions that operate on the chromosome-array attribute. These functions either read, write to, copy or compare the chromosome array; they were written as part of the extra layer of abstraction.

The extra abstraction allows a change of chromosomal representation while the other classes of the module remain constant.

The functions of the individual class are the expression functions, genetic operators and the fitness function. These are all problem domain dependent functions. There are many population class functions that make use of the individual class attributes. The function used to copy an individual from the main to sub populations uses one of the chromosome functions, the chromosome copy function.

4.3 POPULATION CLASS

The population classes form the main element of the GA core. The class produces an object that embodies the population that evolves the required solution. In the simple GA described in chapter 3 there is only one population. In each of the GA modules there are two populations; the main and sub populations. Both populations have a number of elements in common. To produce the required two types of population there are three population classes. There is a population super class and two sub classes. The classes are the population, main population and sub population classes respectively. The population class describes the elements common to both the main and sub population.

4.3.1 POPULATION CLASS DEFINITION

The population class contains the attributes needed by both the main and sub population classes. The class has no functions. All the functions operate on either or both the two sub classes. The eight attributes of the class are listed in Section 4.7. These attributes are added to by the class definitions of the main and sub population classes to form the main and sub population objects.

4.4 MAIN POPULATION CLASS

This class defines the main population of a GA module. There is one instance of the main population class in each module. The main population is the population of individuals that evolves for a set number of generations to produce a solution. The initial individuals of the main population are constructed and then evolved a set number of generations. The best individual to have existed in the population is used as the solution to the problem. The main population is equivalent to the population in a simple GA. The sub population is used to reproduce the individuals of the main population. The crowding factor requires the use of two populations (section 4.1).

4.4.1 Main Population Definition

This class inherits the attributes of the population class. There are two attributes defined in the main population class. These attributes are listed in Section 4.7.

The population current generation attribute will determine how long the GA module will run for and the best ancestor will be the solution produced by the module. The best ancestor is used rather than the best individual because even the fittest individual in the population can be replaced.

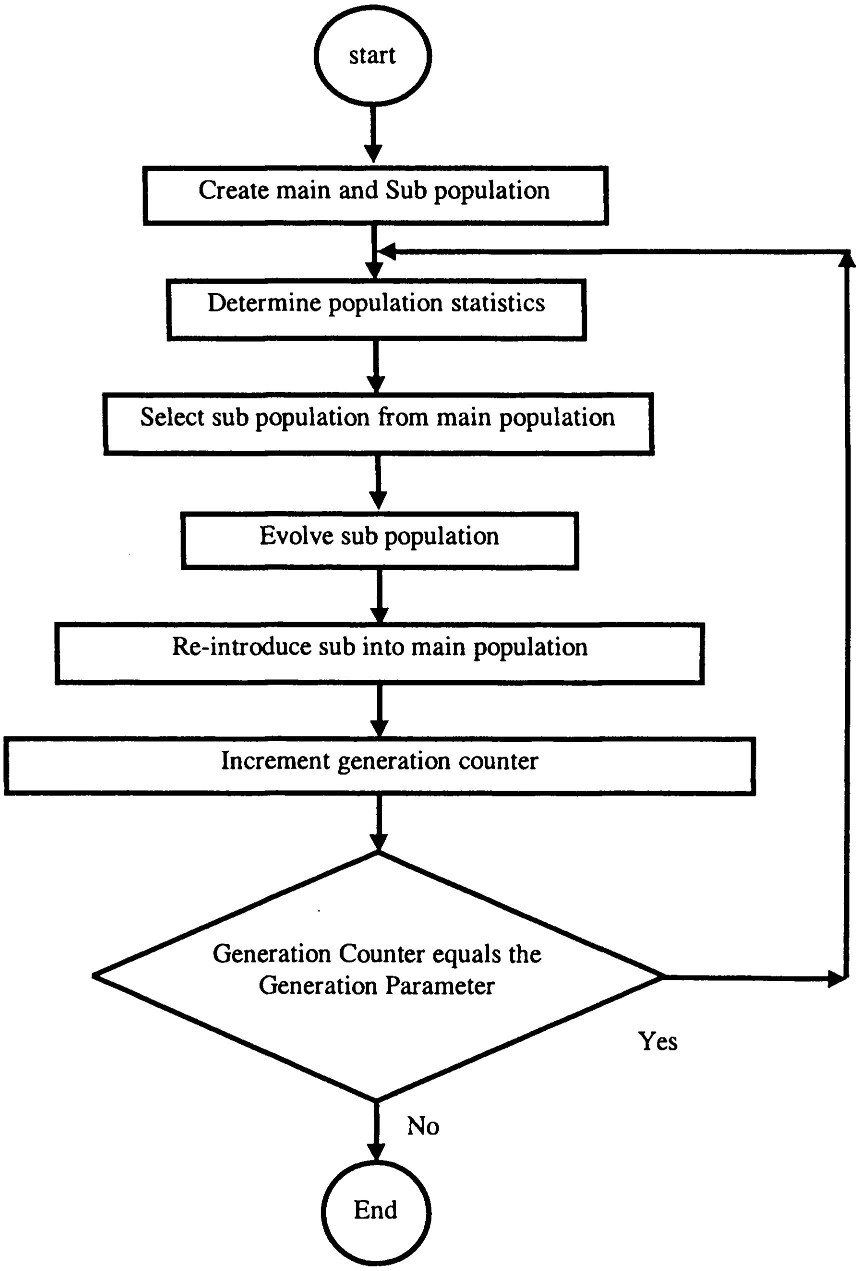
4.4.2 Main Population Class Functions

The functions of the main population class are the construct population function and those functions contained within the evolve population function. The construct population class creates the main population object. The evolve population function creates a consistent population, controls the number of generations that main population evolves, controls interaction with the sub population and performs a series of recording functions. A flow chart of the functions of the main population is shown in Figure 4.1.

The main population is constructed using three user defined parameters, all of which are integers. The parameters are the population size, the length of the protein amino acid sequence and the number of integers needed to encode one amino acid in the chromosome. The population size defines the population-size attribute of the main population and the number of individuals in the population-individuals attribute. The other two attributes are used to calculate the length of the chromosome array of each individual (Section 4.2.1). The number of amino acids in the protein is multiplied by the number of integers needed to encode the information for one amino acid. The default value for the integers of the chromosome array, an integer array, is 0 which denotes the relevant blank objects in all the GA modules. Thus the first assignment for every individual is a blank one.

The create consistent population is the first function called by the evolve population function and creates a consistent and plausible initial assignment. The function invokes the create consistent individual function for each individual of the population. The function gives an individual a consistent and plausible initial assignment. It is a problem domain dependent function; it depends on the coding used (Sections 5.1.1, 6.1.1, 7.1.1 and 8.1.1). A consistent assignment is one where each object in an assignment is used once. For example, where an NOE peak is used only once in a sequential assignment

Figure 4.1 A flow chart of the main population class functions.


start
Create main and Sub population
Determine population statistics
Select sub population from main population
Evolve sub population
Re-introduce sub into main population
Increment generation counter
Generation Counter equals the Generation Parameter
Yes
No
End

(Sections 5.1.1 and 6.1.1). A plausible assignment is created using the information encoded in the fitness function. The information is used to construct the fittest possible local assignment. An example would be creating an initial peak system assignment from several triple resonance experiments (Chapter 7). The first peak in a peak system is chosen at random. The next peak chosen is one that has the best alignment with the previous peak (alignment equates with fitness in this context). The process is repeated for the remaining peaks of the peak system and then the remaining peak system of the assignment. The result of the process is an assignment that is locally fit and therefore plausible.

The control of the number of generations is dependent on the generations parameter and the population-current-generation attribute. The generations parameter, an integer set by the user, defines the generation at which the evolution of the main population will terminate. The population-current-generation attribute records the number of generations that the main population has evolved. When the population-current-generation attribute is greater than the generations attribute the evolution of the main population is stopped.

The interaction between the main and sub populations is controlled by two functions. One transfers individuals from the main population to the sub population, and the other transfers the individuals of the sub population to the main population. The transfer of individuals from the main to sub population is performed by the select sub population function. The function randomly selects individuals from the main population and places them in the population-store attribute (section 4.5.1) of the sub population, until the attribute is filled with individuals. The attribute is an array of individuals and once filled the evolve sub population function is invoked (section 4.5.2). This is the main function of the sub population class.

Once the sub population has been evolved the individuals of the sub population are transferred to the main population. The transfer involves replacing an individual in the main population with one from the sub population. It is during this transfer that the crowding factor is used (section 4.1). The function that accomplishes the transfer of individuals between the two populations is the introduce sub population function. The function attempts to transfer all individuals of the sub population to the main population.

The first step in attempting to transfer an individual from the sub population is selecting an individual in the main population it will replace, a candidate for replacement. A candidate is found using the population-pointer attribute and the crowding factor parameter. The population-pointer attribute of the main population points to the first individual to be compared. Before an individual is compared to the individual from the sub population its age is examined. If its age is greater than the individual-death-age attribute the individual becomes the replacement candidate and the search is terminated. If the age of the individual is less than or equal to the death age; its chromosome array is compared with that of the individual from the sub population. If the individual is the best match so far it becomes the replacement candidate. The population-pointer attribute is then incremented[18] so that it points to the next individual. The number of individuals in the comparison is defined by the used-population-crowding-factor attribute[19]. Once the search has finished the replacement candidate is either the most similar individual to the one in the sub population or its age is greater than the death age.

The second step in the process is the decision of whether to replace the individual in the main population with the individual from the sub population. This decision is based on two comparisons. The first compares the age of the replacement candidate against its individual-old-age attribute. If its age is greater than the individual-old-age attribute then it is replaced. If it is not replaced its fitness is compared to the fitness of the individual from the sub population. If the individual from the sub population is fitter then it replaces the individual from the main population, the replacement candidate.

The last function is the recording function, of which there are essentially three sets. They are grouped by the type of recording being performed and the frequency with which they are invoked. The frequency will be the number of generations that pass between invocations. The first set is the most commonly invoked and it outputs to the screen or a file. The output will be a single line of text for all the modules giving: the current generation, the maximum fitness of the population, the average fitness of the population, the minimum fitness of the population and the current time. The second set of record functions will vary from module to module and will commonly be invoked less frequently

[18] When the population-pointer attribute points to the last individual in the population-individuals attribute it is not incremented, but is reset to zero so that it points to the first individual in the population-individuals attribute.

[19] Section 4.5.1

than the first set of functions. These functions will record the performance of the GA module throughout its evolution. The last set of recording functions will be invoked infrequently, every thousand generations for instance. The purpose of these functions will be to give interim reports on the best solution. The reports are to give the best solution so far if the evolution of the GA module terminates prematurely.

4.5 Sub Population Class

The sub population class defines the sub population of the GA modules. There is one instance of the sub population class in each module. The sub population class contains a subset of the individuals from the main population. The individuals in the subset are reproduced, mutated and their fitness is determined. The reproduction phase is problem domain independent, while the mutation phase is problem domain dependent, since domain specific genetic operators are used. The determination of the fitness is domain independent at the population level, the domain dependent element being at the individual level. Once the fitness of the individuals of the sub population has been determined they can be returned to the main population by the re-introduce sub population function.

4.5.1 Sub Population Class definition

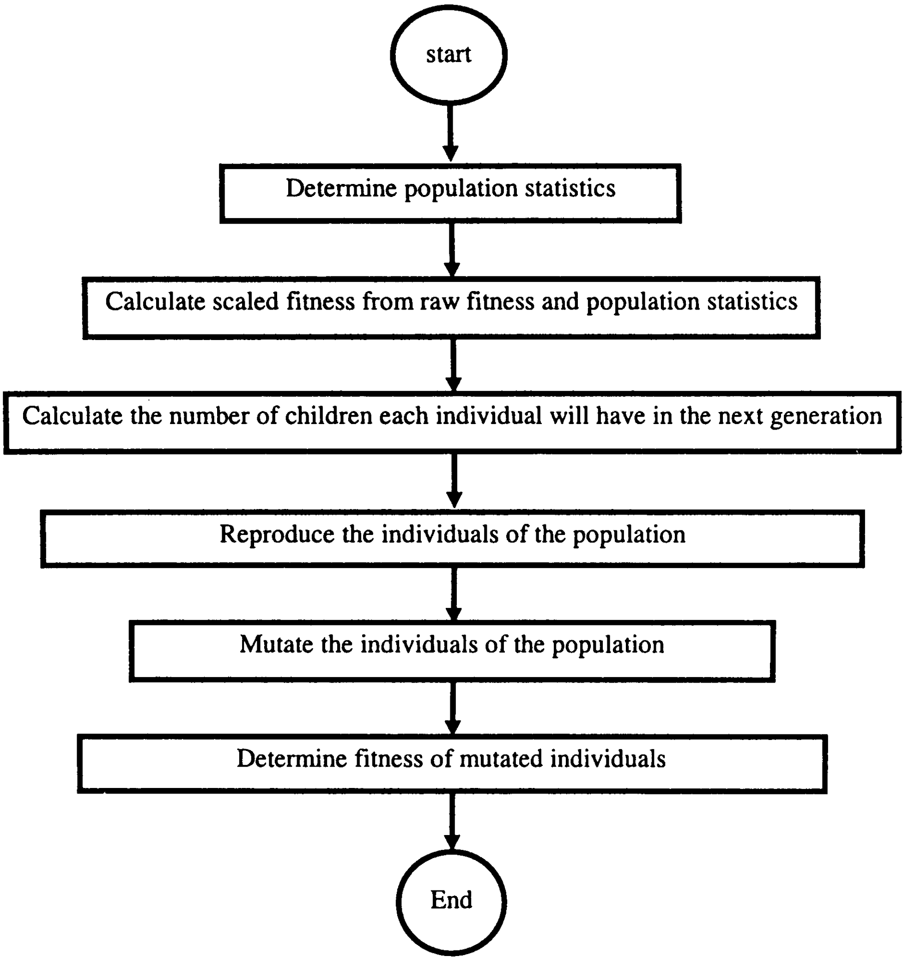
The sub population class has five additional attributes to the population class. The attributes are to enable the class to reproduce the individuals it contains. The attributes are listed in Section 4.7. All the attributes, with the exception of the population-store, are used to contain user defined parameters that control the reproduction of the sub population.

4.5.2 Sub Population Class Functions

The functions of the sub population class reproduce, mutate and determine the fitness of its individuals. The reproduction phase is performed using four functions: fitness statistics, scale fitness, calculate children and reproduce. They are called in sequence and all operate on the individuals in the population-store attribute. The reproduce function copies the individuals from the store to the population-individuals attribute of the sub population, which is the next generation. All the subsequent functions operate on individuals from this attribute. A flow chart of the functions of the sub population is shown in Figure 4.2.

The population statistics function calculates certain statistics about the fitness of the population. The statistics are the total, maximum, average and

Figure 4.2 A flow chart of the sub population functions.


start
Determine population statistics
Calculate scaled fitness from raw fitness and population statistics
Calculate the number of children each individual will have in the next generation
Reproduce the individuals of the population
Mutate the individuals of the population
Determine fitness of mutated individuals
End

minimum fitness of the population. The statistics are used by the scaling fitness function to create the scaled fitness for each individual.

The scaling function scales the fitness of all the individuals in the sub population. The scaling is used to keep the fitness of individuals within certain limits (Section 4.1). There are two calculations performed depending on the statistics generated by the previous function. The first calculation scales the fitness so that the maximum fitness is twice the average. The second calculation is used when the first reduces the scaled minimum fitness below zero. Both calculations use the equation of a straight line, Equation 4.1 with F, F_s, g, c, being the fitness and scaled fitness of an individual, the gradient and the constant respectively. The difference between the two calculations is the method used to calculate the gradient. The first method is shown in Equation 4.2 with max, avg and min being the maximum, average and minimum fitness of the population. The second method can be seen in Equation 4.3. The calculation of the constant is shown in Equation 4.4. The relevant gradient and the constant are then used to calculate the scaled fitness for each individual. Once the scaled fitness for each individual has been determined; the number of children the individual will have in the next generation will be calculated.

$$F_s = gF + c \qquad \text{Equation 4.1}$$

$$g = \frac{avg \times 1}{\max - avg} \qquad \text{Equation 4.2}$$

$$g = \frac{avg}{avg - \min} \qquad \text{Equation 4.3}$$

$$c = avg - (g \times avg) \qquad \text{Equation 4.4}$$

The function calculate children determines how many children each individual will have in the next generation. The calculation is based on the stochastic remainder sampling concept described in section 4.1. The function uses Equation 4.5 to calculate the number of children an individual will have (N_c). The equation uses three numbers: the scaled fitness of an individual (F_s), the total fitness of the sub population (F_T) and the number of individuals in the sub population (N). The calculated number of children number is then split into its integer and remainder components, for example 1.45 will become 1 and 0.45. The two figures are calculated for each individual and then stored in its individual-child-number and individual-fractional-child attributes.

$$N_c = \frac{F_s}{F_T} \times N \qquad \text{Equation 4.5}$$

Once the number of children from each individual has been calculated the next generation can be produced by the reproduce function. The function creates the next generation in the individuals attribute of the sub population. The current generation is held in the store attribute of the sub population. The next generation is produced in two stages. In the first stage every individual in population-store is examined. If the individual has individual-child-number greater than zero then the individual is copied to population-individual array individual-child-number number of times. In the second stage the individuals of the store are examined again. The attribute examined this time is the fractional child attribute. A random number is generated between 0.0 and 1.0. If the number is less than the individual-fractional-child then the individual is copied to the population-individuals array. The second examination and copying of individuals continues until the population-individuals array has been filled with individuals. All further processing of the individuals of the sub population is performed on the individuals in the population-individuals array.

The last two functions of the reproduction phase are quite simple. They both set an attribute of the individuals in the sub population to a certain value. The first function sets the age attribute of all the individuals to zero. The individuals have just been born and therefore have zero age. The second function sets the individual changed attribute to nil to indicate that the individual has not yet been changed by the genetic operators.

The mutation phase is almost entirely problem domain dependent, due to the use of domain specific genetic operators. The number and types of operators will vary from module to module. The genetic operators of each module will be described in the chapters dedicated to that module, but there are some elements that are domain independent. The frequency or rate at which the genetic operators are invoked is controlled by the same attributes of the sub population, the crossover and various mutation numbers (section 4.5.1). The way the individuals are selected for mutation is the same for all the modules. The individuals are selected by using the population-pointer attribute of the sub population. This points to the first individual in the sub population at the start of the mutation process. As required by the various operators the individual pointed to is selected and the pointer is incremented. When the pointer is incremented it points to the next individual in the sub population.

This is done so that each individual is subjected to only one mutation event. This improves the performance of the modules over a random selection of individuals. A plausible explanation of this is that probability of a beneficial mutation event is small and decreases as the population evolves. The chances of two beneficial mutation events occurring is geometrically less probable. A beneficial mutation followed by harmful one will probably result in an overall decrease in the fitness of the individual. Therefore having a single mutation event per individual is more efficient. This would be especially true with the more sophisticated genetic operators which can be quite demanding on system resources. When an individual is subjected to a genetic operator its changed attribute is set to true. This is to improve the efficiency of the fitness determination for each individual.

The population fitness function is a relatively simple function. It invokes the relevant expression and fitness function. The expression and fitness functions are domain dependent; in certain modules the expression function is integral to the fitness function. The expression and fitness functions for each module are described in the chapter dedicated to that module. Before the fitness function is invoked for an individual its changed attribute is examined. If the attribute is nil, indicating that the individual has been unchanged by the genetic operators, then the fitness function is not invoked. If an individual has not been mutated the fitness recorded in its fitness attribute is still accurate. The expression and fitness functions are the two most time consuming elements of the various GA modules. Therefore any reduction in number of times they are invoked improves the performance of the module.

Once the fitness attribute of each individual is accurate the functions of the sub population have completed their purpose of evolving the sub population one generation and the individuals are then returned to the main population by the introduce sub population function of the main population object (section 4.4.2).

4.6 PROBLEM DOMAIN DEPENDENT ELEMENTS

The problem domain dependent elements are the coding, the fitness function and the genetic operators. Even though these elements are domain dependent and they are related because the problem domains are related;. the assignment of a protein NMR spectrum or spectra. The next three sections describe some of the design concepts common to the domain dependent elements of the various modules.

4.6.1 CODING

The modules all use the same chromosome representation. The chromosome representation used in the simple GA described in chapter 3 uses a binary array representation. There are other representations (37): integer and floating point arrays and lists. For the first module a binary array was used (section 5.1.1). During the development of the module the chromosome representation was changed to that of an integer array. The integer array proved to be a more intuitive and efficient chromosome representation for the data and was used in all the subsequent modules.

The conversion from the integer array to the relevant assignment also has common elements in all the modules. Each assignment is composed of objects. The objects can be of one type e.g. peaks objects or there can be several types e.g. peak and spin system objects. Each integer in the chromosome array will be the position of an object in an array of such objects. If the first integer in a chromosome array is 78 the first object encoded by the chromosome is the 79th element of the relevant array. What the objects are and which array each integer selects an object from forms the domain dependent elements of the coding.

The assignments produced by the modules all needed to be consistent. To ensure this, each of the objects used to produce an assignment keeps a record of its use in the sub population. The record is an attribute called the used at. The attribute is an array of integers. The array is the same length as the size of the sub population. The integer at each position in the array states the position of the object in an individual of the sub population. The individual is the one that occupies the same position in the population-individuals array as the integer, e.g. if the 46th integer in the array is 345 then the object is used by the 46th individual in the sub population and pointed to by the 345 integer of the individual's chromosome array. By recording the use of each individual a consistent assignment can be generated and then maintained by designing genetic operators that do not introduce inconsistencies. If consistencies were allowed the number of possible assignments would increase; the number would be the number of combinations not the number of permutations.

4.6.2 FITNESS FUNCTION

The fitness functions of all the modules have only one element common to all the GA modules. The common element is the alignment of peaks. The peaks all have a centre. The centre of a peak will be recorded in chemical shifts. The fitness function of all the GA modules determines the quality of the alignment of the chemical shifts of two or more peaks. The peaks will normally have a tolerance within which their chemical shifts are expected to fall. In some of the modules other factors are also used in determining the fitness of an individual. The fitness functions of the modules are quite diverse.

4.6.3 Genetic Operators

The genetic operators have only one factor that is common to all the GA modules. All the genetic operators are phenotypic genetic operators. There are two categories of genetic operators: genotype and phenotype operators. Operators in the first category are those used in the simple GA described in chapter 3. These operators function at the level of the genotype; they mutate the chromosome. The phenotype operators function at the level of the phenotype; they mutate the phenotype. They alter the phenotype of the individual causing a corresponding change in the genotype. Phenotypic genetic operators are used so that the assignments encoded by each individual remains consistent. The phenotypic operators alter the objects or position objects already used to create an assignment. When the objects or their position is altered their used at arrays are checked to insure that they are not already used in the individual. If they are used the mutation cannot take place or the object or objects must be removed from their existing position. The use of phenotypic operators allows the use of "smart" genetic operators. A smart genetic operator does not perform a random mutation it looks for the best mutation. The best mutation will either implicitly or explicitly use the criteria used by the fitness function to find the best possible mutation.

The next five chapters describe the domain dependent elements of each module in more detail. The domain independent elements described in this chapter are the same for all five of the GA modules.

4.7 GA Core Object Definitions

This section describes the objects of the GA core.

4.7.1 Chromosome Class Attributes

- Chromosome length attribute: an integer stating the length of the chromosome integer array. The attribute will be controlled by the input to the relevant module. The allocation of this attribute is to the class. This means that this attribute is stored once for the whole class.

- Chromosome bytes per residue attribute: defines how many integers are used to encode the assignment of an amino acid. The attribute will be defined by the user. It is used to calculate the chromosome-length attribute in conjunction with the input to the relevant module. The number is problem domain dependent. The attribute will be an integer and the allocation is to the class.

- Chromosome residue num attribute: states the number of amino acid residues in the protein under investigation. This will be derived from the amino acid sequence class that is found in all the modules, and is used to calculate the chromosome length attribute. The attribute will be an integer and the allocation is to the class.

- Chromosome read position attribute: this integer acts as pointer to an integer in the chromosome array. The attribute is used to read the chromosome sequentially. It is initially set to 0, this points to the first integer in the chromosome array. When an integer is read sequentially from the chromosome array the integer is incremented by 1, and then the position of the next integer to be read from the array. When the integer equals the chromosome length attribute it reset to 0. The sequentially reading functions are used either by the decoding functions or by fitness functions. The attribute will be an integer and the allocation is to the class.

- Chromosome array attribute: contains a one dimensional integer array. The length of the array is determined by the chromosome-length attribute. The array forms the chromosome of an individual. This array encodes a possible assignment and forms the genotype of an individual.

4.7.2 Individual Class Attributes

- Individual number attribute: this integer is the index[20] of the individual in an array of individual objects. The array will be the individuals attribute of a population object. For example, if the attribute is 345 the individual will occupy the 346th element[21] of a population individual attribute.

- Individual fitness attribute: this floating point number records the fitness of the individual. The fitness will be determined by the fitness function of the module and recorded in this attribute. Once determined this is used by the

[20] An index gives the position of an element of in an array.

[21] It is 346th element as the first element in the array has an index of 0.

GA core.

- Individual scaled fitness: this a floating point number records the scaled fitness of the individual (Sections 4.1 and 4.5.2). The individual-fitness attribute is scaled to give the scaled fitness.

- Individual parent attribute: this will be another individual object. The attribute is used by individual objects in the sub population to record the individual in the main population from which they were copied. This was once used as part of the crowding factor (Sections 4.1 and 4.5.2).

- Individual child number attribute: this integer records the whole number of children the individual is calculated as having in the next generation. For example, if it calculated that an individual will have 1.67 children in the next generation the attribute will be 1 (Sections 4.1 and 4.4.2).

- Individual fractional child attribute: this floating point number records the fractional number of children the individual is calculated as having in the next generation. For example, if it calculated that an individual will have 1.67 children in the next generation the attribute will be 0.67 (sections 4.1 and 4.4.2).

- Individual age attribute: this integer records the number of times an individual has been selected for reproduction in the main population. Every time the individual is selected for reproduction this attribute is incremented by one (sections 4.1 and 4.5.2). While undergoing reproduction in the sub population the age of an individual is reset to 0 (sections 4.1 and 4.4.2).

- Individual changed attribute: this symbol records when an individual has been changed by the genetic operators of the module it is in (section 4.4.2).

- Individual old age attribute: this integer states the old age for each individual (sections 4.1 and 4.5.2). The attribute is set by the user defined old age parameter. The allocation of the attribute is to the class.

- Individual death age attribute: this integer states the death age for each individual (sections 4.1 and 4.5.2). The attribute is set by the user defined death age parameter. The allocation attribute is to the class.

4.7.3 Population Class Attributes

- Population size attribute: this integer states the size of population, the number of individuals in the population.

- Population total fitness attribute: this floating point number is the sum of the fitness for all the individuals in the population. This is the sum of individual-fitness attributes for all the individuals in the population-individuals attribute of object.

- Population average fitness attribute: this floating point number is the average fitness of the individuals in the population.

- Population min fitness attribute: this floating point number attribute is the fitness of the individual with the lowest fitness in the population.

- Population max fitness attribute: this floating point number is the fitness of the individual with the highest fitness in the population.

- Population best individual attribute: this individual is the fittest individual in the population. The fitness attribute of the individual will be the population-max-fitness attribute.

- Population pointer attribute: this integer gives the position of an individual in the individuals attribute of the population object.

- Population individuals attribute: this is an array of individual objects. The length of the array will be defined by the population-size attribute.

4.7.4 Main Population Class Attributes

- Population current generation: this integer states the current generation of the main population. The attribute is incremented every generation that the population evolves.

- Population best ancestor: this is the fittest individual object that has existed in the population. At the start of the evolution of the population a blank individual, with negative fitness, is constructed to become the best ancestor. Every generation the fitness of the best individual in the main population is compared to the fitness of the best ancestor. If the best individual is fitter it is copied to the best ancestor individual.

4.7.5 Sub Population Class Attributes

- Population crossover number attribute: this integer states the number of pairs of individuals in the sub population that undergo crossover. The number is calculated from by multiplying the used defined parameter crossover rate by the size of the sub population and dividing by two. The number is then rounded down to the nearest integer.

- Population mutation num1 attribute: this integer defines the number of individuals in the sub population that undergo mutation by genetic operators which have their frequency set by the mutation rate 1 parameter. The number is calculated by multiplying the used defined parameter mutation rate 1 by the size of the sub population. The number is then rounded down to the nearest integer.

- Population mutation num2 attribute: this integer defines the number of individuals in the sub population that undergo mutation by genetic operators which have their frequency set by the mutation rate 2 parameter. The number is calculated by multiplying the used defined parameter mutation rate 2 by the size of the sub population. The number is then rounded down to the nearest integer.

- Population mutation num3 attribute: this integer defines the number of individuals in the sub population that undergo mutation by genetic operators which have their frequency set by the mutation rate 3 parameter. The number is calculated by multiplying the used defined parameter mutation rate 3 by the size of the sub population. The number is then rounded down to the nearest integer.

- Population crowding factor attribute: this integer states the number of individuals in the main population compared to an individual from the sub population to find a candidate for replacement. The attribute will contain the user defined crowding factor parameter.

- Population store attribute: this is an array of individuals. Its size is defined by the population-size attribute of the sub population. When the select sub population function is selecting individuals from the main population the individuals are placed in the array. The array stores the individuals of the sub population before reproduction takes place..

5.0 Two Dimensional Sequential Assignment Module

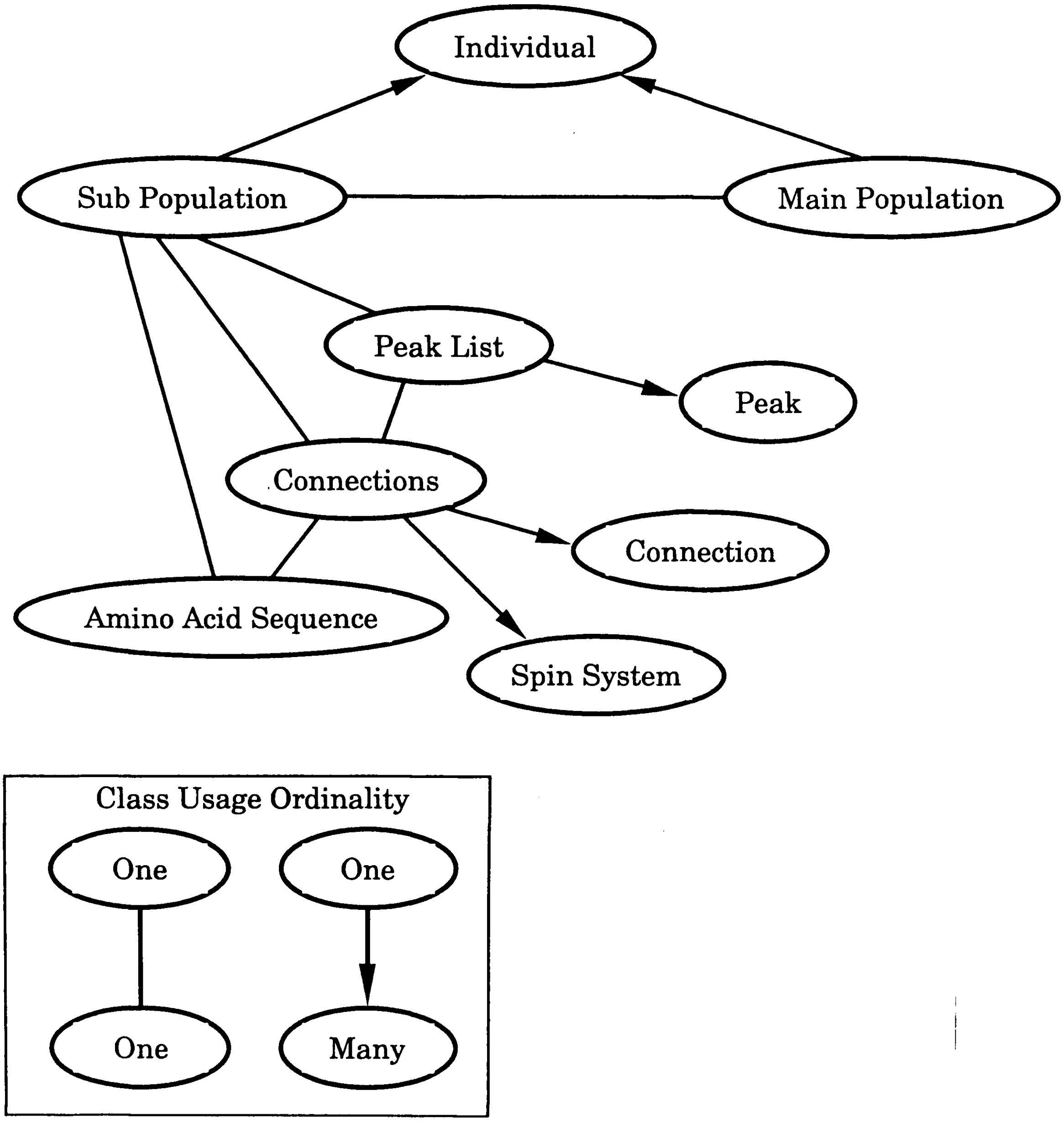
The two dimensional sequential assignment module (2D-SAM) is a GA that takes the amino acid sequence, spin system identification and NOESY spectrum of a protein and produces a sequential assignment. The production of a sequential assignment from the elements listed above is essentially a travelling salesman problem. The identified spin systems have to be placed in the optimal sequence according to several criteria.. The 2D-SAM was originally conceived as part of a collaborative project in which the spin system identification module would be designed and implemented by a student in Aberdeen.

The sequential assignment of a protein is described in section 2.2. The amino acid sequence determines where a spin system can appear in a sequential assignment. If the amino acid at position 137 is a proline then only a spin system identified as being generated by a Proline can go at position 137. The spin system identification process (manual or automatic) supplies the chemical shifts of each spin system and the type of amino acid that generated it. The identification of the type of amino acid may be ambiguous, and thus the spin systems might have multiple identities, e.g. a spin system may be identified as either a Leucine or an Isoleucine. This increases the number of positions at which a spin system can be placed in the amino acid sequence. The NOESY spectrum, in the form of a peak list, supplies the information required to link the spin systems together. The spin systems are linked to each other by between zero and four NOESY cross peaks. There will be several thousand cross peaks in a NOESY spectrum, but for any two spin systems there will normally be only a small number of cross peaks in the NOESY spectrum that can possibly act as a link between them. This creates a large number of possible permutations (the relevant number is the number of permutations not combinations as each spin system or cross peak can be used only once).

5.1 Design of the 2D-SAM

The design of the 2D-SAM was optimised over a period of months of development and testing of a number of prototypes. The most successful design is described in this section. As described in Chapter 4 the design of the 2D-SAM is an object oriented one, based around the same GA core as the other modules. The classes and their usage can be seen in Figure 5.1. The core part of the GA is made up of the individual, sub population and main population classes. The problem specific part is contained in the amino acid sequence, connections, peak list, connection, peak and spin system classes. The sequence

Figure 5.1 The design of the 2D-SAM.



and peak list objects model the amino acid sequence of the protein and its peak list[22]. The peak list is derived from the NOESY spectrum of the protein. The peak object models the individual NOESY peaks. The peak list object will contain a list of all the peak objects. The spin system objects each model a spin system and collectively model the spin system identification. A connection object models the inter-connections of a spin system object and duplicates some of its attributes. The connections object lists all the connection objects and maintains an array of arrays that records all the connection objects that can be placed at each position in the sequential assignment.

The rest of the design section is split into 3 sections: coding, fitness function and problem specific genetic operators, i.e. those factors that differ between the GA of the various modules.

5.1.1 Coding of 2D-SAM

There are two inter-related factors to be considered in designing a method for the coding: the chromosome representation and the method of converting the chromosome to a solution, in the case of the 2D-SAM a sequential assignment. The most intuitive chromosome representation for a sequential assignment also proved to be the most effective, an integer array. Each integer in the array defines either a peak or a spin system by stating their position in an array of peaks or spin systems. The first integer in an array encodes the first spin system in a sequential assignment. The next four integers define the peaks that link the first spin system to the second spin system. The second spin system is encoded by the sixth integer in the array. This pattern is repeated for the entire length of the integer array. An example of the coding can be seen in Figure 5.2. The pattern would be repeated for each connection in the protein's sequential assignment. The size[23] of the integer array would be the number of amino acids in the protein multiplied by five.

The objects in Figure 5.1 are used to encode a sequential assignment into a individual object. Each individual object will have an integer array or chromosome. Each integer of the array will define either a connection or a peak object. The process of creating a sequential assignment from a individuals chromosome is shown in Figure 5.3. An integer is converted to a

[22] The peak list contains the peaks picked out by a spectroscopist or a peak picking program. In this case the peaks would have been picked from the protein's NOESY spectrum, see section 2.1.3.

[23] Number of integers in the array.

Figure 5.2 The expression of the 2D-SAM integer chromosome to produce a sequential assignment.

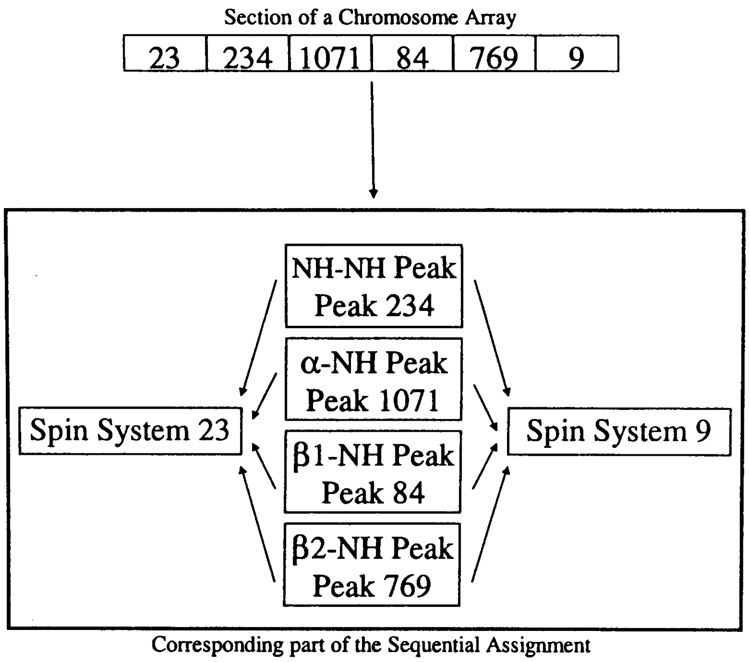
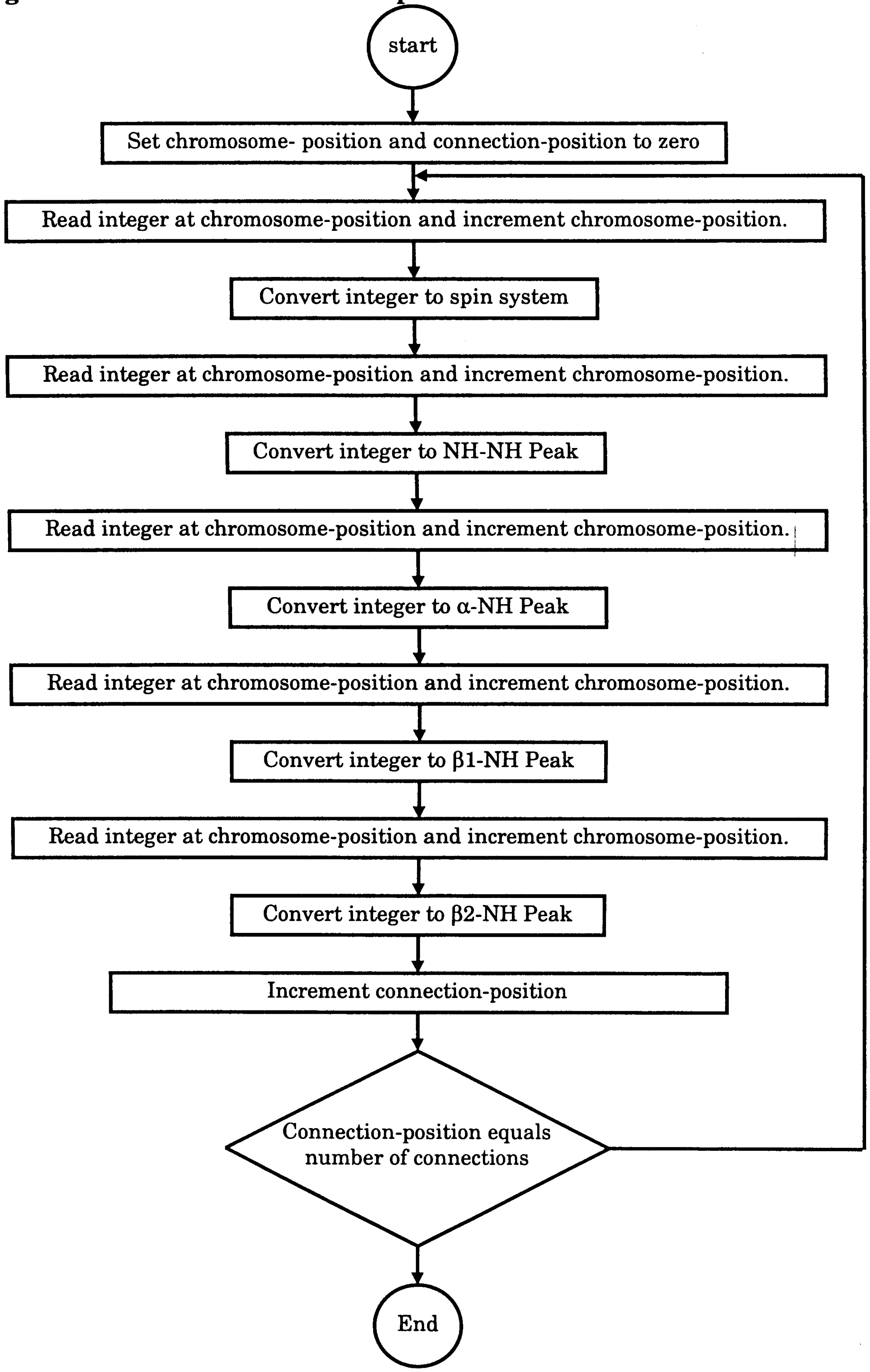


Figure 5.3 Flow chart of the expression of the 2D-SAM.



start
Set chromosome- position and connection-position to zero
Read integer at chromosome-position and increment chromosome-position.
Convert integer to spin system
Read integer at chromosome-position and increment chromosome-position.
Convert integer to NH-NH Peak
Read integer at chromosome-position and increment chromosome-position.
Convert integer to α-NH Peak
Read integer at chromosome-position and increment chromosome-position.
Convert integer to β1-NH Peak
Read integer at chromosome-position and increment chromosome-position.
Convert integer to β2-NH Peak
Increment connection-position
Connection-position equals number of connections
End

connection using the connection-arrays attribute of the connections object. The connection-position variable is used to find the correct array and the integer from the chromosome defines the position of the connection in the array. An integer is converted to a peak using the peak array attribute of the peak list object. The integer defines the position in the array of the peak to be selected.

For the initial genotypes the conventional approach would be to generate a random integer array as the chromosome of each individual. This would create inconsistent sequential assignments, which would increase the size of the search space (Section 3.2.4) dramatically by allowing combinations not just permutations. This would cause the GA to waste time evaluating impossible solutions. There are two ways to avoid this. The simplest is to include a penalty function into the fitness function. The penalty function reduces the fitness of an inconsistent individual. The level of the penalty function is difficult to set, e.g. should the penalty be set to be proportional to the level of inconsistencies or is it set to heavily penalise any inconsistent individual, making inconsistency a lethal characteristic for the individual. The more complex solution is to make inconsistent individuals impossible. Consistent individuals are produced initially and the genetic operators are designed not to introduce inconsistencies. The complex solution proved to be the more effective as it kept the search space smaller, even though there is a penalty in the time taken to evolve each generation.

Consistent individuals are produced by keeping a record of the use of each object used to construct a sequential assignment, connections and peaks. The record is the used-at attribute of the peak and connection objects. Whenever an object is placed in an individual its position is checked to ensure that it is not already used in that individual. The construction of the initial genotypes is thus only partly random. A connection is chosen at random from the list of connections that can be placed at that point. If the connection has already been used then another is randomly chosen. The process is repeated until an unused connection is found or if there are no unused connections the position is left blank. Once the connections have been chosen the peaks that link them are chosen.

If the peaks were randomly chosen then any peaks could be placed in any linking position. However only a small number of the peaks in a spectrum, if any, will link any two connections. To link two connections a peak must be in the correct place in the spectrum; the chemical shifts of the peak must align with the chemical shifts of the connections. To reduce the time spent evaluating highly improbable connection-peak-connection links all the probable links are found before the 2D-SAM starts evolving. These 'probable' links are used in creating the initial assignment encoded into an individual.

The probable links for a connection are found by finding all the peaks that could have been partly generated by the object, i.e. those peaks that have a chemical shift that aligns with the chemical shift of the connection, within a certain tolerance[24]. For each peak found all the connection objects that have a chemical shift alignment with its other chemical shift are found, again within a certain tolerance. By finding all the peaks a connection object is linked to and then all the other connection objects that link to each of the peaks all the connection-peak-connection links are found (Figure 5.4). For each connection object 4 lists are created: NH-NH list, α-NH list, β1-NH list and β2-NH list. The list in which each connection-peak-connection link appears depends on the chemical shift of the connection with which the peak aligns. The lists form the NN-connections, a-connections, B1-connections and B2-connections attributes of the object. Each element of the list has two parts: a peak and a connection, forming the second and third elements of the connection-peak-link. Each peak or connection can appear more than once in the list, i.e. a peak could potentially link several connections and two connections could potentially be linked by several peaks (Figure 5.4). Once all the 'probable' links for each connection have been found the initial assignments can be created.

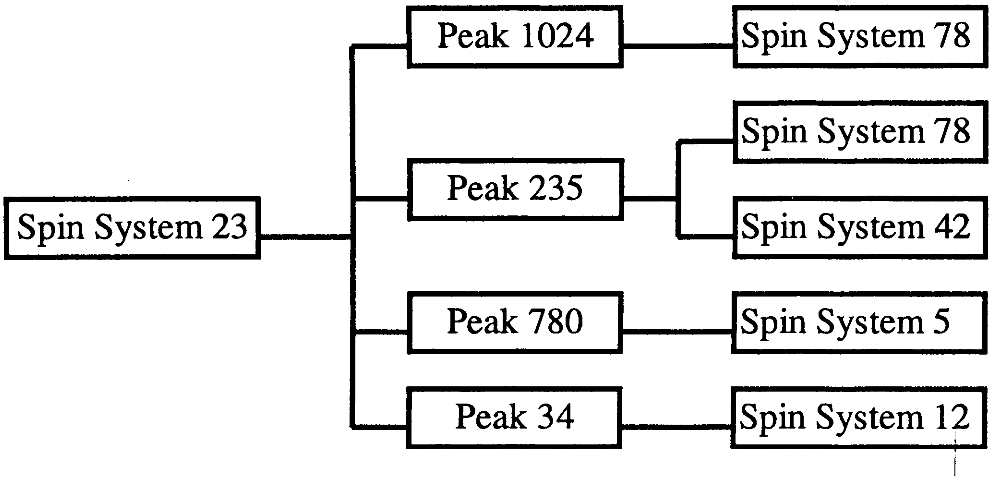
The sequence of connections in each individual is randomly selected. A connection is then randomly chosen in the sequence of connections encoded by an individual. The NH-connections attribute of the connection is examined for entries that contain the next connection in the sequence. The first entry that matches the search criteria is examined. If the peak is unused its index is placed at the connection's NH-NH link position in the chromosome (Figure 5.2). If it is already used the next of the selected entries is examined. If no unused peaks are found or there are no entries selected the position is left blank. The process is repeated for the a-connections, B1-connecions and B2-connections attributes of the connection and then for each attribute of the other connection objects.

The result of the above procedures is an initial population composed of individuals that are random sequences of connections and peaks. The individuals are consistent and contain no wildly improbable connection-peak-

[24] The tolerance is designated by the user.

Figure 5.4 Multiple spin system connections.

A spin system's connection to other spin systems



connection links. The individuals of the population remain consistent throughout the running of the GA. The consistency of the GA population, once it is running, depends on its genetic operators. It is possible that improbable connection-peaks-connection links will occur but the genetic operators and selection pressure should keep the number small. The selection pressure arises from the fitness function ascribing zero fitness to improbable links. The fitness function produces a number that represents the fitness of the individual.

5.1.2 FITNESS FUNCTION OF2D-SAM

The fitness function determines the fitness of an individual; the quality of the solution it represents. In the case of the 2D-SAM the solution is a sequential assignment and the fitness is the probability of the connections being in the right sequence. The position which a connection occupies in the sequence should be dependent on two things: the amino acid identity of the connection and the peaks that link it to the neighbouring connections.

Spin system identification can be ambiguous. The spin system identification may state that a connection may have been generated by any of several amino acid types. When a connection has multiple possible identities it can be placed at any position where any one of these amino acid types occurs in the amino acid sequence. Each amino acid identity of a connection has an associated probability, e.g. the amino acid is probably a glutamate but it could be a glutamine. The probability will be a number between 1.0 and 0.0, e.g. glutamate 0.8 and glutamine 0.3. The probability will be assigned by the program that performs the spin system identification or by a spectroscopist. A connection is more likely to be in the correct position when it is a position which uses a high probability identity. Using the previous examples the connection is more likely to be in the correct position if it is in a position where it is a glutamate, 0.8, rather than a glutamine, 0.3.

A connection is linked to its neighbours by between 0 and 8 peaks. The more NOE peaks observed that link a connection to its neighbours the greater the probability they are sequential. The probability that each peak actually links the 2 connections also affects the probability of the connections being sequential. The probability of a connection object being in the right position in the sequential assignment will depend on the probability of its amino acid identity being correct as well as its links to the neighbouring connection objects. A combination of these probabilities for each connection object in a sequential assignment is used to assess the fitness for the sequential assignment. The probabilities cannot be calculated. They can however be estimated. These estimates are referred to in the following sections as scores.

The fitness function takes an individual and decodes or expresses the solution it represents. The fitness is the sum of the scores for each connection. The score of each connection, *Score*, is calculated from three factors: the identity score $spin_{id}$, the score of the preceding NOE links nOe_p and the score of the succeeding NOE links nOe_s. The calculation is shown in Equation 5.1. The identity score and the NOE score, the sum of both the preceding and succeeding NOE score, are equally important. The multiplication of the two scores insures no distortions occurs. For example if the identity score is 0.001 and the NOE score is 0.95 then the score for the spin system is 9.5 x 10^{-4}, appropriate for the low identity score. Alternatively, if the scores were added together the score would be 0.951, inappropriate given the low identity score.

$$Score = spin_{id} \times \left(nOe_p + nOe_s\right) \qquad \text{Equation. 5.1}$$

The preceding and succeeding NOE scores are the sum of the individual link scores. There are 4 links between each connection, NH-NH, α-NH, β1-NH and β2-NH (equation 5.2).

$$nOe\_score = NH\_NH + \alpha\_NH + \beta 1\_NH + \beta 2\_NH \qquad \text{Equation 5.2}$$

The score of each link has to be calculated. Each link is calculated in the same way except different chemical shifts are used, e.g. for a α-NH link the α chemical shift of the first connection and the NH chemical shift of the second connection are used. The calculation of the score for a β1-NH link, is used as an example (Equation 5.3). The β1 chemical shift of the first connection, $CS_{\beta 1}$, and the NH chemical shift of the second connection, CS_{NH}, are found. Where these two chemical shifts overlap defines the ideal position for a peak to link the two connections. There will be two such positions created; one above and one below the diagonal of the spectrum. The two chemical shifts of a peak are found; in both the d1, CS_{d1}, and d2, CS_{d2}, dimension. The ideal position to which the peak is closest is used as the ideal point. The closer the peak is to the ideal point the higher the score. Beyond certain distances[25], $D1_{Max}$ and $D2_{Max}$, from the ideal point the peak scores zero. The distances are the tolerances used to determine the possible links for each connection described

[25] Each dimension needs a specific distance due to the variation in digital resolution in the two dimensions.

in section 5.1.1. The two distances, the peaks chemical shifts and the connections chemical shifts are used to calculate a score. The score will be between 0.0 and 1.0.

$$\beta 1 - NH - score = 1 - 0.5\left(\frac{|CS_{d1} - CS_{??}|}{D1_{\max}} + \frac{|CS_{d2} - CS_{??}|}{D2_{\max}} \right) \qquad \text{Equation 5.3}$$

The question marks denote that the connection chemical shifts used depend on which of the two ideal points is used.

The above equations are the ones currently used in the fitness function. Several variations of this equation were used. The first variation was the use of the square of the distances to calculate the score of an NOE link (Equation 5.4). This creates a geometric rather than linear relationship between distance from the ideal point and the score.

$$\beta 1 - NH - score = 1 - 0.5\left(\frac{|CS_{d1} - CS_{??}|^2}{D1_{\max}^{\ 2}} + \frac{|CS_{d2} - CS_{??}|^2}{D2_{\max}^{\ 2}} \right) \qquad \text{Equation 5.4}$$

This change in the relationship was intended to make the score conform more to the spectroscopists estimations. However there was no conclusive improvement in the performance of the 2D-SAM with this modification. As the Modification increased the complexity and decreased the speed of the fitness function it was abandoned.

Another modification was at the level of the calculation of the links between the two connections (Equation 5.2). The modification involved the introduction of a synergistic factor into the equation. For each link score above zero an amount was added to the synergistic factor, e.g. for each link score greater than zero 0.25 is added to the synergistic factor. The synergistic factor starts at 1.0 and if all 4 links are greater than zero then synergistic factor becomes 2.0. The synergistic factor is then multiplied by the sum of all 4 NOE links' scores. The synergistic factor was incorporated into the fitness function to allow for the greater certainty that a spectroscopist would ascribe to connections linked by several peaks as opposed to one even if the scores were the same, e.g. two peak scores of 0.5 are better than one score of 1.0.

A series of different synergistic factors were tried. Each using different increments for each NOE link above zero. The addition was even varied for the number of peaks, e.g. the synergistic factor would start at 1.0, the first NOE link scoring above zero would add 0.0, the second NOE link scoring above zero would add 0.07, the third NOE link scoring above zero would add 0.13 and the fourth NOE link scoring above zero would add 0.3. With four NOE links scoring greater than zero the synergistic factor would be 1.5. The synergistic factor proved to have a negative impact on the performance of 2D-SAM. Since it led to the anomaly that a peak that was the only link between two connections would add more to the solutions score if it appeared elsewhere. The peak could score 1.0 in the correct position but by being the fourth peak elsewhere scoring 0.001 it could add greater than 1.0 to the score through the synergistic factor. The use of synergistic factors was therefore abandoned. It is possible that a carefully designed moderate synergistic factor could have a beneficial impact on the performance of the 2D-SAM, but it is doubtful that the improvement would be great enough to warrant the effort involved.

5.1.3 Genetic Operators of 2D-SAM

The function of the genetic operators in the 2D-SAM is to create new consistent sequential assignments. The operators must not disrupt the initial consistency of the sequential assignments. There are five genetic operators in the 2D-SAM:

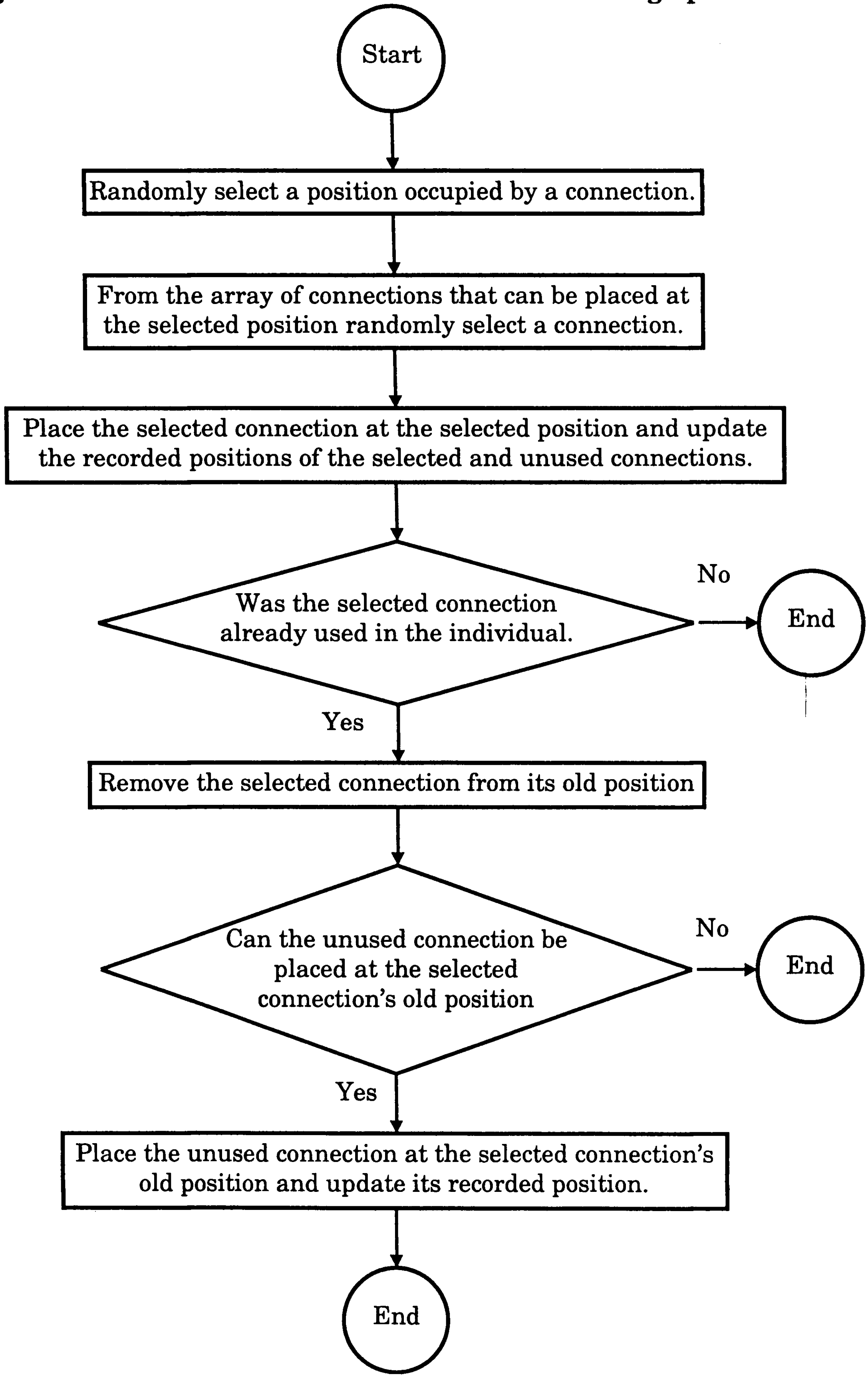
- Blank Removal
- Connection Reordering
- Segment Reordering
- Peak Reordering
- Crossover

The blank removal operator is used to keep the number of blank spaces to a minimum. As the other genetic operators function they occasionally introduce blank spaces into individuals. To offset this effect the blank removal operator is run. The operator fills in any blanks in the individuals sequential assignment with unused connections and peaks. A connection inserted at a blank space will be randomly chosen from the unused connections that can go at that position. A peak inserted at a blank position will be randomly chosen from the unused peaks that possibly link the two adjacent connections. The operator is used at a relatively low level. It will normally have only a small positive effect on the fitness of an individual. The other genetic operators could perform the same task but it would take a lot longer to produce the same effect. The chances of one of the other operators filling a blank position without disrupting the occupied positions is small.

The connection reordering genetic operator randomly reorders the connections in a sequential assignment, see Figure 5.5. The procedure outlined in Figure 5.5 keeps the individual's sequential assignment consistent and keeps as many connections in the sequential assignment as possible. Swapping the position of the two connections, if possible, is important for two reasons. The first is to ensure that, when the connections are swapped, the sequential assignment remains as complete as possible, i.e. there are as many peaks and connections in it as possible. The second reason is a little more complex. If two connections were each occupying the other's correct position any attempt to move them independently could result in a reduction of the fitness of the individual. The connections could be linked to their incorrect positions by a number of NOE peaks and thus contribute to the fitness of the individual. The deletion of one of the connections to place the other in its correct position would remove the contribution of the deleted connection to the overall fitness of the individual. The connection now in the correct position would probably increase the amount contributed to the overall fitness of the individual but it may not be enough to offset the loss of deleted connection. It is possible that the next time the individual underwent a reordering of its connections the deleted connection could be inserted into its now vacant correct position but the probability is small, given by the inverse of the number of connections in the sequential assignment multiplied by the inverse of the number of connections that can be placed at that position. Therefore swapping rather than independent movement of connections can be beneficial. When two connections are reordered the NOE peaks that link them to the neighbouring connection are also reordered. This is discussed later in the section describing the peak reordering operator.

The segment reordering operator swaps segments of sequential assignment in the chromosome of an individual. The operator is designed to swap segments of sequential assignment that can occur at two or more positions. Take for example a protein with two segments of amino acids Gly-Ala-Ser. If the two segments of connections that correspond to the two amino acid sequences are well linked by NOE peaks but are in the wrong positions then, without the segment reordering operator the segments would remain at their current positions. There are two reasons for this; the first is that the sequence of

Figure 5.5 Flow chart of the connection reordering operator.


Start
Randomly select a position occupied by a connection.
From the array of connections that can be placed at the selected position randomly select a connection.
Place the selected connection at the selected position and update the recorded positions of the selected and unused connections.
Was the selected connection already used in the individual.
No
End
Yes
Remove the selected connection from its old position
Can the unused connection be placed at the selected connection's old position
No
End
Yes
Place the unused connection at the selected connection's old position and update its recorded position.
End

operations required to swap the two segments by the connection and peak reordering operators has a low probability of occurring. The second is that any attempt to swap the segments of sequential assignment piecemeal, i.e. using the connection and peak reordering operators, would initially disrupt the segments, causing a reduction in the fitness of the individual. The sequence of operations would be working against the selection pressure of the 2D-SAM. The two factors combine to make the swapping of the segments improbable even though the final result would be a better sequential assignment and thus a fitter individual.

The segment reordering operator has two distinct phases, first to find segments that can be swapped and secondly to swap them. A flow chart of the operator is shown in Figure 5.6. The check to see if two connections can swap position is dependent on the amino acids that correspond to both positions and the amino acid identities of the two connections, the connection must have the right amino acid identity to be placed at the new position. When the two segments are swapped all the connections of the segment and all the NOE cross peaks that link them are swapped. The NOE links to the swapped segments are then reordered. This is discussed later in the section describing the peak reordering operator.

The crossover operator is similar to the segment reordering operator. The operator swaps segments of sequential assignment between individuals as opposed to within an individual. The function of the operator is shown in Figure 5.7. The swapping of the connections and peak objects is itself a complex procedure. The complexity is required to ensure that the two sequential assignments encoded by the individuals undergoing crossover remain consistent. The process of swapping two connections is shown in Figure 5.8; it is essentially the same as that of swapping two peaks.

The peak reordering operator (Figure 5.9) will reorder the sequence of peaks in the sequential assignment an individual encodes. The operator does not reorder the peaks completely randomly. The operator randomly selects a peak in the sequential assignment and replaces it with another peak, randomly chosen from all the peaks that could possibly link the two adjacent connections. The connection attributes, NN-connections, a-connections, B1-connections and B2-connections, of a connection object contains a list of all the peaks that can link the connection to another connection.

The peak reordering operator is also used by the crossover, connection

Figure 5.6 Flow chart of the connection reordering operator.

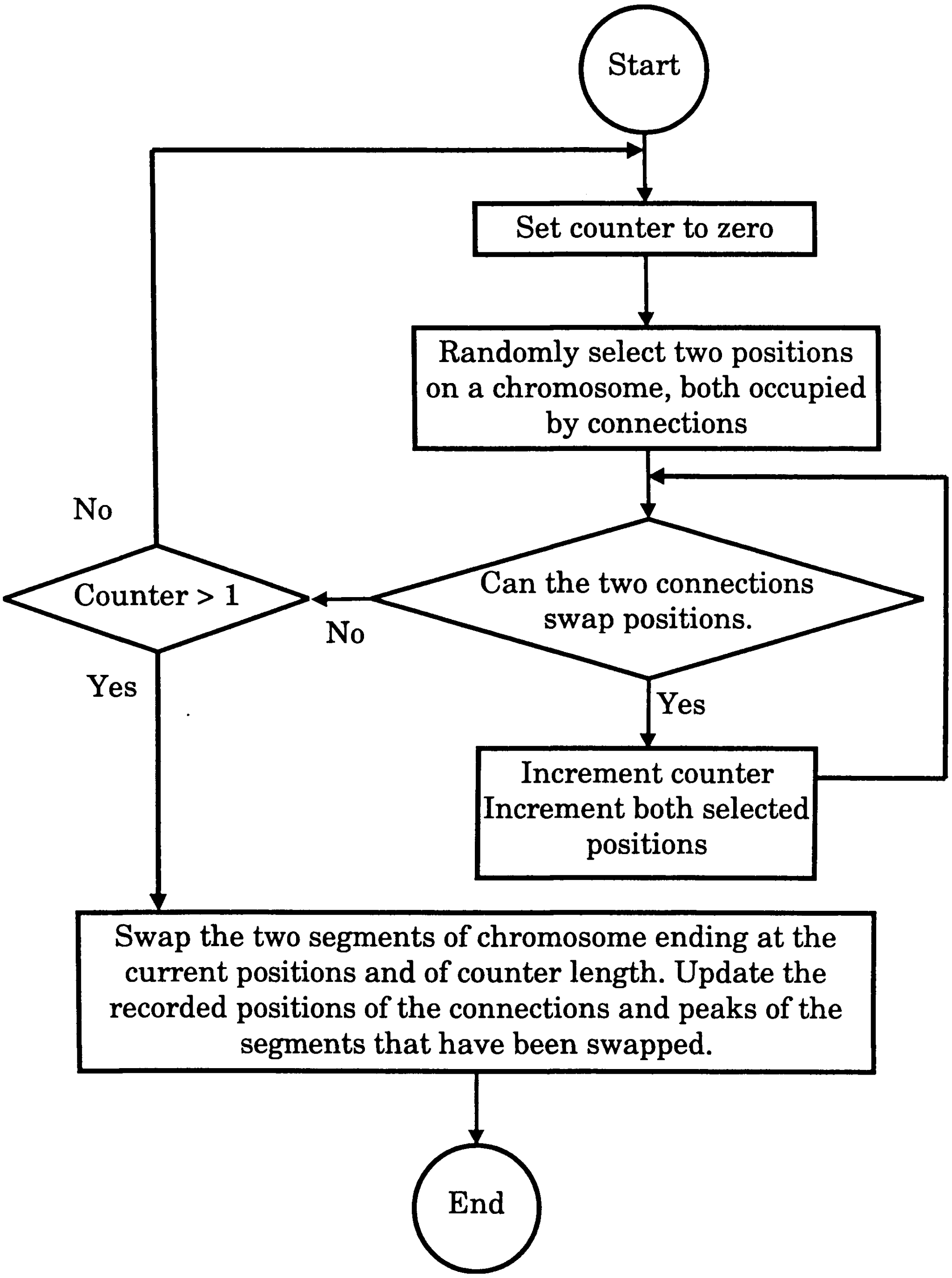

Start
Set counter to zero
Randomly select two positions on a chromosome, both occupied by connections
Can the two connections swap positions.
No
Yes
Increment counter Increment both selected positions
Counter > 1
No
Yes
Swap the two segments of chromosome ending at the current positions and of counter length. Update the recorded positions of the connections and peaks of the segments that have been swapped.
End

Figure 5.7 Flow chart of the crossover operator.

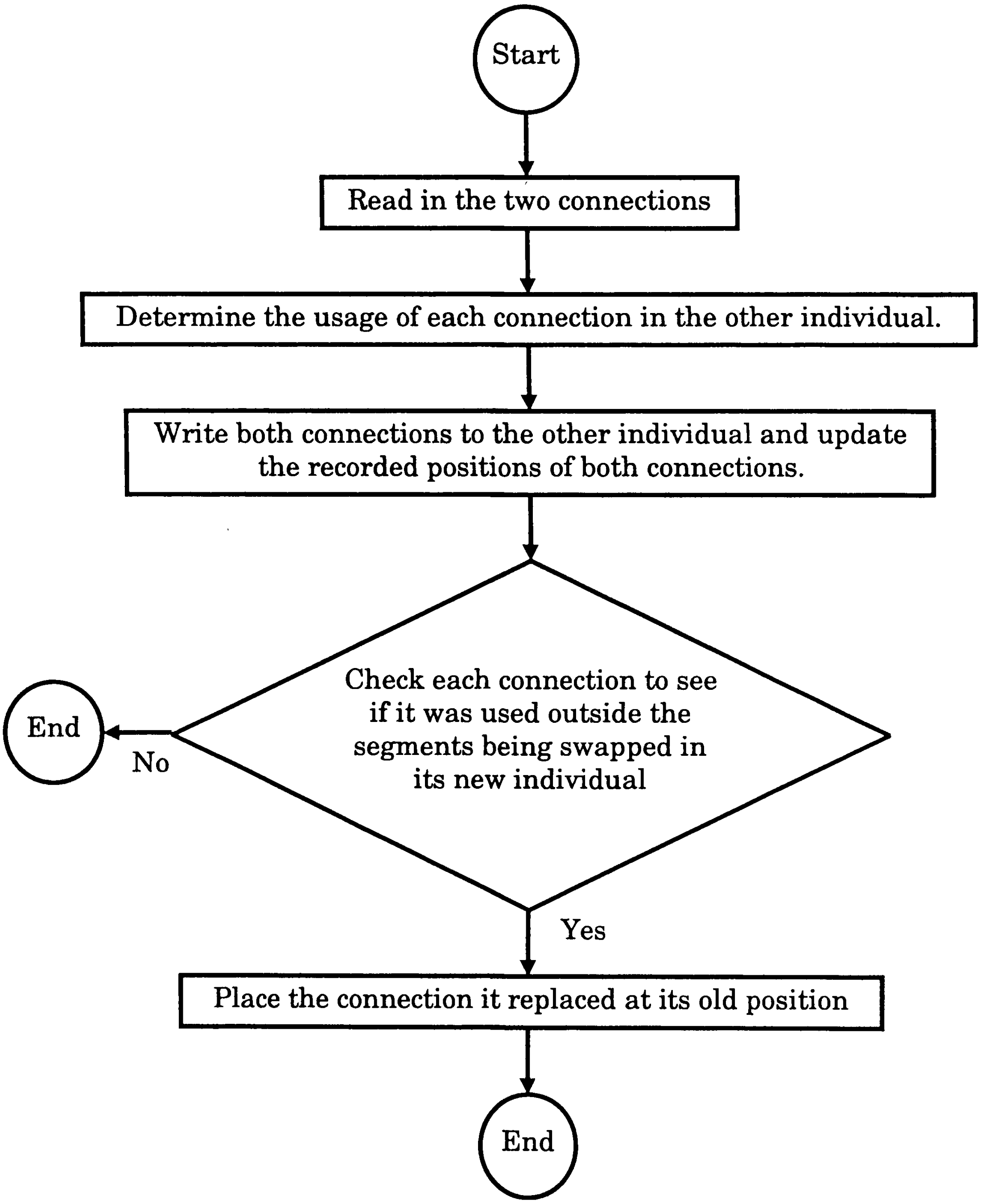

Start
Read in the two connections
Determine the usage of each connection in the other individual.
Write both connections to the other individual and update the recorded positions of both connections.
Check each connection to see if it was used outside the segments being swapped in its new individual
No
End
Yes
Place the connection it replaced at its old position
End

Figure 5.8 Connection reordering (see Figure 5.7).

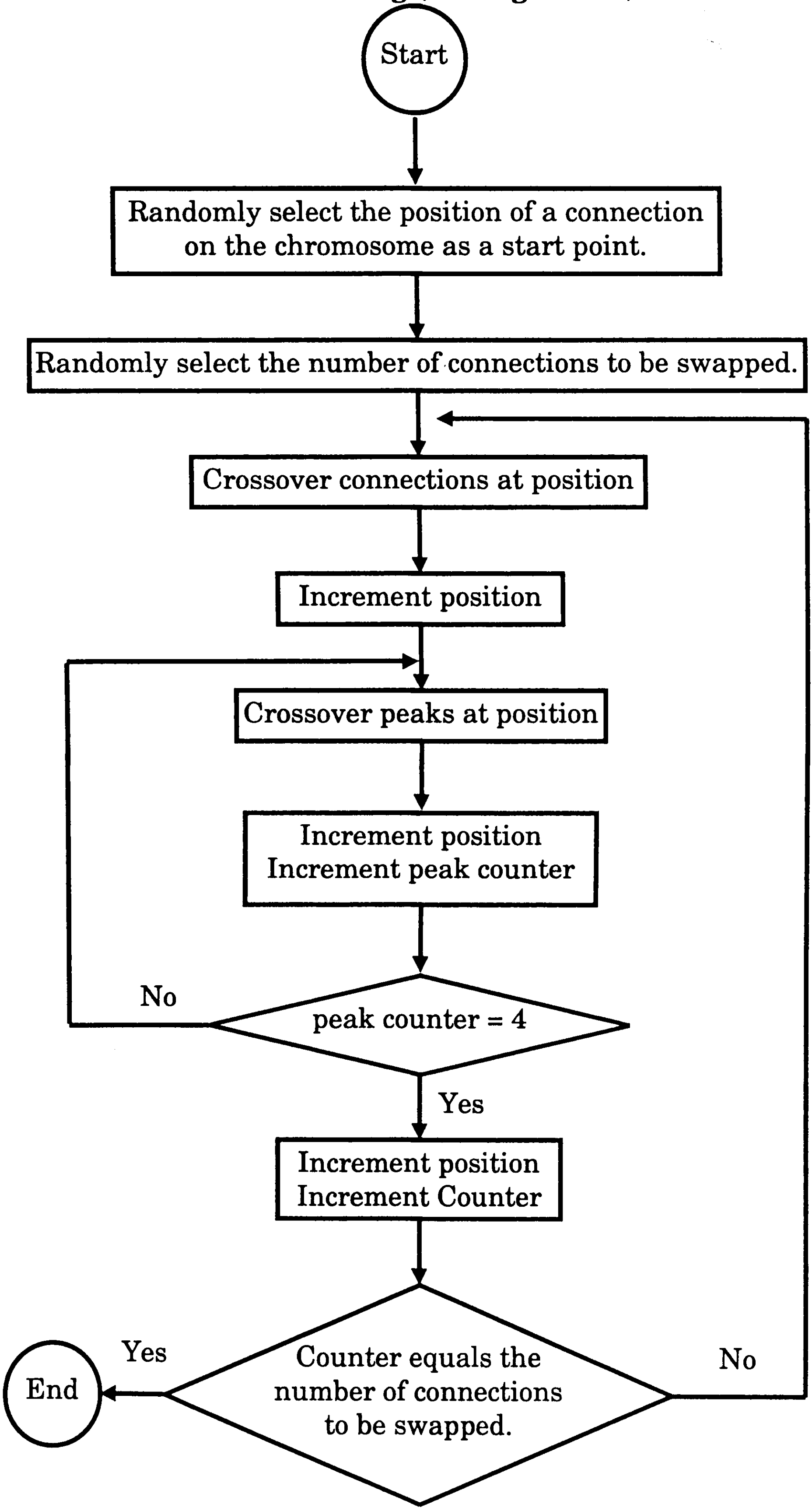
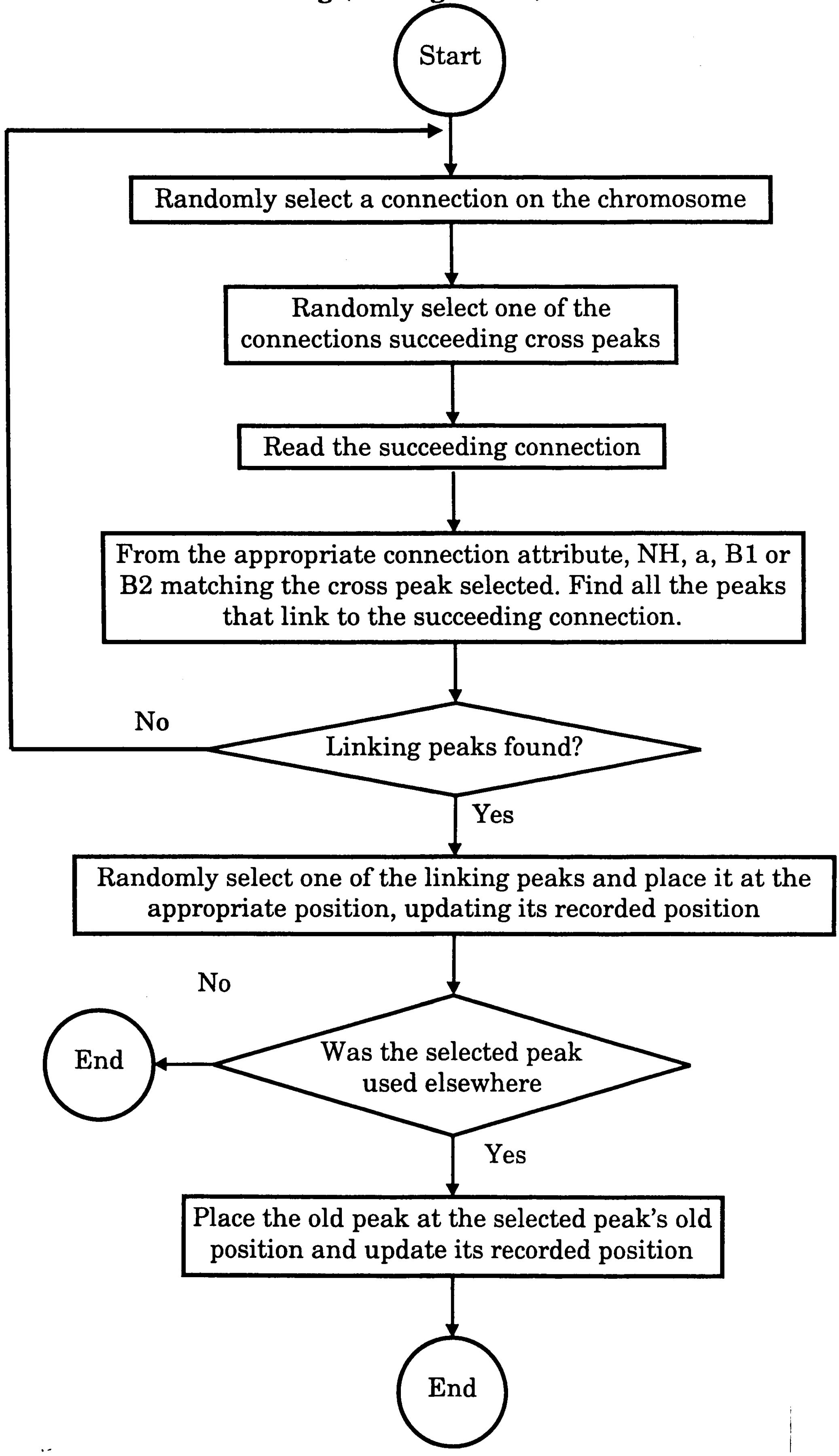

Start
Randomly select the position of a connection on the chromosome as a start point.
Randomly select the number of connections to be swapped.
Crossover connections at position
Increment position
Crossover peaks at position
Increment position
Increment peak counter
peak counter = 4
No
Yes
Increment position
Increment Counter
Counter equals the number of connections to be swapped.
Yes
No
End

Figure 5.9 Peak reordering (see Figure 5.7).


Start
Randomly select a connection on the chromosome
Randomly select one of the connections succeeding cross peaks
Read the succeeding connection
From the appropriate connection attribute, NH, a, B1 or B2 matching the cross peak selected. Find all the peaks that link to the succeeding connection.
No
Linking peaks found?
Yes
Randomly select one of the linking peaks and place it at the appropriate position, updating its recorded position
No
End
Was the selected peak used elsewhere
Yes
Place the old peak at the selected peak's old position and update its recorded position
End

reordering and segment reordering operators. It is used systematically to link the new connection or segments of sequential assignment to the existing sequential assignment. The four peaks before and after the new connection or segment are unlikely to be possible links between their adjacent connections. To insure that the peaks are possible links the reordering operator is used to insert possible linking peaks into the eight positions.

All the genetic operators were developed to accelerate the speed and reliability with which the 2D-SAM comes to a near optimal solution (Section 3.2.4 and Chapter 4).

5.2 EVALUATION OF THE 2D-SAM

The 2D-SAM was evaluated using the spectrum of an antibody binding domain of protein G. The domain is small, 60 amino acids in length, and its sequential assignment is known (38). The peak list of the NOESY spectrum was also available. The protein was relatively simple to assign given the quality of its NOESY spectrum and its small size. The 2D-SAM should assign at least 80% of the domain to be considered a success, although it is working from only one spectrum instead of several used by a spectroscopist. The 2D-SAM was also tested with Dihydrofolate Reductase (DHFR). The spin system identification is known (39) and a peak list was also available. DHFR is more of a challenge to the 2D-SAM, it is a large protein, 162 amino acids in length, and even experienced spectroscopists could not completely assign the protein from its 2D ^{1}H-^{1}H spectra alone, ^{15}N and ^{13}C labelling being required for full assignment. Approximately 35% of the protein was assigned using 2D spectra (GCKR personal communication).

The input data for the 2D-SAM can be found on the enclosed CD-ROM.

5.2.1 TESTING OF THE 2D-SAM

A number of test runs were performed to evaluate the performance of the 2D-SAM, in which the correctness and precision of the spin system identification was varied. The precision of the spin system identification refers to the number of possible amino acid identifications for a given spin system. The probability associated with each amino acid identity also has an impact on precision; for example when a Ser spin system is identified as either Ser 1.0 or Val 0.1 it is a more precise identification than when identified as either Ser 1.0 and Val 0.9. A spin system identification is considered to be correct when the correct amino acid identity has a probability greater than or equal to the best of the incorrect identities. Four sets of experiments were performed. The first was with a precise and correct spin system identification. The second and third sets of experiments had a correct, but imprecise spin system identification. The fourth set of experiments had an incorrect and imprecise spin system identification. The 2D-SAM is a stochastic program and its output will vary even if the input remains constant; in practical use it would be necessary to run the 2D-SAM several times in order to have confidence in the results.

The 2D-SAM has a number of user defined parameters. Some parameters are concerned with the spectrum, others with the working of 2D-SAM. The spectral parameters are the tolerances used in determining when a peak could possibly link two connections. All the other parameters affect the environment in which the individuals evolve. The parameters set for each experiment are those listed below except where stated otherwise. The parameters were all derived by experience (Sections 3.1.6). The default parameters are

- The population parameter is 400. The main-population has 400 individuals; the sub-population has 40 individuals. The larger the population the better the search space is sampled and the longer the time required to evolve one generation.

- The generations parameter is set at 6000. The sub-population is evolved for 6000 generations; effectively 600 generations for the main-population.

- The crossover rate parameter is set at 0.5. 50% of the individuals in the sub-population undergo crossover every generation.

- The connection reordering rate is set at 0.2. 20% of the individuals in the sub-population undergo connection reordering.

- The segment reordering rate is set at 0.1. 10% of the individuals in the sub-population undergo segment reordering.

- The peak reordering rate is set at 0.1. 10% of the individuals in the sub-population undergo peak reordering.

The fittest individual that evolved during the running of the 2D-SAM was taken as the solution. The experiments were run on a number of SGI workstations. The first run of the first set of experiments took 1hr 26mins on a workstation with a 250 MHz R4400 CPU and 128 Mb of RAM. The experiments were all run in a low priority batch queue. Thus the speed of a run will depend on the concurrent usage of the computer.

The first set of experiments was with the accurate and precise spin system identification. All the spin systems were accurately identified as being generated by one amino acid type. 24 experiments were performed. All the experimental runs produced a correct sequential assignment. There was some minor variation in the fitness of the best individuals produced in each run.

The second set of experiments was performed with an accurate spin system identification but with a reduced precision. The spin systems were all identified as being generated by a class of amino acid rather than an amino acid. The classes used were those described by Redfield (40). Each amino acid identity had a probability of 1.0 associated with it. Three groups of 24 experiments performed. The first set had the 2D-SAM parameters as the previous experiments, the second group had the generation parameter set at 8000 and the third group had it set at 10000. The results for the 6000 generation run are in Table 5.1. 4, of the 24 experimental runs 24 failed, to generate a correct sequential assignment. The experiments using the 10000 generation parameter failed to find the correct sequential assignment 2 out of 24 times. On 3 occasions when it did find the correct sequential assignment it was at generations above 6000 (6600, 7400 and 9400). Two of the errors involved the T6-T7 and T22-T23 spin systems being interchanged.

The third set of experiments were performed with an accurate spin system identification but with a reduced precision. The spin systems that had been identified by their Redfield classes in the previous set of experiments were identified as all the amino acids in the class. The correct amino acid identity had a score of 1.0 while the other amino acid identities in the class had a score of 0.9, 0.5 or 0.1 depending on the experimental subset being run. Each subset experiment was performed 24 times. The 0.1, 0.5 and 0.9 subsets produced 0, 1 and 2 incorrect sequential assignments respectively. The errors were the same errors as those observed in the second set of experiments.

The fourth set of experiments had the same precision as one of the previous subset of experiments; using a score of 0.5 for incorrect amino acid identities of the class. Errors were then introduced to the spin system identification by making a certain spin system have a score 0.5 for its correct identity and assigning one of its incorrect identities a score of 1.0. The changes were as follows K9R, E29R, Q37E, D41N and D51N, e.g. the spin system generated by

Table 5. 1 - Precision Reduced to Class Spin system identification

Experiment Number	Residues Correct	Percentage Correct	Fitness
1	60	100	213.6
2	60	100	213.1
3	60	100	213.1
4	60	100	213.1
5	60	100	213.6
6	56	93.3	210.4
7	60	100	213.1
8	60	100	213.1
9	60	100	213.8
10	60	100	213.1
11	60	100	213.1
12	60	100	213.1
13	60	100	210.7
14	60	100	213.1
15	56	93.3	210.4
16	46	76.7	196.9
17	60	100	213.1
18	60	100	213.1
19	60	100	213.1
20	60	100	213.1
21	60	100	213.1
22	60	100	213.1
23	60	100	213.1
24	57	95	209.1

a lysine at position 9 is misidentified as an arginine. 24 experiments were performed with each incorrect spin system identification. The results of the five of experiments can be seen in Table 5.2, Table 5.3, Table 5.4, Table 5.5 and Table 5.6.

In all the cases where there is an error in the sequential assignments of all four sets of tests the problem involves the T6 and T7 connections being swapped with those of T22 and T23.

The generation in which the correct sequential assignment was found for the above experiments is shown in Table 5.7 and Figure 5.10. The first set of experiments are recorded in the "Precise" column, the second set of experiments are recorded in the "Class" column and the third set of experiments are recorded in the 0.1, 0.5 and 0.9 identity columns. The number refers to the score assigned to the incorrect amino acid identities. The experiments are listed in decreasing precision left to right.

The fifth set of experiments was conducted using the 2D NOESY spectrum, spin system identification and amino acid sequence of DHFR. The parameters used were 12000 generations with a main population of 1200 individuals. The last run in the set of experiments took 14 hours 48 minutes to run. The results are far more variable than for protein G. The percentage SD of the percentage of residues in the correct position is 15%. Although the fitness only varies by 2.6%. The difference in the two figures is caused by a number of residues that have no NOE links to their neighbouring residues. These residues can then be in a number of positions, which may or may not be the correct positions, without affecting the fitness of the sequential assignment. The average result and standard deviation are calculated for this set of experimental runs. The 24 experiments gave an average of 26.9% correct sequential assignment. The results of all the experiments can be seen in Table 5.8.

5.2.2 CONCLUSIONS

The 2D-SAM exceeds the criterion of success decided upon before the module was started, namely that the program perform a sequential assignment that formed a useful start point for the spectroscopist in reasonable time. The criterion of a useful start point for protein G was a sequential assignment approximately 80% correct. The criteria of a useful start point for DHFR was a sequential assignment approximately 28% correct, 80% of the 35% of the assignment performed manually. For practical purposes a "reasonable time" was considered to be an overnight run on an SGI workstation, less than 18

Table 5. 2 - Precision Reduced to 0.5 Scoring Alternate Identity, K9R

Experiment Number	Residues Correct	Percentage Correct	Fitness
1	60	100	211.2
2	60	100	211.2
3	60	100	211.2
4	60	100	211.7
5	60	100	211.2
6	60	100	211.2
7	60	100	211.2
8	60	100	211.2
9	60	100	211.2
10	60	100	211.2
11	60	100	211.2
12	60	100	211.2
13	60	100	211.7
14	60	100	211.2
15	60	100	211.2
16	60	100	211.2
17	60	100	211.2
18	60	100	211.2
19	60	100	211.2
20	60	100	211.2
21	60	100	211.2
22	60	100	211.2
23	60	100	211.9
24	60	100	211.2

Table 5. 3- Precision Reduced to 0.5 Scoring Alternate Identity, Q37E

Experiment Number	Residues Correct	Percentage Correct	Fitness
1	60	100	210.2
2	60	100	210.7
3	60	100	210.2
4	60	100	210.2
5	60	100	210.2
6	60	100	210.2
7	60	100	210.2
8	60	100	210.2
9	60	100	210.2
10	60	100	210.7
11	60	100	210.2
12	60	100	210.2
13	60	100	210.7
14	60	100	210.2
15	60	100	210.2
16	60	100	210.2
17	60	100	210.2
18	60	100	210.2
19	60	100	210.2
20	60	100	210.2
21	60	100	210.2
22	56	93.3	207.5
23	60	100	210.2
24	60	100	210.2

Table 5. 4 - Precision Reduced to 0.5 Scoring Alternate Identity, E29R

Experiment Number	Residues Correct	Percentage Correct	Fitness
1	56	93.3	207.2
2	60	100	209.9
3	60	100	209.9
4	60	100	210.3
5	60	100	209.9
6	60	100	209.9
7	60	100	209.9
8	60	100	209.9
9	60	100	209.9
10	60	100	209.9
11	56	93.3	207.2
12	60	100	209.9
13	60	100	209.9
14	60	100	209.9
15	60	100	209.9
16	60	100	209.7
17	60	100	209.9
18	60	100	209.9
19	60	100	209.9
20	60	100	209.9
21	60	100	209.9
22	60	100	209.9
23	60	100	209.9
24	60	100	209.9

Table 5. 5- Precision Reduced to 0.5 Scoring Alternate Identity, D41N

Experiment Number	Residues Correct	Percentage Correct	Fitness
1	60	100	210.7
2	60	100	210.7
3	60	100	210.7
4	60	100	210.7
5	60	100	210.7
6	60	100	210.7
7	60	100	211.2
8	60	100	210.7
9	60	100	210.7
10	60	100	211.2
11	60	100	211.2
12	60	100	210.7
13	60	100	210.7
14	60	100	210.7
15	60	100	210.7
16	60	100	210.7
17	60	100	210.7
18	60	100	211.2
19	60	100	211.2
20	60	100	210.7
21	60	100	210.7
22	60	100	210.7
23	60	100	210.7
24	60	100	211.4

Table 5. 6- Precision Reduced to 0.5 Scoring Alternate Identity, D51N

Experiment Number	Residues Correct	Percentage Correct	Fitness
1	60	100	211.7
2	60	100	211.7
3	60	100	211.7
4	56	93.3	209.7
5	60	100	211.7
6	60	100	211.7
7	60	100	211.7
8	60	100	211.7
9	60	100	211.7
10	60	100	211.7
11	56	93.3	209
12	60	100	211.7
13	60	100	211.7
14	60	100	211.7
15	60	100	212.1
16	60	100	211.7
17	60	100	211.8
18	60	100	211.7
19	60	100	211.7
20	60	100	211.7
21	60	100	211.7
22	60	100	211.7
23	60	100	211.7
24	60	100	211.7

Table 5. 7 - Generation Correct Sequential Assignment Obtained

Experiment	Precise	0.1 Identity	0.5 Identity	0.9 Identity	Class
1	1000	3400	4400	2200	2000
2	1200	4000	2000	5400	2400
3	1600	1400	2600	2200	3200
4	1800	1400	2800	2200	2000
5	1000	3800	2600	4200	3000
6	1400	1600	2000	1400	2800
7	800	2200	2000	2000	3800
8	1000	2200	3200	3400	1800
9	1200	800	2400	2000	3400
10	1600	1400	1200	2600	5000
11	1400	1600	1600	3800	1400
12	2800	1200	3400	2200	4600
13	1200	1200	1200	2400	4400
14	1600	1000	1800	1000	4400
15	1400	1800	1600	3800	1400
16	1400	1400	4600	3200	2800
17	800	3800	2200	5000	3000
18	1400	2800	4800	2800	1400
19	1000	2400	4400	1200	5400
20	2200	1200	2000	4200	6000
21	1400	1600	2400	4200	6000
22	1200	2400	2800	1800	6000
23	2200	2000	2600	6000	6000
24	1600	2400	6000	6000	6000
Average	1425	2042	2775	3133	3675
Std. Dev. %	32.9	45.6	44.8	46.2	44.9

Table 5. 8 - DHFR

Experiment Number	Residues Correct	Percentage Correct	Fitness
1	41	25.3	169.8
2	54	33.3	183.2
3	44	27.2	172.5
4	43	26.5	172.6
5	57	35.2	171.4
6	34	21.0	171.2
7	37	22.8	179.3
8	47	29.0	173.7
9	43	26.5	169.1
10	39	24.1	176.0
11	38	23.5	174.8
12	35	21.6	166.7
13	50	30.9	172.4
14	46	28.4	173.0
15	40	24.7	165.9
16	36	22.2	169.3
17	45	27.8	180.6
18	45	27.8	164.1
19	52	32.1	173.5
20	50	30.9	171.4
21	44	27.2	169.5
22	51	31.5	171.9
23	34	21.0	173.7
24	39	24.1	175.8
Average	43.5	26.9	172.6
% Stan. Dev.	14.9	14.9	2.6

hours.

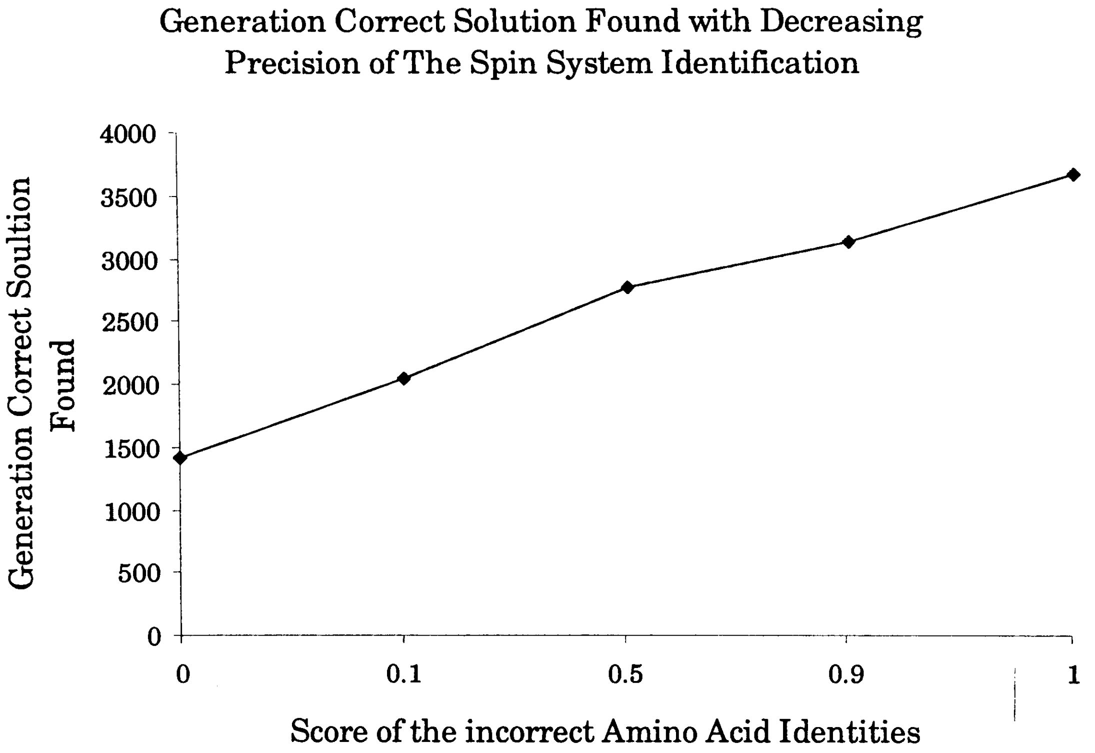
When a correct and precise spin system identification of protein G is used as input for the 2D-SAM it produces a correct sequential assignment as output. Given a good spin system identification, a good NOESY spectrum and a moderately sized protein the 2D-SAM can generate a good sequential assignment. This is demonstrated by the first set of experiments, in which the 2D-SAM produces a 100% correct assignment. The performance of the program exceeds the objectives for the module, working with only one spectrum it can produce a correct sequential assignment in about 1hour 30 minutes. This is impressive given that there are 1.930 x 108^{26} possible sequential assignments with the input used in the first set of experiments.

An example of a sequential assignment generated by the first run of the first experiment by the 2D-SAM can be see in Section 5.3.

In the second and third set of experiments the reduction in precision leads to an increase in the size of the search space. The spin systems can now be placed in a greater number of positions which increases the number of possible spin system sequences from 2.426 x 10^{38} to 4.575 x 10^{48}. The change in the precision increases the number of sequential assignments by 10 orders of magnitude to 3.640 x 10^{836}. The size of the search space is the same for each of these two sets of experiments, but there is a difference in performance in various sets of experiments (Figure 5.10).

The variation in performance is due not to the size of the search space but its shape. As can be seen in Figure 5.10, the performance of the 2D-SAM decreases on going from experiments where the wrong identities were ascribed a value of 0.1 to the class identity experiments where the wrong identities ascribed a value of 1.0. As the score of the incorrect identities is increased to effectively 1.0 in the class identity experiments then the incorrect areas of the search space have greater and greater fitness. This increase in fitness for these areas means that the 2D-SAM spends more time examining these areas of the search space and this decreases its performance. The 2D-SAM takes longer to arrive at a near optimal solution. In certain random cases the 2D-SAM will not find a near optimal solution in the time available to it. This is demonstrated by the fact that when the generations parameter was increased in the class experiments from 6000 to 10000 the number of near optimal solutions increased. The second and third set of experiments demonstrate that the 2D-SAM can work with an imprecise or ambiguous spin system identification. The

Figure 5.10 Effect of reducing assignment precision.



2D-SAM will on average take longer to reach a near optimal solution but it will reach it.

Despite an increase in the size of the search space by ten orders of magnitude and the reduction in the quality of the information supplied by the spin system identification the 2D-SAM still reached a correct sequential assignment the vast majority of the time. In the second set of experiments which were the most demanding, the 2D-SAM gave a 100% correct sequential assignment 83% of the time. On the four occasions it did not give a 100% correct sequential assignment it did give a sequential assignment greater that 93% correct on three of those experimental runs.

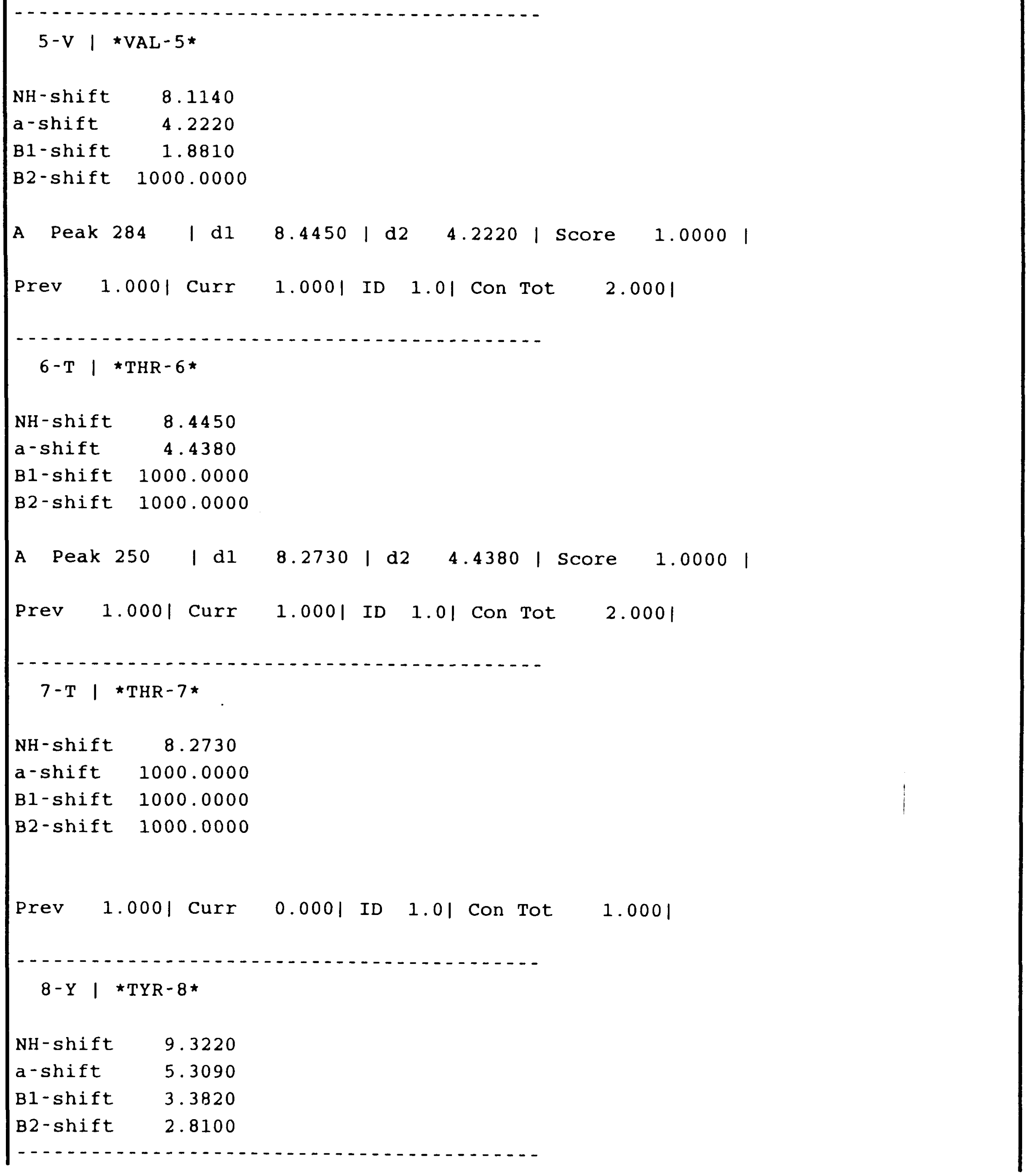
The fourth set of experiments demonstrate that the 2D-SAM is able to cope with a certain amount of error and imprecision in the spin system identification, providing that the quality of the NOESY spectrum is reasonable. The results were equivalent to the experimental runs performed with a correct spin system identification of the same precision. The errors that occurred in the sequential assignment produced were not related to the errors introduced into the spin system identification. The quality of the data derived from the NOESY spectrum was such that the errors in the spin system identification could be corrected by the 2D-SAM.

The most critical factor in affecting the performance of the 2D-SAM is the quality of the NOESY spectrum and spin system identification. An example of the spectrum effects is the recurring problem with the T6-T7 and T22-T23 segments of sequential assignment. These two segments cause a problem because the NOE spectrum can link them almost equally well to either position. The difference in fitness between the correct and incorrect positions for the two segments is about. 3.0. This is a less than 1.5% of the total fitness. Sample out put from the 2D-SAM showing the T6-T7 part of the sequential assignment with both correct and incorrect is shown in Figure 5.11. T22 is linked to V5 by a score of 0.783 and T23 to Y8 by a score of 1.0. The V5 is linked to T6 by a score of 1.0 and T7 is linked to Y8 by a score of 0.0. The T7 to Y8 is 0.0 because there are no NOE peaks that link the two spin system. The incorrect NOE links have a higher score. This is counteracted by the corresponding scoring in the T22-T23 region of the sequential assignment, to give a small benefit to the correct sequential assignment scoring.

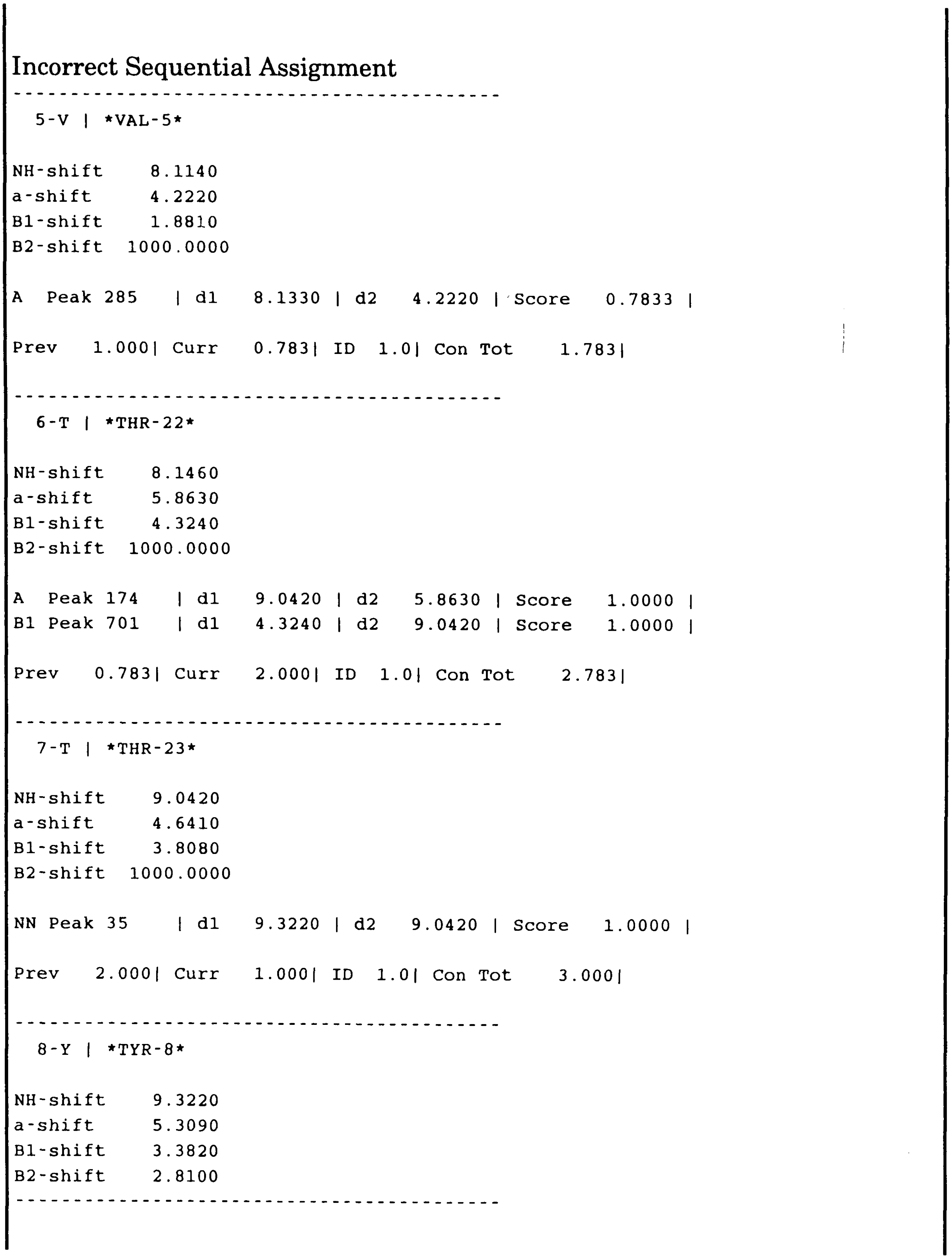
The fifth set of experiments uses DHFR, a much larger protein as a test. The size of the search space increases to 2.609 x 10^{2285} possible sequential

Figure 5. 11

Correct Sequential Assignment



```
-------------------------------------------
  5-V | *VAL-5*

NH-shift    8.1140
a-shift     4.2220
B1-shift    1.8810
B2-shift  1000.0000

A  Peak 284   | d1   8.4450 | d2   4.2220 | Score   1.0000 |

Prev   1.000| Curr   1.000| ID  1.0| Con Tot    2.000|

-------------------------------------------
  6-T | *THR-6*

NH-shift    8.4450
a-shift     4.4380
B1-shift  1000.0000
B2-shift  1000.0000

A  Peak 250   | d1   8.2730 | d2   4.4380 | Score   1.0000 |

Prev   1.000| Curr   1.000| ID  1.0| Con Tot    2.000|

-------------------------------------------
  7-T | *THR-7*

NH-shift    8.2730
a-shift   1000.0000
B1-shift  1000.0000
B2-shift  1000.0000


Prev   1.000| Curr   0.000| ID  1.0| Con Tot    1.000|

-------------------------------------------
  8-Y | *TYR-8*

NH-shift    9.3220
a-shift     5.3090
B1-shift    3.3820
B2-shift    2.8100
-------------------------------------------
```



Incorrect Sequential Assignment

```
-----------------------------------------
  5-V | *VAL-5*

NH-shift    8.1140
a-shift     4.2220
B1-shift    1.8810
B2-shift  1000.0000

A  Peak 285   | d1   8.1330 | d2   4.2220 | Score   0.7833 |

Prev   1.000| Curr   0.783| ID  1.0| Con Tot    1.783|

-----------------------------------------
  6-T | *THR-22*

NH-shift    8.1460
a-shift     5.8630
B1-shift    4.3240
B2-shift  1000.0000

A  Peak 174   | d1   9.0420 | d2   5.8630 | Score   1.0000 |
B1 Peak 701   | d1   4.3240 | d2   9.0420 | Score   1.0000 |

Prev   0.783| Curr   2.000| ID  1.0| Con Tot    2.783|

-----------------------------------------
  7-T | *THR-23*

NH-shift    9.0420
a-shift     4.6410
B1-shift    3.8080
B2-shift  1000.0000

NN Peak 35    | d1   9.3220 | d2   9.0420 | Score   1.0000 |

Prev   2.000| Curr   1.000| ID  1.0| Con Tot    3.000|

-----------------------------------------
  8-Y | *TYR-8*

NH-shift    9.3220
a-shift     5.3090
B1-shift    3.3820
B2-shift    2.8100
-----------------------------------------
```

assignments with the data used in the DHFR runs. The data supplied from the spectra is also of reduced quality due to greater peak overlap (Section 2.2.2), arising from the increased number of peaks in NOESY spectrum and their increased line width. Despite this dramatic increase in the size of the search space and decrease in the quality of the spectrum the 2D-SAM nearly meets the success criterion, 80% of the performance of a spectroscopist. The level of the manual assignment was 28% correct sequential assignment. The level achieved was a little under that at 26.9 %. The was a 14% standard deviation in the correctness of the assignment but only a 3% standard deviation in the fitness of the assignments. The difference between the two figures is due to the fact that a number of connections have no NOE peaks to link them into the neighbouring connections. Their positions are entirely random, within the constraints of their amino acid identity list. Therefore whether or not any of these connections are in the correct position and contributing to the percentage correct is random.

The program would still have been worth using as start point for manual sequential assignment of the DHFR spectrum. This was eventually completely assigned using ^{15}N and ^{13}C labelling, together with heteronuclear experiments. One of these experiments is used by the 3D-SAM to perform sequential assignment.

5.3 2D-SAM Object Definitions

This section contains a definition of the objects used in the design and implementation of the 2D-SAM. A list of the attributes of each object are listed below.

5.3.1 Peak Class

The peak class is designed to model cross peaks and their use. The attributes of the peak object are listed below:

- Id attribute: the symbol identifies the peak to the user. It is the number or symbol that was used to identify the peak in the text file that contains the peak list.

- System-id attribute: the symbol identifies the peak in the 2D-SAM.

- Type attribute: the symbol states the type of spectrum the peak is selected from.

- The chemical shift attributes d1 and d2; these floating point numbers give the chemical shift of the centre of the peak in dimensions one and two respectively.

- Used-at attribute: is an integer array of the same size as the sub population (Section 4.5). The array records the position of the peak object in the individuals of the sub population. The integer at position 5 will record the position of the peak object in the chromosome array of the 5th individual in the sub population. If the peak is not used in the 5th individual the integer is -1.

- Index attribute: the integer gives the position of the peak in an array of peaks. The array of peaks is one of the attributes of the peak list class.

There will be as many peak objects as there are peaks in the peak list input into the 2D-SAM.

5.3.2 Peal List Class

The peak-list class, is designed to model a peak list, is a relatively simple class with only 4 attributes.

- Type attribute: this symbol states the type of spectrum from which the peak list was selected from. In the 2D-SAM this will always be a 2D NOESY spectrum.

- Peak-number attribute: this integer states the number of peaks in the peak list.

- Peak-list attribute: this is a list of all the peaks, in the form of peak objects, in the peak list.

- Peak-array attribute: this is an array of all the peaks, in the form of peak objects, in the peak list. The position of a peak object in this array defines its index attribute.

There will be only one peak list object in the 2D-SAM as only one spectrum is used. The peak-list object will contain all the peak objects (Figure 5.1).

The spin system and connection classes are intertwined and could now be combined into one. They were originally separated to allow the 2D-SAM to interact with spin system identification module being developed by our collaborators at Aberdeen. The spin system object definition is the only non original code in the 2D-SAM, although the class definition has been added to. The spin system object code was deliberately retained to allow the 2D-SAM to interact with the spin system identification module that was being written in Aberdeen.

5.3.3 Spin System Class

The spin system class contains the information from the spin system identification. The spin system class is not used other than in the creation of the connection class (Figure 5.1).

5.3.4 Connection Class

The connection class models the inter-connections of a spin system that has been identified during spin system identification. For each spin system object a connection object is generated. A number of the attributes of a connection object are copied from the spin system objects whose connections it models. The attributes of the connection class are:

- Connection-id attribute: the symbol identifies the connection object within the 2D-SAM.

- Spin-system attribute: contains the spin system object whose connections the object models.

- Chemical shift attributes (NH-shift, a-shift, B1-shift[26] and B2-shift): are floating point numbers that store the NH, α, $\beta 1$ and $\beta 2$ chemical shifts of the spin system. This allows faster and simpler access than searching the list of chemical shifts found in the spin system object.

- Connection attributes (NN-connections, a-connections, B1-connections and B2-connections): list the connections of the spin system. The elements of the list will consist of pairs of objects, e.g. ((connection peak) (connection peak)). The first object will be another connection object. The second object will be the peak that possibly links the two connection objects.

- Connecting-peaks attribute: a list of all the peaks that possibly link the

[26] In the case that the spin system has been identified as possibly being generated by a glycine the B1-shift will be the second α chemical shift.

connection to other connection objects. The list will have pairs of elements, a peak and the connection it links to. The attribute will be a list.

- Intra-peaks attribute: lists the intra-residue peaks of the spin system. Derived from an attribute of the spin system object. The elements of the list will be peak objects.

- Identity-scores attribute: is a list of the possible amino acid identities of the spin system. Each possible identity will have an associated probability with it; the probability that the spin system is an amino acid of that type. The list will have pairs of elements; an amino acid and an associated probability, e.g. ((V 0.4) (T 0.987)).

- Connection-used-at attribute: records where the connection is used in the individuals of the sub-population. The attribute is an integer array, the same size as the sub-population. Each integer in the array records the position of the connection in the corresponding individual in the sub-population. If the 23rd integer in the used-at array is 5 then the index of the connection will be found at the 5th position of the chromosome array of the 23rd individual of the sub-population. If the connection is not used in an individual then a -1 appears at the corresponding position in the used-at array. The connection used-at attribute performs the same function as the peak used-at attribute.

- Connection-index attribute: is an integer that gives the position of the connection object in an array of connection objects[27]. The array is an attribute of the connections class. The index of a connection is used in the chromosome array of an individual to define the position of the connection in the sequential assignment the chromosome encodes. The connection index attribute performs the same function as the peak index attribute.

- Connection-position attribute: is an integer that states the actual position of the spin system whose connections the object models in the sequential assignment. This is a testing tool used with proteins whose sequential assignment is known. The attribute will be used when calculating the number of spin systems in the correct position in the sequential assignment.

[27] The array is the array attribute of the connections class.

The connection objects are grouped together and ordered in the connections class.

5.3.5 Connections Class

The connections class has 5 attributes that contribute towards the creation of a sequential assignment from the chromosome array of an individual. The attributes of the class are:

- Number attribute: an integer stating the number of connection objects in the 2D-SAM. This is equivalent to the number of spin systems in the spin system identification.

- List attribute: a list of the connection objects in the 2D-SAM.

- Array attribute: an array of the connection objects in the 2D-SAM. The attribute is the array from which the index of a connection is derived.

- Arrays attribute: an irregular multi-dimensional array of connection objects. The 1st dimension of the array is the same as the number of amino acids in the protein. The 2nd dimension of the array will be the number of connections that could be assigned to that position in a sequential assignment, e.g. if the 9th amino acid in a protein is a glutamine then the 9th array of the arrays attribute will contain all the connections that have been identified as possibly being generated by a glutamate.

The sequence class[28] models the amino acid sequence of the protein. This is the standard sequence class described in (Chapter 4).

6.0 Three Dimensional Sequential Assignment Module

The three dimensional sequential assignment module (3D-SAM) was derived from the two dimensional sequential assignment module; modifying it to use a different type of spectrum as input. The spectrum input into the module is a three dimensional heteronuclear ^{15}N-^{1}H HMQC NOESY spectrum (section 2.3). The other input and the output remain the same. The design and implementation of the 3D-SAM is essentially the same as the 2D-SAM. The differences are to enable the module to use a three dimensional spectrum.

The interpretation of a three dimensional heteronuclear ^{15}N NOESY spectrum (section 2.3) is different from a two dimensional homonuclear spectrum. The centre of the peaks in the three dimensional spectrum will be defined by three chemical shifts. The three chemical shifts will each come from a different nucleus[29], ^{1}H, ^{15}N and $N^{1}H$. If the peak is an intra-residue peak the ^{1}H chemical shift will come from the same amino acid as the ^{15}N and $N^{1}H$ chemical shifts. If the peak is an inter-residue peak the ^{1}H chemical shift will come from one amino acid and the ^{15}N and $N^{1}H$ chemical shifts will come from another amino acid. The peaks generated by an amino acid[30] will the same ^{15}N and $N^{1}H$ chemical shifts. In a spectrum displayed with $N^{1}H$, ^{1}H and ^{15}N dimensions as the x, y and z axis respectively the peaks generated by the same amino acid spin system will appear as a vertical strip of peaks (Figure 2.10). The intra-residue peaks have the same ^{1}H chemical shift (y axis) as a proton of the amino acid that generated the peak. The inter-residue peaks have the same ^{1}H chemical shift as the proton of the other amino acid that contributed to the generation of the peak. The links between spin systems are found by aligning ^{1}H chemical shift of an intra-residue peak of one spin system with the ^{1}H chemical shift of an inter-residue peak of another spin system. The extra dimension and the different pattern of peaks makes sequential assignment using a three dimensional heteronuclear ^{15}N NOESY spectrum much easier and often allows the sequential assignment of proteins that could not be assigned using a two dimensional homonuclear spectrum alone.

The rest of this chapter is split into two sections: design and evaluation.

[29] The exception being an intra-residue NH peak. The $N^{1}H$ and ^{1}H will come from the same nucleus.

[30] A peaks can be considered as being generated by the amino acid whose nuclei supplied the ^{15}N and $N^{1}H$ chemical shifts.

6.1 DESIGN OF THE 3D-SAM

The design of the 3D-SAM is identical to that of the 2D-SAM (Section 5.1) with some modifications to the coding and fitness function. The modifications occur at a low level in the module, leaving most of the design unchanged.

6.1.1 CODING 3D-SAM

The modifications to the design of the coding are to the peak, spin system and connection objects and a change in the method used to find possible links between connections. The change in the objects is the addition of an extra dimension or chemical shift. The change in the way that links are found is to allow for the different patterns of peaks generated in a three dimensional heteronuclear ^{15}N NOESY spectrum.

The peak object is modified to include and extra chemical shift attribute, d3, a floating point attribute. The spin system and connection objects each have an additional chemical shift attribute, floating point attributes. These attributes contain the ^{15}N chemical shift of a spin system from the spin system identification.

The method of finding links between spin systems changes due to the change in the spectrum used. To find all the links for each connection object takes three steps:

- The first is to find all the peaks generated by each amino acid. The spin system identification will contain the ^{15}N and N^1H chemical shift of each spin system. The peak list is then searched for peaks that have the same ^{15}N and N^1H chemical shifts, within certain tolerances. The tolerances will be user defined and each chemical shift dimension will have its own tolerance. Any peak that matches the search criteria is recorded as possibly belonging to that connection. A peak can belong to more than one connection. (Figure 5.4)

- The second step is to identify whether the peak is either an inter or an intra residue peak. The ^1H chemical shift of an intra-residue peak will match one of the ^1H chemical shifts of the connection to which it belongs, while the ^1H chemical shift of an inter-residue peak will not match one of the ^1H chemical shifts of the connection to which it belongs. To match the chemical shifts must be within the tolerance set for intra-residue ^1H chemical shifts. Each connection is examined in turn and its peaks are listed as either intra or inter residue peaks.

- The third step is to find all the possible links between connection objects. Each connection is examined in turn. The ^{1}H chemical shifts (NH, α, β1 or β2) are compared against the ^{1}H chemical shifts of the inter-residue peaks of the other connections. If the chemical shifts of a connection and a peak are within the defined tolerance then there is a possible link between the connection and the connection to which inter-residue peak belongs. The possible links found are recorded. In the connection being examined the link is recorded in the relevant connections list (NH, α, β1 or β2), depending on the chemical shift which aligns with the peak's chemical shift. In the connection to which the peak belongs the link is recorded in the NH connection list.

Once all the possible links have been found the coding is the same as for the 2D-SAM.

6.1.2 FITNESS FUNCTION FOR 3D-SAM

The differences between the fitness function in the 2D-SAM and 3D-SAM are confined to the way in which each NOE link is assessed The score for each NOE link must represent the probability that the peak links the two connection objects. The ^{15}N and N^{1}H chemical shifts of the peak will align with the chemical shifts of one connection and with the ^{1}H chemical shift of the other connection. This gives three chemical shift alignments to be assessed. The better the alignment of chemical shifts the greater the probability that the peak links the two connection objects. In 2D-SAM the chemical shifts were of the same type and they were weighted equally; weighting means the weight or importance ascribed to each factor. In the 3D-SAM the chemical shifts are of different types so different weighting would seem appropriate. The obvious weighting would be to have the ^{1}H alignment ascribed a weighting of 0.5 and the ^{15}N and N^{1}H alignments ascribed a weighting of 0.25 each. In effect the alignment of the peak with each connection is weighted equally, 0.5 each. The most likely source of ambiguity is in the number of connection objects with a ^{1}H chemical shift that can align with the ^{1}H chemical shift of an inter-residue peak. The assigning of a weighting of 0.5 to this alignment should mean that the variations in this alignment will have a critical effect on the score of a link. The assessment of the fitness of an NOE link with a peak with the chemical shifts P^{1}H, P^{15}N and PN^{1}H, and two connections with chemical shifts C1^{1}H and C2^{15}N and C2N^{1}H can be seen in Equation 6.1. The C1 or C2 in the variable name denotes the connection object to which the chemical shifts belong. The variable with Max in the name define the maximum values for which a score is calculated for the three chemical shifts. The Max values will be the same as the tolerances used when searching for possible connection-peak-connection links. The max values also allow for differences in the scale of the different types of chemical shifts.

Equation 6.1

$$NOE\_score = 1 - \left(0.5\frac{|P^1H - C1^1H|}{Max\_^1H} + 0.25\left(\frac{|PN^1H - C2N^1H|}{Max\_N^1H} + \frac{|P^{15}N - C2^{15}N|}{Max\_^{15}N} \right) \right)$$

6.2 Evaluation of the 3D-SAM

The 3D-SAM was evaluated using Dihydrofolate Reductase (DHFR) (41). The protein is fairly large, 162 amino acids in length. The sequential assignment of the protein is known and was performed using three dimensional heteronuclear NOESY. The peak list used to perform the sequential assignment was also available. The conclusions about the performance of the 3D-SAM were drawn from the results of the tests outlined in the next section. The input data for the 3D-SAM can be found on the enclosed CD-ROM.

6.2.1 Testing of the 3D-SAM

A number of experiments were performed to evaluate the 3D-SAM. The initial experiments were performed using a correct and precise connection assignment. In later experiments the precision of the connection assignment was reduced and in the final set of experiments errors were introduced into the connection assignment supplied to the 3D-SAM. Each experiment was repeated twenty four times. The 3D-SAM is a stochastic program and its output will vary even if the input remains constant. The variation in the output, the sequential assignment, is partly what the 3D-SAM is being evaluated for.

The 3D-SAM has a number of user defined parameters. Some are concerned with the spectrum others with the working of 3D-SAM. The spectral parameters are the chemical shift tolerances used in finding possible links and in the fitness function. All the other parameters affect the environment that the individuals of sub-population evolve in. The parameters set for each were the same as those used for the 2D-SAM except where stated otherwise . The default parameters are listed below:

- The population parameter is 800. The main-population has 800 individuals and the sub-population has 80 individuals. The larger the population the better the search space is sampled and the longer taken to evolve one generation.

- The generations parameter is set at 12000. The sub-population is evolved for 12000 generations; effectively 1200 generations for the main-population.

Each experiment had 24 test runs performed; multiple runs were performed to allow for the stochastic nature of the 3D-SAM. The tests were performed using a correct and precise spin system identification.

6.2.2 Conclusions

When the 3D-SAM is given a precise and correct assignment it can generate a sequential assignment that is 71 percent correct. The variation in the percentage correct is due to the size of the search space and the quality of the data supplied to the 3D-SAM. The DHFR protein is 170% larger than protein G, but the size of the search space is exponentially larger. With the data entered into the 3D-SAM for the DHFR protein there are 1.935 x 10^{2108} possible sequential assignments. The corresponding figure for protein G is 1.930 x 10^{826} possible sequential assignments. The DHFR search space is enormous by any definition, though in practice the more critical number is the number of possible permutations in the sequence of spin systems, which is considerably smaller 1.166 x 10^{130}. The permutations of the sequence of spin systems will contribute to how many local maxima there are. The problem with reordering the sequence of spin systems is that between one sequence and another of higher fitness the intermediate sequences can have a lower fitness than the original sequence. This is the problem the genetic operators are designed to overcome. A change in the sequence of peaks where the sequence of spin systems remaining the same does not have this problem. Any intermediates will not have a lower fitness. Despite the huge increase in the size of the search space the 3D-SAM still gives a reasonably accurate answer, considering that a spectroscopist will normally derive an answer from a number of spectra run under different conditions (Section 2.1).

The quality of the data supplied, the NOESY spectrum, will also affect the variation in the accuracy of the assignment between runs. Sequences of spin systems that are well connected by NOE peaks will remain relatively constant throughout all the experimental runs. Those sequences that are not well connected will vary between experimental runs. When a sequence of spin systems is not well linked a single incorrect NOE link can result in the sequence of spin systems being incorrect. This is particularly true of non-sequential NOE links; these links can give particularly good scores. The incorrect sequences of spin systems can give scores comparable to the scores of the correct sequence. Which sequence is found first determines which will be used as their scores are very similar. In certain cases there are no NOE peaks to link the spin systems. The sequence in these cases is determined by random chance; allowing for the restriction that the amino acid identity list places on the possible positions of a connection. The result of poor NOE links is that position of certain connections in the sequential assignment are determined by random chance. The randomness is result of the random action of the various genetic operators.

The variation in the accuracy produced by the size of the search space and data can be seen in the difference between the percentage standard deviation of the number of residues in the correct position (10.5%) and fitness (3.2%), (Table 6.1). The standard deviation is more than three times greater for the residues in the correct position than for the fitness. The difference is due to the poorly linked regions of sequential assignment being very variable in sequence of spin systems with a small variation in the fitness of the sequence. This could indicate that the fitness function is not working properly but investigation of the sequential assignments produced by the 3D-SAM suggest that poor NOE links are the cause of the variation rather than a poor fitness function. The better figure to use when estimating the variation between experimental runs would be the standard deviation in the fitness (3.2%) rather than the number of spin systems in the correct position.

The generation of a 71% accurate sequential assignment from one NOESY spectrum is sufficient to be of use. Although this figure was obtained with a 'perfect' spin system identification. The first experimental run took 11 hours and 4 minutes to run on an SGI workstation, the same as the one used in testing the 2D-SAM. The experiments could run overnight to give an assignment of the quality listed in the enclosed CD-ROM.

Figure 6.1 Results of 3D-SAM.

Experiment Number	No. Spin Systems Correct	Percentage Correct	Fitness
1	115	74.2	309.3
2	124	80.0	312.8
3	126	81.3	315.6
4	117	75.5	298.6
5	109	70.3	297.7
6	96	61.9	288.3
7	105	67.7	300.2
8	108	69.7	300.9
9	112	72.3	302.9
10	93	60.0	283.3
11	121	78.1	307.7
12	112	72.3	306.0
13	134	86.5	315.6
14	123	79.4	304.1
15	92	59.4	282.5
16	118	76.1	299.3
17	108	69.7	296.3
18	104	67.1	285.3
19	100	64.5	293.1
20	104	67.1	296.5
21	92	59.4	298.6
22	102	65.8	304.1
23	125	80.6	312.3
24	114	73.5	308.4
Average	110.6	71.4	300.8
Stan Dev %	10.5	10.5	3.2

7.0 Backbone Assignment Module 1

The backbone assignment module 1 (BAM-1) assigns the backbone resonances of a protein using triple resonance spectra. The BAM-1 uses three sources of information: the amino acid sequence of the protein, several triple resonance spectra of the protein and the rules on the assignment of the spectra. The BAM-1 takes this information and produces a list of peak systems. A peak system is a group of peaks drawn from the various spectra that were generated by the same amino acid. The peaks could have been completely or partially generated by the amino acid depending on the type of NMR experiment that generated them. There are a large number of triple resonance experiments that can be used in the assignment of the backbone of a protein. The number and combination of experiments used will vary depending on the protein and the spectroscopist running the experiments. To cope with this plethora of experiments the rules for the assignment of spectra are external to the BAM-1. The separation of the assignment rules and BAM-1 allows the user to define how to interpret any triple resonance spectra desired.

The BAM-2 (described in chapter 8) takes the peak systems produced by the BAM-1 and creates a sequential assignment.

The backbone assignment of a protein using triple resonance spectra is described in section 2.4. Each element of the input to the BAM-1 supplies certain information. The amino acid sequence of a protein gives the number of peak systems that can to be found in the spectra. The spectra, in the form of peak lists, supply the peaks generated by the amino acids of the protein. The peaks will have the chemical shifts of the various nuclei of an amino acid or a pair of adjacent amino acids. The interpretation or assignment rules define how the chemical shifts of the peaks are used to create the peak systems. The rules do this by defining which chemical shifts of a peak are shared with other peaks generated by the same amino acid. The rules can also define the relative intensity of peaks in a peak system.

The rest of this chapter is divided into two sections: design and evaluation.

7.1 Design of the BAM-1

The design of the BAM-1 was produced by the separation of the original BAM into the BAM-1 and the BAM-2. The BAM-1 or the part of the BAM that became the BAM-1 was largely unmodified from the first prototype developed with the exception of one of the genetic operators. The BAM used some of the code and concepts from an earlier non GA program designed to use three dimensional heteronuclear spectra in spin system assignment. The design of the BAM-1 is an object oriented one based around the same GA core (Chapter 4) as the other modules. The design of the BAM-1 is shown in Figure 7.1. The individual and population classes form the GA core. The peak, alignment and peak system classes form the problem specific element of the BAM-1. The peaks and peak systems classes list and order the objects of the peak and peak system classes respectively.

The remainder of this section is split into three sub-sections: the coding, fitness function and the genetic operators. The program is designed to be flexible in the number and type of triple resonance spectra it can use. To describe the remaining sections of the design example triple resonance spectra must be used. The examples used are those described in Section 2.4.

7.1.1 CODING

The two inter-related elements of the coding, the type of chromosome and the method of conversion from the chromosome to a solution, are relatively easy choices in the case of the BAM-1. The ease of choice of the coding is due to the simplicity of the spectra and the solution being generated. The solution is a list of peak systems, each of which is a list of peaks (Figure 7.2). The number of peaks will depend on the number of spectra being used and the number of peaks an amino acid generates in each spectrum. For example when four spectra are used with two spectra generating one peak per amino acid and the other two generating two peaks per amino acid, there will be up to six peaks in each peak system (Section 2.4). To encode this solution in a chromosome the simplest method proved to be effective. The chromosome is an integer array where each integer defines the position of a peak in an array of peaks derived from peak list input of the BAM-1. The array from which each peak is chosen, and the number of peaks that comprise a peak system are defined by the user. Using the previous example the definition will be "HNCOCA HNCA HNCO HNCACO HNCA HNCACO". Thus first peak of the peak system is selected from the HNCOCA peak list, the second from the HNCA peak list and so on through the peak system. The size of the integer array will be equal to the number of peaks in a peak system multiplied by the number of peak systems.

In practice the coding is performed using the chromosome array of an individual object and the peak systems object. The peak systems object contains all the peak system objects in its peak-system-list attribute. The peak system objects in the list are processed consecutively. The first step in the processing is to find the section of the chromosome array that encodes the

Figure 7.1 Design of the BAM-1.

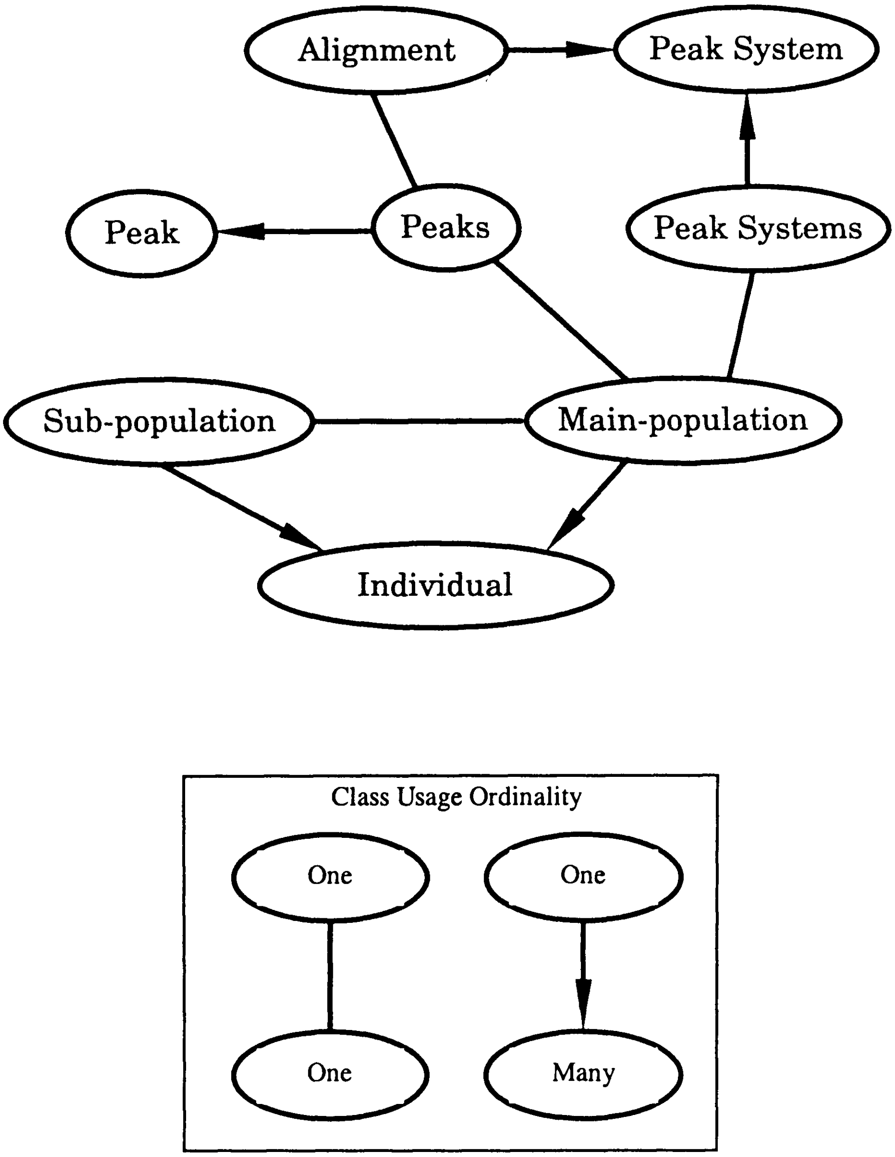
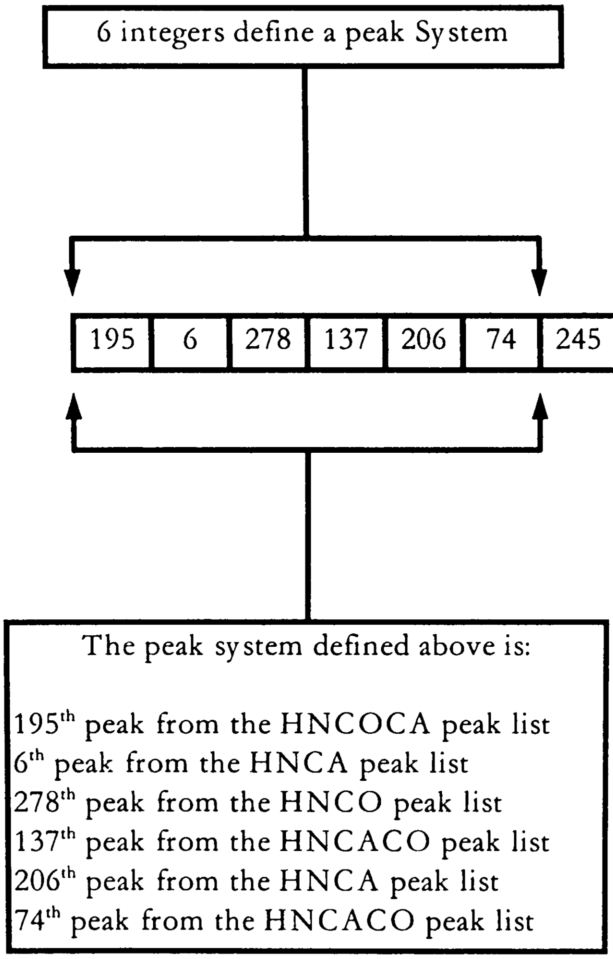


Figure 7.2 Expression of the BAM-1 integer chromosome to a peak system assignment.



195	6	278	137	206	74	245	20	99

The peak system defined above is:

195th peak from the HNCOCA peak list
6th peak from the HNCA peak list
278th peak from the HNCO peak list
137th peak from the HNCACO peak list
206th peak from the HNCA peak list
74th peak from the HNCACO peak list

peaks of the peak system. This section starts at the integer designated by the peak-system-chromosome-index attribute. The integer at that position is read and used as the index to the first array contained in the peaks-arrays attribute of the peak system object. The peak object is inserted into the first position of the peaks attribute of the peak system. The process is repeated until the peaks attribute has its full complement of peaks and the next peak system object is processed. At the end of the decoding process each peak system object peaks attribute is full of peak objects. As in the previous modules a blank peak object is used to define blank positions for implementation reasons.

7.1.2 FITNESS FUNCTION

The fitness function of the BAM-1 determines the quality of the peak systems of an individual. The fitness of an individual will be the sum of the fitness of each peak system it represents. Each peak system will have a fitness between 0.0 and 1.0. The fitness of each peak system is calculated using two factors, the first is derived from the rules contained in the alignment object and the peaks of the peak system and the second is calculated using just the peaks of the peak system.

The first factor is the "distance factor". This is calculated using the alignment rules contained in the alignment object. The distance factor is the average of the factors calculated for each rule. Consider for example the alignment rule (0 1 d2 d2 0.3). In this rule the d2 chemical shift of the 1st peak ($cs0^{d2}$) in a peak system is subtracted from the d2 chemical shift of its 2nd peak ($cs1^{d2}$). The absolute value of the subtraction is divided by the range (r) 0.3 ppm (Equation 7.1). If the result of the calculation (d) is less than or equal to 1.0 then it is added to a running total (T). If the result is greater than 1.0 then the evaluation of the alignment rules is terminated and the distance factor (*df*) is ascribed a value of 0.0. The distance factor is an assessment of the alignment of the peaks. Also contained in the list of alignment rules are the intensity rules. An intensity rule states that the first peak defined by the rule must have an intensity less than or equal to the second, for example the rule (2 4 in in) states that the intensity of the 3rd peak must be less than or equal to the 5th peak in the peak system[31]. If an intensity rule is true then 0.0 is added to the total (T), if it is false then 1.0 is added to the total (T). The intensity rules are used in experiments where two peaks are generated in the spectrum for each amino acid and there is an intensity difference between the two peaks. An

[31] The peaks of a peak system are stored in an array, the peaks-system peaks attribute. The array is indexed 0,1,2...n; the index 3 therefore refers to the 4th element of the array.

example would be an HNCA experiment (Section 2.4). The intra residue peak is usually, but not always, more intense than the inter residue peak. Once all the rules have been evaluated the total is divided by the alignment-constraint-number attribute (N) of the alignment object. The results of the division is then subtracted from 1.0 to give the distance factor (*df*) for that peak system (Equation 7.2). When one or both of the peaks being examined in a alignment rule is a blank peak then the distance (d) is zero; blank peaks give a perfect alignment.

$$d = \frac{\left|cs0^{d2} - cs1^{d2}\right|}{r}$$ Equation 7.1

$$df = 1 - \frac{T}{N}$$ Equation 7.2

The second factor deals with blank peaks in peak systems. If a peak system is composed entirely of blank peaks the distance factor will give a perfect score. The blank peak factor would give a factor of zero. The blank peak factor (bf) is the number of non-blank peaks in a peak system divided by the total number of peaks.

The two factors, distance (df) and blank peak (bf), are multiplied together to give the fitness of the peak system (*F^{ps}*), Equation 7.3. The original fitness function used only the distance factor; similar to the fitness function used in the 2D-SAM and 3D-SAM. The blank peak was added during development of the BAM-1, when the assessment of a peaks alignment with a blank peak was changed. The original blank peak alignment was assessed as being the worst possible alignment; the tolerance of the appropriate dimension being used as the alignment distance. The current blank peak alignment is assessed as being the best possible alignment; zero being used as the alignment distance. The constraint factor was developed to emphasise the difference between viable and non-viable peak systems.

$$F^{ps} = df \times bf$$ Equation 7.3

7.1.3 Genetic Operators

The BAM-1 has three genetic operators: crossover, mutate aligned peak and mutate ideal peak. The crossover operator is a simple genetic operator and swaps sections of backbone assignment between individuals. The other two operators perform a function similar to the mutation operator in the simple

GA described in chapter 3. Both operators either implicitly or explicitly make use of the alignment rules to place a peak system. The peaks are not randomly chosen. Two variants of the aligned and ideal peak operators are also used. These variants operate on an entire peak system.

The crossover operator randomly selects the position of a peak system on the chromosome array of an individual. Another peak system is randomly chosen between the first peak system and the end of the chromosome array. The peaks of the two peak systems and all the peak systems in between are then swapped between the two individuals undergoing crossover. The crossover is done in such a way as to maintain the consistency of the backbone assignment the individuals represent. A flow chart of the crossover operator is shown in Figure 7.3.

The mutate aligned peak operator implicitly uses the alignment rules by using the aligned-peaks attribute of a peak object (Figure 7.4). The operator inserts a peak into a peak system that is aligned with one of the other peaks in the peak system. A peak is considered to align with another peak in a peak system when the two peaks conform to the relevant alignment rules contained in the alignment object. A peak object has all the peak objects with which it can align with listed in its aligned-peaks attribute. The attribute is an array that lists the peaks that can be aligned with the peak at each position in the peak system. A peak is randomly selected in a randomly selected peak system from a randomly selected individual. Another peak position in the selected peak system is randomly chosen. The aligned-peaks attribute of the peak chosen is searched for the list of peaks that can be aligned with the peak position selected. A peak is then randomly selected from the list. The new peak is then inserted at the selected peak position. The recorded positions of the new peak and the peak that previously occupied the selected position are then updated.

A genetic operator was created that mutates an entire peak system using the mutate aligned peak operator. The operator is called the mutate aligned peak system operator. An individual is chosen at random from the sub population, then a peak system is randomly chosen from the chromosome array of the individual. The position of a peak is randomly selected on the chosen peak system. A peak is then randomly chosen from the list of peak objects that can be placed at that position in the peak system. The chosen peak in then inserted into the chosen position. The used at attributes of the chosen peak and the one that has just been overwritten are then updated. The mutate aligned peak operator is then invoked for all the other peaks in the peak

Figure 7.3 Flow chart of the crossover operator.

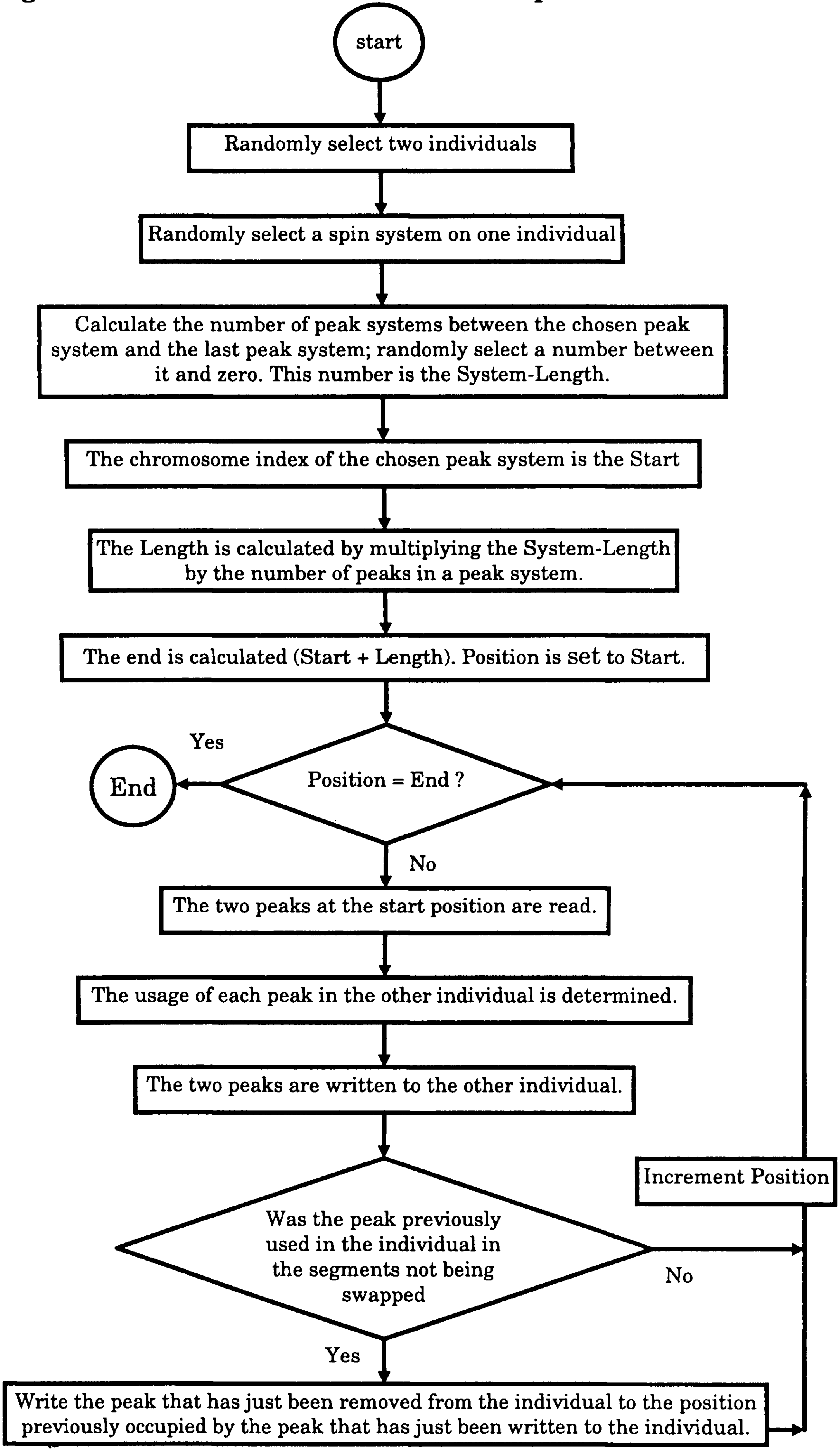
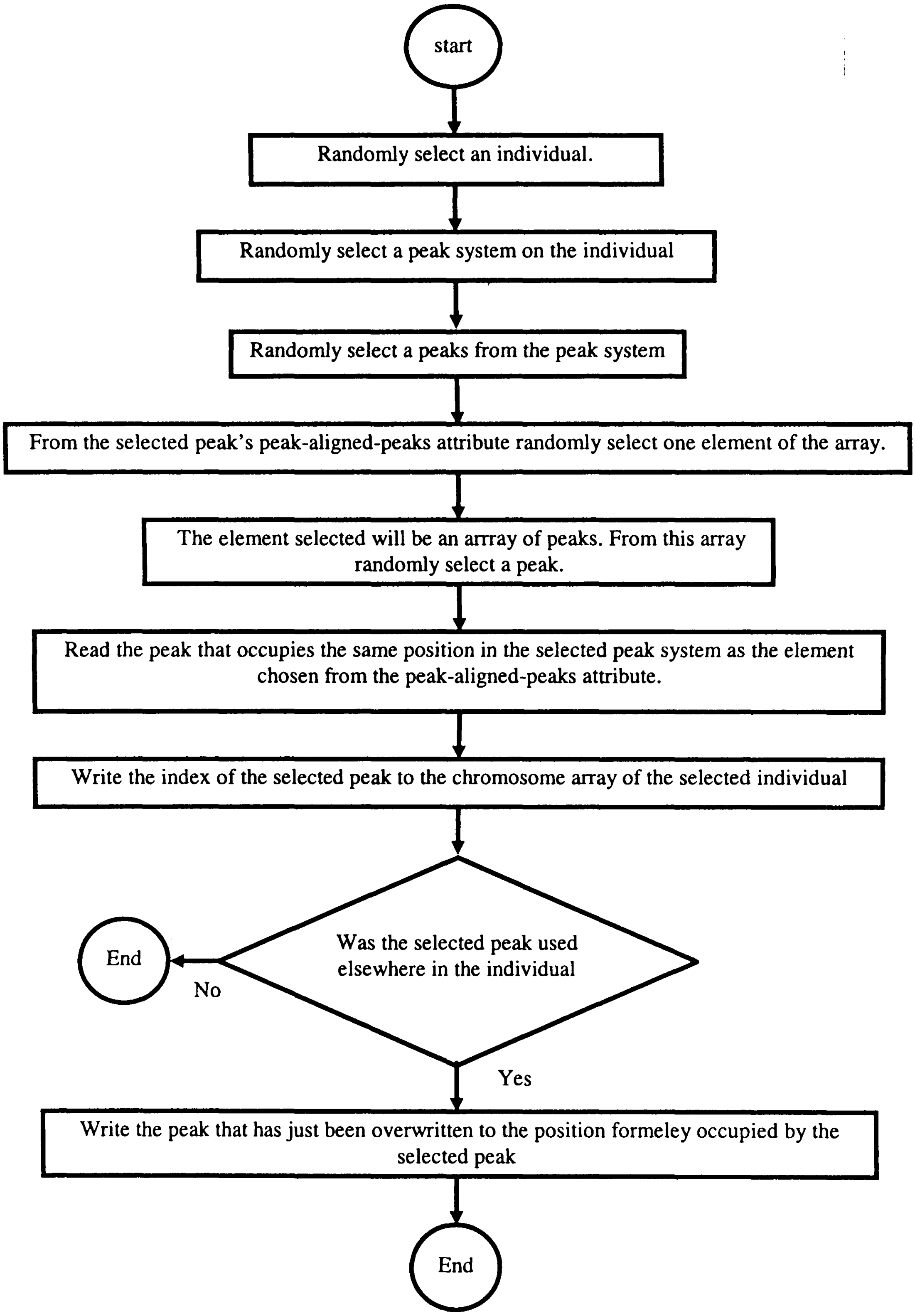

start
Randomly select two individuals
Randomly select a spin system on one individual
Calculate the number of peak systems between the chosen peak system and the last peak system; randomly select a number between it and zero. This number is the System-Length.
The chromosome index of the chosen peak system is the Start
The Length is calculated by multiplying the System-Length by the number of peaks in a peak system.
The end is calculated (Start + Length). Position is set to Start.
Position = End ?
Yes
End
No
The two peaks at the start position are read.
The usage of each peak in the other individual is determined.
The two peaks are written to the other individual.
Was the peak previously used in the individual in the segments not being swapped
No
Increment Position
Yes
Write the peak that has just been removed from the individual to the position previously occupied by the peak that has just been written to the individual.

Figure 7.4 Flow chart of the mutate aligned peak operator.


start
Randomly select an individual.
Randomly select a peak system on the individual
Randomly select a peaks from the peak system
From the selected peak's peak-aligned-peaks attribute randomly select one element of the array.
The element selected will be an arrray of peaks. From this array randomly select a peak.
Read the peak that occupies the same position in the selected peak system as the element chosen from the peak-aligned-peaks attribute.
Write the index of the selected peak to the chromosome array of the selected individual
Was the selected peak used elsewhere in the individual
End
No
Yes
Write the peak that has just been overwritten to the position formeley occupied by the selected peak
End

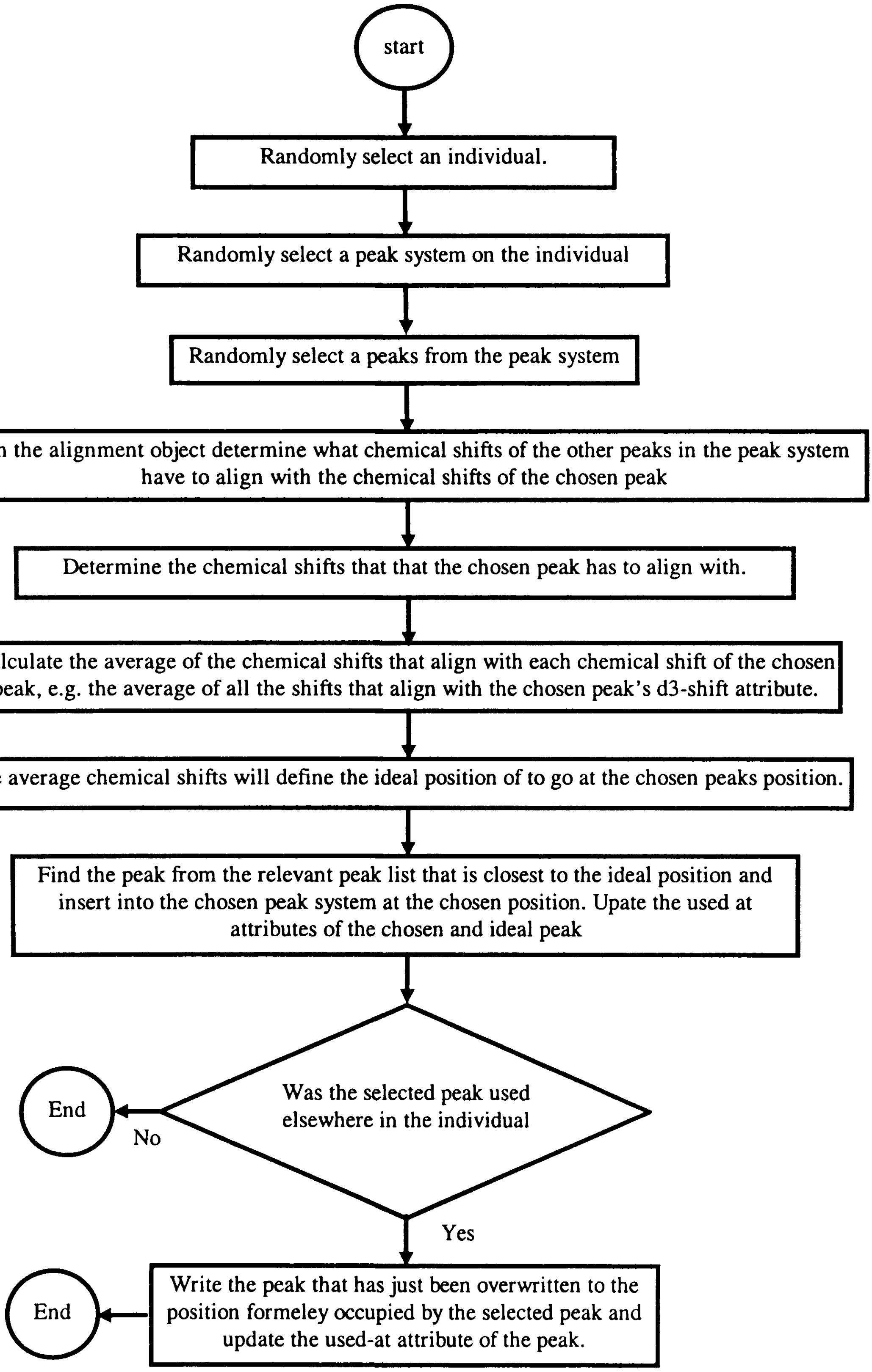
system using the chosen peak as the peak they must align with. The purpose of the operator is to create original peak systems. The other operators all use the existing peaks of a peak system to modify the other peaks. This operator creates a peak system without reference to the existing peaks of the peak system. The operator is similar to the method used to create the initial backbone assignments.

A modified version of the aligned peaks operator is used to create the initial backbone assignments of each individual. The modification is that the operator inserts only unused aligned peaks into a spin system. A peak system is randomly chosen on the chromosome array of an individual. An randomly chosen unused peak is then inserted into a randomly chosen position in the peak system. A peak is chosen from the appropriate type of spectra to be placed at that position. The aligned-peaks attribute of the inserted peak is then used to find the unused aligned peaks. The other positions in the peak system are then filled with the unused aligned peaks. The process is repeated for all the other peak system in the individuals.

The mutate ideal peak operator uses the alignment rules explicitly (Figure 7.5). The operator determines the ideal peak for a position in the peak system. An ideal peak in this context is the peak that best aligns with the other peaks in the peak system. A peak position is randomly chosen from a randomly chosen peak system from a randomly chosen individual. The alignment rules in the alignment object are searched for references to the peak position e.g. if the peak position is 0 all the rules that contain 0 are found. For each rule found the other peak referred to is examined. The chemical shift referred to by the rule is then determined e.g. for the rule (3 0 d2 d3 -11) the d2 chemical shift of the peak at position 3 is determined. The chemical shift is then added to the relevant dimensional total e.g. in the previous example the chemical shift would be added to d3 total. Once all the alignment rules have been processed the dimensional totals are converted to averages. The averages give the ideal position for a to align with the other peaks in the peak system. The peak closest to the ideal position is found and inserted in the selected peak position.

As with the mutate aligned peak operator there is an operator that uses the mutate ideal peak operator. The mutate ideal peak system operator randomly selects an individual from the sub population. A peak system is then randomly selected from the chromosome array of the selected individual. The position of a peak is then randomly selected on the chosen peak system. The mutate ideal

Figure 7.5 Flow chart of the mutate ideal peak operator.


start
Randomly select an individual.
Randomly select a peak system on the individual
Randomly select a peaks from the peak system
From the alignment object determine what chemical shifts of the other peaks in the peak system have to align with the chemical shifts of the chosen peak
Determine the chemical shifts that that the chosen peak has to align with.
Calculate the average of the chemical shifts that align with each chemical shift of the chosen peak, e.g. the average of all the shifts that align with the chosen peak's d3-shift attribute.
The average chemical shifts will define the ideal position of to go at the chosen peaks position.
Find the peak from the relevant peak list that is closest to the ideal position and insert into the chosen peak system at the chosen position. Upate the used at attributes of the chosen and ideal peak
Was the selected peak used elsewhere in the individual
No
End
Yes
Write the peak that has just been overwritten to the position formeley occupied by the selected peak and update the used-at attribute of the peak.
End

peak operator is then invoked for all the peak positions in the peak system using the chosen position as a start point.

The original design of the BAM-1 incorporated a completely random peak mutation operator instead of the aligned peak mutation operator. The ideal peak mutation operator was part of the design from the beginning. Due to the completely random nature of the simple peak mutation operator it would mostly introduce improbable peaks into a peak system. Only rarely would it introduce a peak that aligned with one or more of the existing peaks. The ideal peak mutation operator works well when the peak system needs only one peak to become a viable peak system or when a viable peak system needs to be improved. It does not work as well when some peaks in the peak system do not align with each other. The ideal operator is very effective with viable or near viable peak systems but not with non-viable peak systems. The operator towards the end of a run would often find that the current peak was the ideal peak. Thus no mutation occurred. To improve the performance of the BAM-1, particularly at the beginning of a run, the aligned peak operator was developed to operate alongside the other two operators. The performance of the BAM-1 improved. The performance was improved further when the use of the simple peaks mutation operator was stopped.

The two peak system genetic operators were both subjected to extensive testing. The mutate aligned peak system definitely improved the performance of the BAM-1 by creating original peak systems. The mutate ideal peak system operator gave no noticeable improvement in performance and it was not used in the BAM-1 that performed the test runs in the evaluation section. The use of the mutate ideal peak operator was stopped for one series of test runs; both aligned genetic operators were used. This reduced the performance of the BAM-1. The use of the mutate ideal peak system operator would restore the performance of the BAM-1. As the ideal peak system operator is more complex and more demanding of computer resources than the mutate ideal peak operator the mutate ideal peak the operator was not used in the final configuration of the BAM-1. The final configuration of the BAM-1 uses the aligned and ideal peak operators and aligned peak system operator.

7.2 Evaluation of The BAM-1

The BAM-1 was evaluated using the spectrum of the flavin mononucleotide (FMN) binding domain of Cytochrome P_{450} Reductase (42).The protein is reasonably large, 184 amino acids in length. The backbone assignment of the protein was not known when the BAM-1 was under development, although the spectra had been recorded and the peaks had been picked. The backbone assignment of the protein was being assigned as the BAM-1 was being tested and the results of the manual assignment were not examined until testing was completed.

The input data for the BAM-1 can be found on the enclosed CD-ROM.

7.2.1 Testing of the BAM-1

A number of test runs were performed on the BAM-1. The bulk of these were performed to establish the best combination of genetic operators both in terms of the operators used and the frequency of those operators (Section 7.1.3). The first, second and third mutation rate parameters determine the frequency of use of the aligned peak mutation operator, the ideal peak mutation operator and the peak system aligned peak mutation operator. The parameters that control the behaviour of the BAM-1 are the same as for the previous GA modules. The spectral parameters are contained in the assignment rules that form part of the assignment object. The default parameters are listed below.

- The population parameter is 1000. The main-population has 1000 individuals; the sub-population has 100 individuals.

- The generations parameter is set at 12000. The sub-population is evolved for 12000 generations; effectively 1200 generations for the main-population.

- The crossover rate parameter is set at 0.5. 50 percent of the individuals in the sub-population undergo crossover every generation.

- The mutation rate 1 parameter is set at 0.2. 20 percent of the individuals sub-population are subjected to the aligned peak mutation genetic operator.

- The mutation rate 2 parameter rate is set at 0.1. 10 percent of the individuals sub-population are subjected to the ideal peak mutation genetic operator.

- The mutation rate 3 parameter rate is set at 0.1. 10 percent of the individuals sub-population are subjected to the peak system aligned peak mutation genetic operator.

The alignment rules are in the form of a list. Each alignment rule is itself a list. The first two numbers state the two peaks in a peak system that have an alignment. The second two symbols define the chemical shifts of the two peaks that must align. The last element is a floating point number that defines the tolerance of the chemical shifts alignment. The rules are listed below.

```
((0 1 D1 D1 0.03) (0 1 D2 D2 0.4) (0 1 D3 D3 0.3) (0 2 D1 D1 0.03)
 (0 2 D2 D2 0.4) (0 3 D1 D1 0.03) (0 3 D2 D2 0.4) (0 4 D1 D1 0.03)
 (0 4 D2 D2 0.4) (0 5 D1 D1 0.03) (0 5 D2 D2 0.4) (1 2 D1 D1 0.03)
 (1 2 D2 D2 0.4) (1 3 D1 D1 0.03) (1 3 D2 D2 0.4) (1 4 D1 D1 0.03)
 (1 4 D2 D2 0.4) (1 5 D1 D1 0.03) (1 5 D2 D2 0.4) (2 3 D1 D1 0.03)
 (2 3 D2 D2 0.4) (2 3 D3 D3 0.3) (2 4 D1 D1 0.03) (2 4 D2 D2 0.4)
 (2 5 D1 D1 0.03) (2 5 D2 D2 0.4) (3 4 D1 D1 0.03) (3 4 D2 D2 0.4)
 (3 5 D1 D1 0.03) (3 5 D2 D2 0.4) (4 5 D1 D1 0.03) (4 5 D2 D2 0.4)
 (1 4 IN IN) (3 5 IN IN))
```

The rules state that the d1 and d2 chemical shifts of all the peaks in a peak system must be aligned. That the zero and first peaks d3 must align and the second and third peaks d3 must also align. The intensity of the first peak should be less than that of the fourth peak and the same is true of the third and fifth peaks. The numbers correspond to the user defined list of spectra that forms part of the input to the BAM-1. In the case of the above alignment rules the spectra list is “HNCOCA HNCA HNCO HNCACO HNCA HNCACO”.

The fittest individual that evolved during the running of the BAM-1 was taken as the solution. The experiments were run on a number of SGI workstations. The experimental runs shown below were run on a workstation with a 200 MHz R4400 CPU and 64 Mb of RAM. The experiments were all run in a low priority batch queue. Thus the speed of a run will depend on the concurrent usage of the computer.

There are two sets of test runs. The first set of test runs were to establish the consistency of the performance of the BAM-1. The were four test runs performed with the default criteria (Table 7.1). The second set were to establish the optimum number for the generations parameter. The Generation parameter was set at 12000, 24000, 36000 and 48000 (Table 7.2).

The assignment generated by the third test run in Table 7.2 can be found on the enclosed CD-ROM. The manual backbone assignment of the FMN binding domain of Cytochrome P_{450} (1) can also be found on the enclosed CD-ROM. The peak systems in each assignment were compared against each other. The results of the comparison are shown in Table 7.3. The first column in the table

Table 7. 1

Generation	Maximum Fitness	Average Fitness	Minimum Fitness
12001	125.808	121.272	49.670
12001	125.066	122.037	82.861
12001	125.354	121.497	82.415
12001	125.590	121.813	81.790

Table 7.2 BAM-1 Results

Generation	Maximum Fitness	Average Fitness	Minimum Fitness
12001	125.101	121.581	68.730
24001	126.431	124.049	86.114
36001	126.868	124.651	80.557
48001	127.085	124.813	95.392

Table 7. 3 - Comparison of Automated and Manual Assignment.

Automated Peak System	Manual Peak System	Manual Peak System	Peak System Discrepancies
1	SPIN3		
2	SPIN2		
3	SPIN4		
4	SPIN1		
5	SPIN5		
6	SPIN7		
7	SPIN6		
8	SPIN12		
9	****	SPIN13	
10	####	AmbSpin6	
11	SPIN16		
12	SPIN11		
13	####	AmbSpin8	
14	SPIN15		
15	SPIN18		
16	SPIN19		
17	SPIN20		
18	SPIN41		
19	SPIN23		
20	SPIN24		
21	SPIN9		
22	SPIN26		
23	****	SPIN14	
24	SPIN25		
25	SPIN28		
26	SPIN17		
27	SPIN22		
28	SPIN21		
29	SPIN29		
30	SPIN53		
31	SPIN35		
32	SPIN36		
33	SPIN55		

34	SPIN33		
35	SPIN54		
36	SPIN59		
37	SPIN58		
38	SPIN30		
39	SPIN57		
40	SPIN60		
41	SPIN56		
42	****	SPIN56	
43	SPIN43		
44	SPIN62		
45	SPIN37		
46	SPIN61		
47	SPIN38		
48	SPIN34		
49	SPIN67		
50	SPIN8		swapped HNCA
51	SPIN50		
52	SPIN63		
53	SPIN40		
54	####	AmbSpin24	
55	SPIN65		
56	SPIN31		swapped HNCA
57	SPIN68		
58	####	AmbSpin11	
59	SPIN66		
60	SPIN77		
61	####	AmbSpin8	
62	SPIN48		
63	SPIN74		
64	SPIN73		
65	SPIN70		
66	####	AmbSpin11	
67	SPIN76		
68	SPIN75		
69	####	AmbSpin24	
70	SPIN80		
71	SPIN46		

72	SPIN78		
73	####	AmbSpin24	
74	SPIN70		
75	SPIN47		
76	SPIN83		
77	SPIN72		
78	SPIN84		
79	SPIN87		
80	SPIN90		
81	SPIN88		
82	SPIN81		
83	SPIN82		
84	****	SPIN86	
85	SPIN39		
86	SPIN92		
87	SPIN42		swapped HNCA
88	SPIN94		
89	****	SPIN64	
90	SPIN93		
91	SPIN32		
92	SPIN45		
93	SPIN96		
94	SPIN97		
95	####	AmbSpin10	
96	SPIN91		
97	SPIN49		
98	SPIN98		
99	SPIN51		
100	SPIN99		
101	SPIN101		
102	####	AmbSpin23	
103	SPIN140		
104	SPIN107		missing HNCOCA
105	SPIN143		
106	SPIN108		missing HNCOCA
107	####	AmbSpin23	
108	SPIN142		
109	SPIN141		

110	SPIN113		missing HNCOCA
111	SPIN120		
112	SPIN118		
113	SPIN111		
114	SPIN110		missing HNCOCA
115	SPIN144		
116	SPIN105		
117	SPIN112		
118	SPIN114		missing HNCOCA
119	SPIN145		
120	SPIN146		
121	SPIN109		
122	SPIN115		missing HNCOCA
123	SPIN147		
124	SPIN150		
125	----		
126	SPIN149		
127	SPIN151		
128	SPIN148		
129	SPIN152		
130	SPIN154		
131	####	AmbSpin28	
132	SPIN155		
133	SPIN119		missing HNCOCA
134	SPIN153		
135	SPIN124		missing HNCOCA
136	SPIN156		
137	SPIN121		missing HNCOCA
138	SPIN123		missing HNCOCA
139	SPIN102		missing HNCOCA
140	SPIN157		
141	SPIN129		
142	SPIN132		missing HNCOCA
143	SPIN128		
144	----		HNCOCA peak used in SPIN183
145	SPIN136		
146	----		

147	SPIN137		swapped HNCA
148	SPIN130		
149	SPIN125		
150	SPIN134		missing HNCOCA
151	SPIN158		
152	SPIN127		
153	####	AmbSpin9	
154	SPIN159		
155	----		
156	SPIN122		
157	----		
158	SPIN126		missing HNCOCA
159	SPIN160		
160	SPIN135		missing HNCOCA
161	****	SPIN161	missing HNCOCA
162	SPIN131		missing HNCOCA
163	SPIN133		missing HNCOCA
164	SPIN138		missing HNCOCA
165	SPIN163		missing HNCOCA
166	----		
167	SPIN168		
168	SPIN162		missing HNCOCA swapped HNCA
169	SPIN169		missing HNCOCA
170	####	AmbSpin5	
171	SPIN170		missing HNCOCA
172	SPIN164		
173	SPIN172		
174	SPIN165		swapped HNCA
175	****	SPIN175	missing HNCO
176	####	AmbSpin27	
177	SPIN176		missing HNCOCA
178	SPIN177		missing HNCOCA
179	****	SPIN106	missing HNCOCA HNCO
180	####	AmbSpin29	
181	SPIN178		
182	----		
183	####	AmbSpin40	

184	####	AmbSpin25	

identifies a peak system from the automated backbone assignment, the number refers to the position of the peak system in the file. The second column lists the matching peak system from the manual backbone assignment. There are three symbols used when there is no matching peak system in the manual backbone assignment. The first symbol "*****" denotes that the peak system is a duplicate peak system. A few of the C terminal residues of the FMN binding domain can have two conformations, giving duplicate peak systems for the amino acids of the C terminal of the protein. The peak systems have different N^{15} and NH^1 chemical shifts but the C^{13} chemical shifts remain the same. The second symbol "####" denotes that the peak system from the automated backbone assignment is part of an ambiguous peak system in the manual assignment. The peaks of two or more peak systems are too close for them to be separated by the spectroscopist. The third symbol "----" denotes that the peak system from the automated assignment has no matching peak system in the manual assignment. The third column in the table lists either the matching duplicate or ambiguous peak system from the manual assignment to the peak system from the automated assignment listed in the first column. The fourth column describes any discrepancies between the peaks of the automated and manual assignment.

7.2.2 CONCLUSIONS

When the two backbone assignments are compared 82% of the peak systems in the automated assignment appear in the manual assignment. If the duplicate and ambiguous peak systems are considered to be part of the manual assignment then 96% of the peak systems from the automated assignment appear in the manual assignment. The module was designed to give a reasonable start point for manual assignment. In this it succeeds, the output from BAM-1 formed the start point for the manual assignment of the FMN binding domain of cytochrome P_{450} reductase.

There are two main discrepancies between the two sets of peak systems. The first is that peaks in the manual assignment are missing from the automated assignment, while the second discrepancy occurs when the intra residue HNCA peak is absent, and the inter residue peak is used at the position of the intra residue peak. This is apparent from the matching C^{13} chemical shifts of the peak at the HNCA position and the HNCOCA.

The HNCACO peak list had peaks added to it during the later stages of manual assignment, which explains the 26 peaks that were in the manual assignment but missing from the automated assignment. The peaks were not in the HNCACO peak list given to the BAM-1 so it could not pick them.

Six peak systems in the automated assignment have the inter residue HNCA peak in the wrong position. When the intra residue HNCA peak is missing the inter residue HNCA peak can be placed in either position without violating any of the assignment rules. The inter residue HNCA peak has to conform to one less alignment rule when it is at the intra-residue position. Since the fitness in each position is inversely proportional to the sum of the distances from the alignment rules and a blank peak alignment gives zero distance, the fittest position for the HNCA peak is the one where it is subject to fewer alignment rules. The mistake is trivial as it occurred in 6 out of 184 peak systems and is usually detected easily. The C^{13} chemical shift of the inter residue HNCA peak aligns with the C^{13} chemical shift of the HNCOCA peak.

To give a complete sequential backbone assignment two additional triple resonance spectra were required, CACBNH and CACBCONH. The information from these spectra were also used to improve the manual backbone assignment. If the two spectra had been added to the input data of the BAM-1 the performance of the module would have been improved.

7.3 BAM-1 Object Definitions

This section contains the definitions of the objects for the BAM-1. The class definitions of the objects are listed below.

7.3.1 Peak Class

The peak class models a peak from any triple resonance spectrum. The peak class has as many instances as there are peaks in all the peak lists that form the main part of the BAM-1 input. The peak class models the characteristics of a peak from a triple resonance spectrum and its interaction with the other classes of the BAM-1. The attributes are listed below:

- Type attribute: this symbol states the type of spectra the peak comes from.

- d1, d2 and d3 attributes: these floating point numbers state the position of the centre of the peak in its three dimensional spectrum. The attributes will be chemical shifts measured in ppm..

- Intensity attribute: this floating point number states the intensity of a peak. The intensity of a peak is calculated from the volume the peak occupies in the spectrum.

- Aligned peaks attribute: this is an array of lists. Each list is composed of peak objects. The array is the same size as the number of peaks in a peak system. The array records the other peaks objects that can be aligned with the owner of the attribute at each position in a peak system. For example in the 3rd list in the array lists the peaks objects that align with the peak object in the 3rd position in the peak system.

- Used at attribute: this is an integer array; it is the same size as the sub population (Figure 7.1 and Section 4.5). The array will record the position of each peak in the individuals of the sub population. The integer at position 5 will record the position of the peak object in the 5th individual in the sub population. If the peak is not used in the individual then the integer at position 5 will be -1.

- Index attribute: this integer states the position of the peak object in an array of peak objects. The array will be all the peaks from the appropriate peak list. The array will form part of the peak arrays attribute of the peak lists class.

- Locked attribute: this symbol states when a peak is has been locked into a certain position by the user. When the attribute is not nil the genetic operators will not move the peak from the peak system in which it resides.

7.3.2 PEAK LISTS CLASS

The peak lists class models all the peak lists input into the BAM-1. There is only one instance of the peak lists class in the BAM-1. The attributes of the class are listed below.

- Types attribute: a list of all the types of triple resonant spectra used in the BAM-1. Each spectrum can appear more than once in the list. The number of times a spectrum appears in the list depends on the number of peaks generated per amino acid by each experiment and how many of the peaks are used in the assignment.

- Type-number attribute: an integer giving the number of spectra listed in the types attribute.

- Peak-numbers attribute: a list of numbers. Each number states the number of peaks in a peak list. The first number states the number of peaks in the peak list derived from the first spectra type listed first in the types attribute.

- Peak-lists attribute: this attribute lists several lists of peak objects. Each list of peak objects is derived from the spectrum listed in the same position in the types attribute.

- Peak-arrays attribute: this attribute is an array of arrays of peak objects. Each array of peak objects is derived from the peak list listed in the same position in the types attribute.

- Find-array: The attribute is an array of arrays of peaks. Some of the genetic operators of the BAM-1 search for a peak close to a certain position in the spectrum as part of their function. The find-array attribute increases the speed of the search operation.

7.3.3 Alignment Class

The alignment class contains the rules for determining the fitness of a solution, a backbone assignment. The rules are a list of alignments of chemical shifts for the peaks of a peak system. The rules can also state the relative intensities of peaks in a peak system. There will be only one instance of an alignment object in the BAM-1. The attributes of the class are listed below.

- Spectra-used attribute: this attribute duplicates the types attributes of the peak-lists class.

- Peak-number attribute: this attribute duplicates the type-number attribute of the peak-lists class.

- Peak-arrays attribute: this attribute duplicates the peak-arrays attribute of the peak-lists class.

- Alignment-list attribute: this is list of the rules used in the assignment of the various triple resonant spectra. The rules are in the form of a list. An example of a rule would be (0 4 d2 d2). The example rule states than the d2 chemical shift of the 1st peak in a peak system must align with the d2 chemical shift of the 5th peak in a peak system.

- Alignment-constraint-number attribute: the attribute states the number of rules in the alignment-list attribute.

7.3.4 Peak System Class

The peak system class is designed to model a peak system. There will one peak system object for each amino acid in the protein being studied. When the chromosome of an individual is expressed the peak system objects will collectively form the phenotype of the individual. The attributes of the class are listed below.

- Id attribute: the attribute identifies the peak system object to the BAM-1. The attribute will be a symbol.

- Alignment attribute: this attribute will contain the alignment object.

- Index attribute: the integer states the position of the peak system in an array of peak systems. The array will be an attribute of the peak systems object

- Chromosome-index attribute: this integer gives the position of the section of the chromosome array that is expressed to give the peak objects that comprise the peak system. The peaks are contained in the peaks attribute of the class.

- Score attribute: this floating point number contains the score of the peak system object in the form of a floating point number. The score will be ascribed to the peak system object by the fitness function.

- Peak-number attribute: this integer states the number of peaks in the peak system and the length of the array that comprises the peaks attribute.

- Peaks attribute: this is an array of peak objects. The peak objects form the peak system that forms part of the backbone assignment.

The peak-list objects are contained and ordered by the peak systems class.

7.3.5 Peak-list Class

The will be only one instance of the peak systems class in the BAM-1. The class is used to contain the phenotype expressed by an individual. The class is very simple and its three attributes are listed below.

- Number attribute: an integer giving the number of peak systems listed in the object.

- List attribute: this is a list of the peak systems in the BAM-1.

- Array attribute: an array of peak systems. The position of a peak system in the array will be the index attribute of the peak system.

8.0 Backbone Assignment Module 2

The backbone assignment module 2 (BAM-2) sequentially assigns the backbone resonances of a protein. The BAM-2 uses two sources of information: the backbone assignment from the BAM-1 and the rules on the sequential assignment of the triple resonance spectra. The BAM-2 takes this information and produces a sequential backbone assignment of the protein. As with the BAM-1 the sequential assignment rules are external to the BAM-2 to allow it to cope with the diversity of triple resonance NMR experiments.

The sequential backbone assignment of a protein using several of its triple resonance spectra is described in section 2.4. Each element of the input to the BAM-2 supplies certain information. The backbone assignment supplies the chemical shifts of the peak systems. The alignment rules state the chemical shifts of a peak system that align with the chemical shifts of the peak systems adjacent to it. The rules do this by stating that chemical shift alignments between the adjacent peak systems must be within a given tolerance. The rules also contain chemical shift ranges for each amino acid type. These chemical shift ranges are used to create a score for each amino acid identity for each peak system.

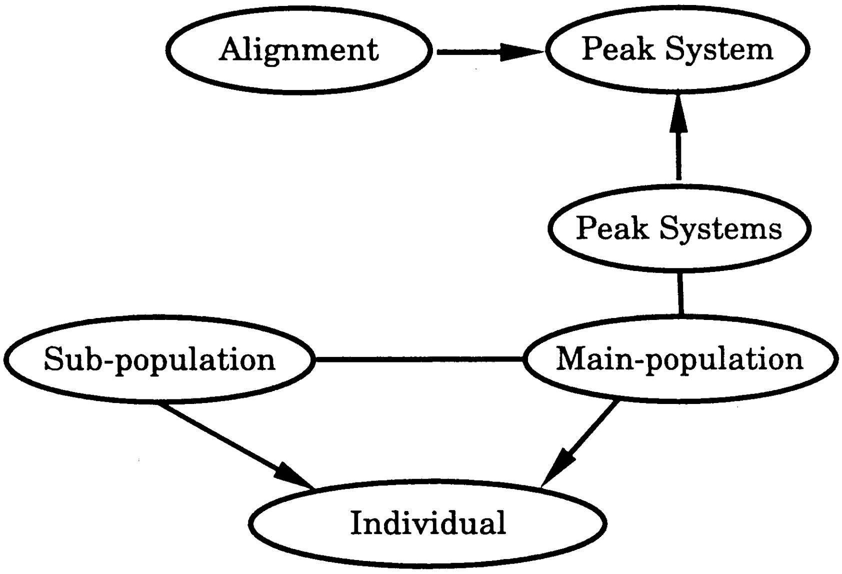
8.1 Design of the BAM-2

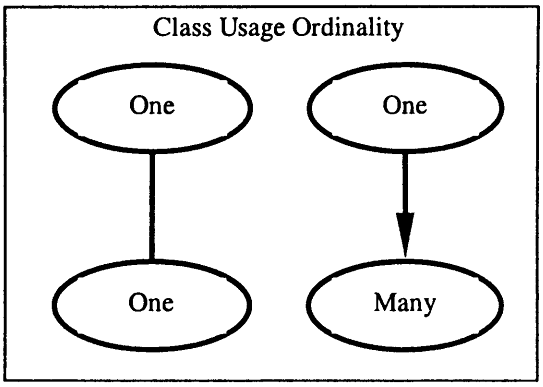
The design of the BAM-2 was produced by the separation of the BAM into the BAM-1 and BAM-2. The BAM-2 has many design and implementation elements in common with the BAM-1. The BAM-2 also has certain design concepts in common with the 2D-SAM and 3D-SAM. The BAM-2 is performing a sequential assignment just as the 2D-SAM and 3D-SAM do. The shared design concepts are mainly seen in the fitness function and genetic operators. The fitness function similarities are the concept of amino acid identity scores being combined with links to preceding and succeeding spin/peak systems. The genetic operators are similar in that they are all reordering operators. The design of the BAM-2 is an object oriented design based around the same GA core (Chapter 4) as the other modules. The design of the BAM-2 is shown in Figure 8.1. As with the other modules the population and individual classes form the GA core. The alignment, peak system and peak systems classes are similar to the classes found in the BAM-1. A new class has also been created. The new class is the assignment class. The class definitions of the BAM-2 are given in Section 8.3.

8.1.1 Coding

The coding used for the BAM-2 is similar to that of the BAM-1. The

Figure 8.1 Design of the BAM-2.



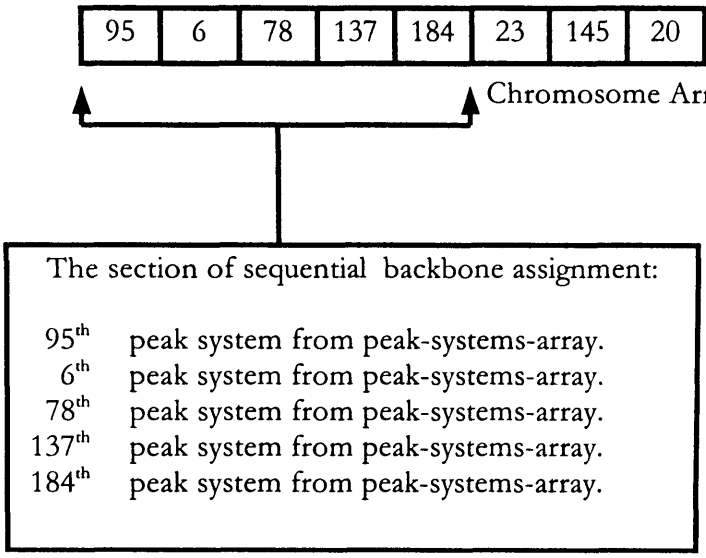


chromosome is an integer array (Figure 8.2). Each integer in the array states the position of a peak system object in the peak-systems-array attribute of the peak systems object. The integer will be the peak-system-index attribute of the peak system object. To decode or express the chromosome array of an individual the integer peaks of the array are read one after another. As each integer is read the peak system that occupies the appropriate position in the peak-systems-array attribute is found and inserted into the corresponding position in the assignment-array attribute of the assignment object. The peak system determined by the nth integer becomes the nth element of the assignment-array attribute. The assignment-array attribute is the sequential backbone assignment of the individual that has just been decoded or expressed.

The initial sequential backbone assignments are created using a variation on one of the genetic operators. Each individual in the initial population starts with a consistent and reasonable sequential backbone assignment. The consistency is ensured as with the other modules by recording the use of each peak system object. Each peak system object has a peak-system-used-at attribute. The attribute is an integer array that records the use of the object in each individual of the sub population (Sections 4.1 and 4.5). For example if the 5th integer in the peak-system-used-at array is 45 then the index of the peak system object will be the 45th element of the chromosome array of the 5th individual in the sub population. As with the other modules -1 indicates that the peak system object is not used in the relevant individual. A blank peak system object is also used for implementation reasons.

A reasonable initial sequential backbone assignment is created by randomly selecting a position on the chromosome array of an individual. A randomly chosen peak system object is then inserted into the chosen position. A variation of the linked peak systems genetic operator is then called to find an unused peak system object that could be adjacent to the randomly chosen peak system. The index of the peak system found is then written to the chromosome array adjacent to the index of the randomly selected peak system. The operator is then called again using the newly inserted peak system. This process is repeated until an unused linked peak system cannot be found. When no unused peak system is found then a blank peak system is used. The next vacant position has an unused peak system randomly inserted into it. This process is repeated until there are peak systems at each position. The used-at attributes of the peak systems are used to determine when a peak system is unused and the linked-to attribute is used to find the other peak systems

Figure 8.2 Expression of the integer chromosome array to produce a section of backbone assignment.



95	6	78	137	184	23	145	20	58

The section of sequential backbone assignment:

95th peak system from peak-systems-array.
6th peak system from peak-systems-array.
78th peak system from peak-systems-array.
137th peak system from peak-systems-array.
184th peak system from peak-systems-array.

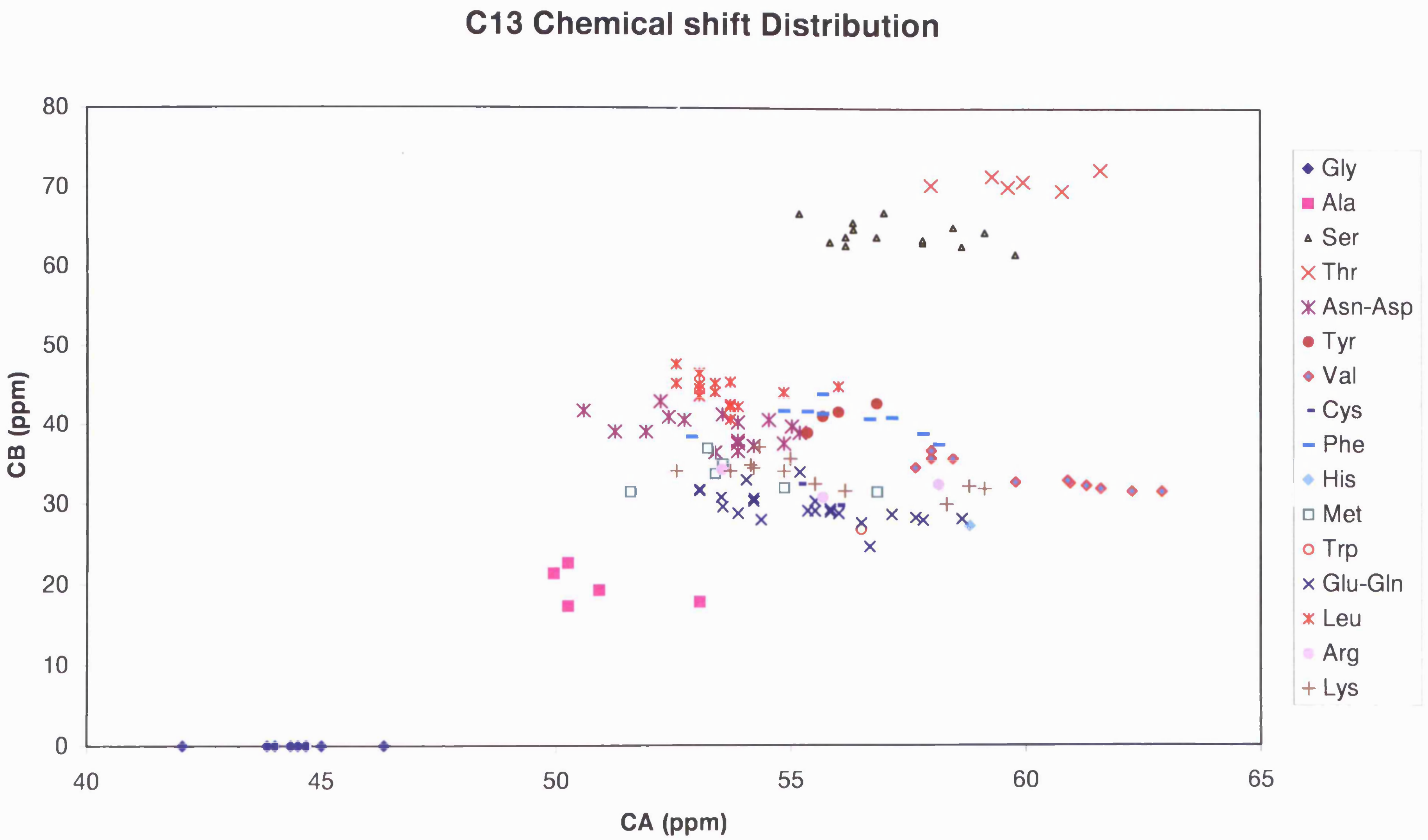
objects the object can be adjacent to.

8.1.2 FITNESS FUNCTION

The fitness function of the BAM-2 determines the quality or fitness of a sequence of peak systems contained in the assignment-array attribute. The fitness of a sequence of peak systems is the fitness of the links between the peak systems of the sequence and their fitness to occupy those positions in the assignment. Each peak system will have chemical shifts in common with the adjacent peak systems. The triple resonance experiments tend to be used in complementary pairs. One experiment will give a chemical shift within an amino acid, e.g. the $C^{13}\alpha_i$, while the complementary experiment will give the same chemical shift but of the preceding amino acid, e.g. the $C^{13}\alpha_{i-1}$. Since the appropriate chemical shift of the preceding amino acid will be known from the first experiment the sequential or adjacent peak system can be found. The fitness of a peak system to be at a specific position in the assignment is dependent upon the amino acid that generated it. This is not known but a probability that it was generated by a specific amino acid can be estimated. The $C^{13}\alpha$ and $C^{13}\beta$ chemical shifts of a peak system are, within limits, characteristic of the amino acid that generated it (Figure 8.3). These characteristic chemical shifts are however affected by the structure of the protein. The structure of a protein affects the chemical environment of a nucleus and therefore its chemical shift; the $^{13}C\alpha$ shifts are particularly sensitive to variations in secondary structure. The variation in the chemical shifts of each amino acid can be seen in Figure 8.3.

The fitness of the links between the peak systems is an estimate of how well the chemical shifts of the peak systems in the sequence align. The quality of the link between each peak system is estimated using the rules contained in the alignment object. The rules have the form (0 5 0.03). This rule states that 0th chemical shift (cs^1) of the preceding peak system must be within a certain tolerance (T),0.03 ppm, of the 5th chemical shift (cs^2) of the succeeding peak system. Ideally the two specified chemical shifts should be the same. To calculate the fitness of the rule Equation 8.1 is used. The absolute difference of the two chemical shifts is divided by the tolerance and subtracted from 1.0. The number produced is the fitness of that rule (F_r). The fitness of a link is the average of the fitness from each rule in the alignment object. If the difference between two chemical shifts for a rule is greater than its tolerance then the fitness for the whole link is zero.

Figure 8.3^{13}Cα & ^{13}Cβ chemical shift scatter diagram of Interleukin 1β.



$$F_r = 1 - \frac{|cs_1 - cs_2|}{T}$$ Equation 8.1

The other component in calculating a fitness for an assignment is the probability that a peak system was generated by the right type of amino acid. This is estimated in the amino acid identity scores of each peak system. This attribute records a score for each possible amino acid identity. The scores are determined using the amino acid identity rules located in alignment object. The rules have the form,

((Q E) (2 55.37 3.5 1.0 0.10 0.10) (6 29.43 3.5 1.0 0.10 0.10) 20.0 2.5)

Each element of the rule has the following function:

- The first section of the rule, delineated by the (), indicate that this rule generates the identity score for glutamine and glutamate amino acids.
- The next two sections, again delineated by the (), are rules for individual chemical shifts in the peak system the first one is explained below.
    - 2 indicates that this section of the rule concerns the 3rd chemical shift of the chemical shift array attribute of the peak system
    - 55.37 is the ideal value of the 3rd chemical shift of the peak system.
    - 3.5 is the tolerance, ±, that the 3rd chemical shift can have. If the 3rd chemical shift of the peak system is equal to the ideal chemical shift (55.37) ± the tolerance (3.5) then its distance factor will be 1.0.
    - 1.0 is the decrement factor. The decrement factor is used when the 3rd chemical shift of the peak system is beyond the tolerance. The tolerance is subtracted from the absolute difference between the ideal value (55.37) and the actual chemical shift. The result is multiplied by the decrement factor and subtracted from 1.0 to give the distance factor for the chemical shift. If the distance factor is less than -1.0 it is then set to -1.0.
    - 0.10 and 0.10 are the scoring factors. These are multiplied by the distance factor (described above) to give the score for that chemical shift. The first number is used when the distance factor is positive, the second when it is negative.

- The last two numbers (20.0 and 2.5) are the synergistic factors; they describe how the scores from the two chemical shift rules are combined. The first step in the combination of the summing of the two scores. If both scores are positive then sum of the two scores is multiplied by the first synergistic factor (20.0). If both are negative then the sum is multiplied by the second factor (2.5).

Finally the result of synergistic factor calculation is added to 1.0 to give the identity score for glutamine and glutamate for that peak system.

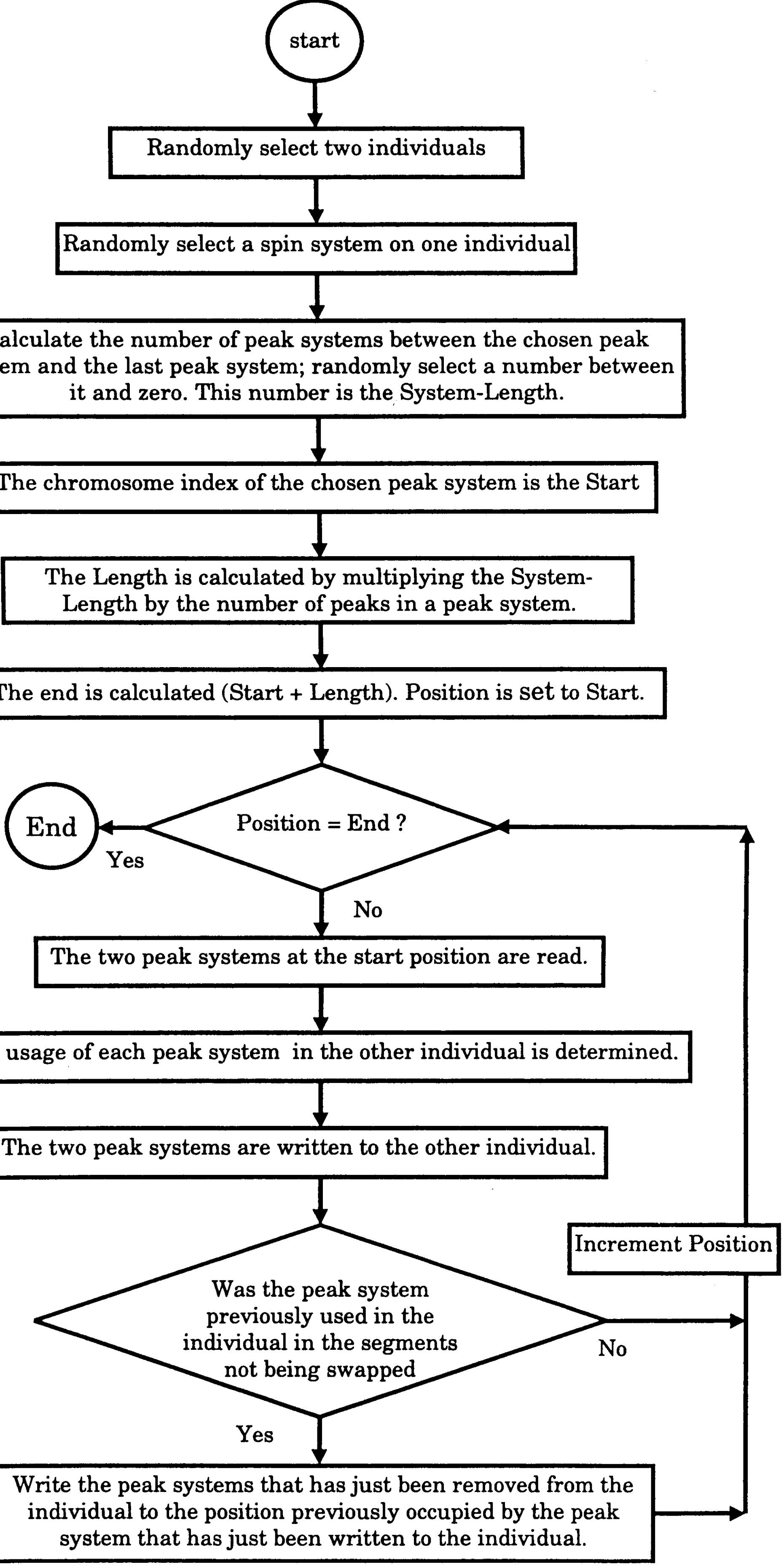
8.1.3 Genetic Operators

The BAM-2 currently uses four genetic operators: crossover, segment reordering, linked peak system reordering and the sequence reordering operator. The first two operators are relatively simple. The crossover operator swaps segments of sequential backbone assignment between individuals in the sub population while maintaining a consistent assignment. The segment reordering operator swaps segments of sequential backbone assignment within an individual. The 'linked peak system reordering' operator reorders the sequence of peak systems within an individual by swapping one or two peak systems. The sequence reordering operator reorders a sequence of peak systems in an individual. As with some of the genetic operators of the BAM-1 these two operators make use of the alignment rules either implicitly or explicitly to determine the manner of the reordering.

The crossover operator is similar to all the crossover operators in the other modules. The operator just swaps segments of sequential backbone assignment between individuals. It randomly selects the position of a peak system on the chromosome array of an individual. Another peak system is randomly chosen between the first peak system and the end of the chromosome array. The peaks of the two peak systems and all the peak systems in between are then swapped between the two individuals undergoing crossover. The crossover is done in such a way as to maintain the consistency of the backbone assignment each individual encodes. If a peak is already used in an individual then it is replaced with either a blank peak system or with the peak system that has just been overwritten by the new copy of the peak system. A flow chart of the crossover operator is shown in Figure 8.4

The segment reordering operator performs a similar function to the segment reordering operator in the sequential assignment modules (Sections 5.1.3 and

Figure 8.4 Flow chart of the crossover operator.


start
Randomly select two individuals
Randomly select a spin system on one individual
Calculate the number of peak systems between the chosen peak system and the last peak system; randomly select a number between it and zero. This number is the System-Length.
The chromosome index of the chosen peak system is the Start
The Length is calculated by multiplying the System-Length by the number of peaks in a peak system.
The end is calculated (Start + Length). Position is set to Start.
Position = End ?
Yes
End
No
The two peak systems at the start position are read.
The usage of each peak system in the other individual is determined.
The two peak systems are written to the other individual.
Was the peak system previously used in the individual in the segments not being swapped
No
Increment Position
Yes
Write the peak systems that has just been removed from the individual to the position previously occupied by the peak system that has just been written to the individual.

6.1.3). It randomly selects segments of the sequential backbone assignment and swaps the position of the segments. A peak system is randomly chosen on a chromosome array. Another peak system is chosen between the first peak system and the beginning of the chromosome array. The maximum length of segment that can be swapped is then found. This is the shorter of either the distance between the two peak system objects chosen or the distance between the first peak system chosen and the last peak system in the chromosome array. The length of the segment is randomly chosen between the maximum length and zero. The two segments of peak systems defined as starting at the two chosen peak systems and by the length chosen swap positions. The operator, by swapping segments, does what would probably not be done by swapping individual peak systems. For example, two highly fit segments of sequential backbone assignment that would confer higher fitness if swapped. The probability of the peak system operators swapping the two segments is low. The probability of swapping the peak systems of the segments sequentially is low. This probability is reduced even further when the changes are initially working against a selection pressure. There would be an initial reduction of fitness while the segments were being swapped.

The linked reordering operator is similar to both the aligned peaks operator of the BAM-1 and the spin system reordering operator of the 2D-SAM and 3D-SAM. The operator inserts a peak system into a chromosome array that can be linked to its neighbouring peak systems. The operator randomly selects a peak system on the chromosome array of an individual. The linked-to-succ attribute of the preceding peak system and the linked-to-prec attribute of the succeeding peak system are examined. The peak systems that appear in both lists are selected and one is randomly chosen. The chosen peak system is then inserted into the chosen position. If no peak system occurs in both lists apart from the peak system currently occupying the chosen position a peak system is randomly chosen from the linked-to-succ attribute of the preceding peak system. The chosen peak system is then inserted in the chosen position and succeeding peak system is replaced with a blank peak system. The position formerly occupied by the newly inserted peak is examined. If the newly overwritten peak system can occupy that position without violating any of the alignments rules it is inserted at the position. If it violates any of the alignment rules then a blank peak system is inserted. The operator implicitly makes use of the alignment rules by using the linked-to attributes. The attributes are constructed using the alignment rules. The attribute is a list of those peak systems that conform to the alignment rules that could be adjacent to the peak system.

The sequence reordering operator operates in a similar way to the linked peak system reordering operator but on a larger scale. The operator randomly selects a peak system on the chromosome array of an individual. The operator then randomly selects a linked succeeding peak system from the peak-system-linked-succ attribute and it is then swapped with the peak system that currently occupies that position. The process is then repeated for the peak system that was just inserted. The swapping process continues until one of three criteria is met: (i) the end of the chromosome array is reached, (ii) the peak-system-linked-succ list of the peak system is empty or (iii) a randomly determined number of insertions have occurred. Once the process has been terminated the process is repeated but using the peak-system-linked-prec list of the first peak system selected and swapping the found peak system with the one before it. The termination criteria for this process are the same except the beginning of the chromosome array is a termination criterion not the end of it.

Effectively the genetic operators of the BAM-2 are a blend of those of the BAM-1 and the sequential assignment modules. The BAM-1 influence comes from the fact that the two modules were originally one module and that the spectra being used are the same. The sequential assignment module influence comes from the fact that the BAM-2 is designed to produce a sequential assignment. Although the genetic operators listed above are the ones used currently several others were developed but proved to be either ineffective or to offer no tangible improvement in the performance of the BAM-2 during development.

8.2 Evaluation of the BAM-2

The BAM-2 was evaluated using the FMN binding domain of cytochrome P450 reductase. The backbone assignment of the protein, determined using HNCA, HNCO, HNCOCA, HNCACO, CACBNH and CACB(CO)NH experiments, was used to create a sequential backbone assignment. The alignment rules only made use of the ^{13}C chemical shifts to perform the sequential backbone assignment. The data is experimental and therefore imperfect. The manual assignment contains only 179 peak systems not 185, of those peak systems that it does have: 2% do not have links to either of the neighbouring peak systems, 20% have links to only one neighbouring peak system and 39 % have a missing ^{13}Cβ chemical shift.

The input data for the BAM-1 can be found in the enclosed CD-ROM.

8.2.1 Testing of the BAM-2

As with the BAM-1 a number of test runs were performed on the BAM-2 to establish the best combination of genetic operators; which operators are used and the frequency of those operators (Section 8.1.3). The first mutation rate parameter determines the frequency of use of the 'reorder peak system linked' operator. The second mutation rate parameter determines the frequency of use of the 'reorder peak system sequence' operator. The third mutation rate parameter determines the frequency of use of the 'reorder peak system segment' operator. The parameters that control the behaviour of the BAM-2 are the same as for the previous GA modules. The spectral parameters are contained in the alignment rules that form part of the alignment object. The default parameters are listed below.

- The population parameter is 1000. The main-population has 1000 individuals; the sub-population has 100 individuals.

- The generations parameter is set at 20000. The sub-population is evolved for 20000 generations; effectively 2000 generations for the main-population.

- The crossover rate parameter is set at 0.5. 50 percent of the individuals in the sub-population undergo crossover every generation.

- The mutation rate 1 parameter is set at 0.1. 10 percent of the individuals sub-population are subjected to the reorder peak system linked genetic operator.

- The mutation rate 2 parameter is set at 0.15. 15 percent of the individuals sub-population are subjected to each of the reorder peak system sequence genetic operators.

- The mutation rate 3 parameter is set at 0.15. 15 percent of the individuals sub-population are subjected to the reorder peak system segment genetic operator.

The alignment rules are in the form of a list. Each alignment rule can itself be a list. The rules are listed below.

```
(HNCA HNCOCA HNCACO HNCO CACBNH CACBCONH)
16
(((G) (2 44.24 3.0 1.0 0.20 0.20) (6 1000.0 0.01 10.0 0.01 0.25) 45.0 2.0)
 ((A) (2 50.89 3.0 1.0 0.10 0.10) (6 19.62 3.0 1.0 0.10 0.15) 20.0 2.0)
((V) (2 60.95 3.0 1.0 0.10 0.10) (6 32.84 3.0 1.0 0.10 0.10) 20.0 2.5)
((S) (2 57.26 3.0 1.0 0.10 0.10) (6 64.08 3.0 1.0 0.10 0.10) 20.0 2.5)
((T) (2 59.88 3.0 1.0 0.10 0.10) (6 70.78 3.0 1.0 0.10 0.10) 20.0 2.5)
((N D) (2 53.37 3.0 1.0 0.10 0.10) (6 39.15 3.0 1.0 0.10 0.10) 20.0 2.5)
((Y F) (2 55.97 3.0 1.0 0.10 0.10) (6 40.90 3.0 1.0 0.10 0.10) 20.0 2.5)
((C) (2 55.61 3.0 1.0 0.10 0.10) (6 31.04 3.0 1.0 0.10 0.10) 20.0 2.5)
((H) (2 58.81 3.0 1.0 0.10 0.10) (6 27.25 3.0 1.0 0.10 0.10) 20.0 2.5)
((M) (2 53.91 3.0 1.0 0.10 0.10) (6 33.31 3.0 1.0 0.10 0.10) 20.0 2.5)
((Q E) (2 55.37 3.5 1.0 0.10 0.10) (6 29.43 3.5 1.0 0.10 0.10) 20.0 2.5)
((L) (2 53.58 3.0 1.0 0.10 0.10) (6 44.05 3.0 1.0 0.10 0.10) 20.0 2.5)
((I) (2 60.40 4.0 0.5 0.15 0.15) (6 38.59 3.0 1.0 0.10 0.10) 20.0 2.0)
((R) (2 55.70 3.0 1.0 0.10 0.10) (6 32.40 3.0 1.0 0.10 0.10) 20.0 2.5)
((K) (2 55.56 3.0 1.0 0.10 0.10) (6 33.43 3.0 1.0 0.10 0.10) 20.0 2.4)
((W) (2 55.51 3.0 1.0 0.10 0.10) (6 26.73 3.0 1.0 0.10 0.10) 20.0 2.5))
((2 3 0.4) (4 5 0.4) (6 7 0.4))
```

The first line identifies the NMR experiments used. The number (16) identifies the number of chemical shifts in the peak system. The last line defines which chemical shifts must align between the current peak system and the preceding peak system in the assignment (Section 8.1.2). For each peak system there are three chemical shift alignments: the Cα-C$_{i-1}\alpha$, CO-C$_{i-1}$O and Cβ-C$_{i-1}\beta$. The data respectively comes from the following pairs of experiments HNCA HN(CO)CA, HNCO HN(CA)CO and CACBNH CACB(CO)NH. The Cα information is duplicated in the CACBNH and CACB(CO)NH spectra. The other lines are used to create the amino acid identity scores for each peak system (Section 8.1.2).

Eighteen test runs were performed with the BAM-2, the results of the runs are shown in Table 8.1. Three criteria were used to assess the performance of the BAM-2, they were:

- The percentage of peak systems in the correct position.
- The percentage of peak systems adjacent to the correct peak system; e.g. peak system valine 23 is adjacent peak system alanine 24.
- The percentage of peak systems which were assigned as being generated by the correct type of amino acid.

The experiments were run on a number of SGI workstations; with differing

Run	Peak Systems in	Peak Sytems in Realtively	Peak Systems with	Fitness of Assignment
	Correct Position (%)	Correct Position (%)	With the Correct AA ID (%)	
1	69.2	79.8	76.5	8886.0
2	80.4	86.5	86.0	8947.3
3	72.6	84.3	78.7	8953.4
4	80.4	87.7	83.7	9030.6
5	74.8	82.1	81.5	8955.2
6	90.5	90.5	94.4	9024.4
7	64.2	74.3	73.1	8888.2
8	72.6	83.2	79.8	8940.0
9	77.0	84.9	84.3	9049.8
10	79.8	86.0	84.3	9036.5
11	71.5	79.3	76.5	8924.6
12	84.3	87.7	87.7	9066.7
13	82.6	89.9	87.1	9081.6
14	72.6	83.7	79.3	9018.7
15	77.0	83.7	81.5	9107.3
16	65.9	78.7	72.0	8902.3
17	69.8	82.1	78.2	8915.3
18	63.6	81.5	70.9	8954.7
Average	74.9	83.7	80.9	8982.4
%SD	9.4	4.8	7.3	0.8

Table 8.1 **BAM-2 Results**

performance characteristics. They were all run in a low priority batch queue. Thus the speed of a run will depend on the concurrent usage of the work station and its performance (the fastest run was 11 hours 50 minutes).

The fittest individual generated during all the test runs was taken and the backbone assignment it encoded was examined further. The sequential backbone assignment generated during the fifteenth test run of the BAM-2 (Table 8.1) took 19 hours 59 and generated 77% correct backbone assignment. The correctly and incorrectly assigned peak systems were examined for the two characteristics that determine their fitness: linkage to other peak systems and amino acid identity; the results are shown in Table 8.2.

8.2.2 CONCLUSIONS

The BAM-2 is a qualified success, it generated a 75% correct backbone assignment. The percentage of peak systems with the correct amino acid identity was 81% and the percentage of peak systems in the relatively correct position was 84% (see Table 8.1). I believe that this makes the BAM-2 a useful tool for NMR spectroscopists. The performance is related to the quality of the data entered into the module.

The BAM-2 is remarkably consistent in its performance with only a 0.8% standard deviation in the fitness. The variation in the other criteria (correct position, correct amino acid identity and correct relative position) is greater (Table 8.1). The variation is due to those peak systems that can be placed in a number of different positions in the assignment which contribute equally to its fitness (Section 6.2.2). The problem is inherent in the data and has two causes. The first, and most important, is that such peak systems have poor or no links to their neighbouring peak systems. Thus a peak system can have the same linking score in a number of positions. The second is that these peak systems can have atypical $^{13}C\alpha$ or $^{13}C\beta$ chemical shifts or the $^{13}C\beta$ is missing. In the case of the atypical chemical shifts this will cause the peak system to have a very low identity score for the amino acid that generated it and possibly high identity scores for other amino acids. In the case of the missing $^{13}C\beta$ chemical shift the peak system will probably have an inconclusive identity score for a number of amino acids. The $^{13}C\alpha$ chemical shift on its own is rarely characteristic of an amino acid (Figure 8.3).

A poor linkage to other peak systems can be overcome, to a certain extent, by a good amino acid identity score. The peak system will contribute most to the fitness of the assignment when in a position where that good identity score is

Category	Correct Position	Incorrect	% Correct for Category	% Incorrect for Category
Number of Peak Systems	139	44		
Linked to both neighbors	118	26	82%	18%
Linked to one neighbor	17	13	57%	43%
Linked to no neighbors	0	5	0%	100%
Best amino acid identity	39	6	87%	13%
One of the best Identities	37	10	79%	21%
Amino acid identity > 1.0	41	16	72%	28%
Amino acid identity < 1.0	18	12	60%	40%

Table 8.2 **Run 15 Analysis (Table 8.1)**

used; this increases the probability of it being in the correct position. Conversely, a good linkage to other peak systems can overcome a poor identity score for the relevant amino acid. The peak system will contribute most to the assignments fitness when adjacent to the correct peak systems. The problem is that often the two effects combine, for example a missing ^{13}Cβ chemical shift will mean that the peak system will probably have an inconclusive set of amino acid identities and will only have, at the very best, 2/3 of the maximum link score.

When the assignment with the highest fitness was examined (run 15 Table 8.1) to produce Table 8.2 the two factors can be seen operating. 82% of the peak systems which had links to both neighbouring peak systems are in the correct position, while only 18% are in the incorrect position. Of those peak systems where the correct identity score is the best score 87% are in the correct position. 74% of those peak systems with a missing ^{13}Cβ are in the correct position. This shows that good or even reasonable links to adjacent peak systems can overcome the lack of a high amino acid identity score.

A third factor will be the amino acid sequence of the protein which will have an affect on those sequences of peak systems that are relatively correct but the whole sequence is in the wrong position. Depending on the amino acid sequence of a protein a sequence of *n* peak systems could be placed in one or more positions while contributing the same to the fitness of the assignment. This effect is exacerbated when the peak systems have a number of equally scoring amino acid identities as this will increase the number of equally scoring positions they can be placed at.

It is possible that refinement of the fitness function might reduce the problem of these 'floating' peak systems, but I think it improbable. The 'floating' peak systems are an inherent in the data supplied to the module.

In conclusion I believe that the performance of the BAM-2 is limited only by the quality of the data it uses. If, for instance, the peak systems had been identified as being generated by one amino acid or had the benefit of additional spectra (3D HCCH ^{13}C TOCSY or COSY) to aid in the creation of the identity scores this would improve the performance of the BAM-2. Or if there were more ^{13}Cβ chemical shifts (39% are missing) this would significantly improve the amino acid identity scores and thus improve the performance of the module. Finally the output from the module (Figures 8.5 & 8.6), in conjunction with a knowledge of the peak systems and the amino acid sequence of the

Figure 8.6 An example of an incorrect assigment of peak systems

```
-------------------------------------------------

Spin System 1 | G | BLANK | BLANK

1000.000 | 1000.000 | 1000.000 | 1000.000 | 1000.000 | 1000.000 |

Spin Score 0.0 | ID 0.0 | Link1.0 | Prec 0.0 | Succ 0.0 |

-------------------------------------------------

Spin System 2 | S | *SPIN-62* | SPIN-F167

  58.991 |   56.350 |  175.596 |  176.050 | 1000.000 |   31.040 |

Spin Score 8.0 | ID 1.1 | Link 7.2 | Prec 0.0 | Succ 0.6 |

-------------------------------------------------

Spin System 3 | H | *SPIN-15* | SPIN-W168

  61.073 |   59.006 |  176.881 |  175.633 |   26.620 |   38.837 |

Spin Score   36.2 | ID 5.0 | Link 7.2 | Prec 0.6 | Succ 0.0 |

-------------------------------------------------
```

The first line identifies where in the assignment the peak system is, the system identification for it and the user's identification for it.

The second line gives the chemical shifts that will align in adjacent peak systems ($^{13}C\alpha_i$, $^{13}C\alpha_{i-1}$, $^{13}CO_i$, $^{13}CO_{i-1}$, $^{13}C\beta_i$, and $^{13}C\beta_{i-1,}$).

The third line gives the overall score for the peak system, the amino acid identity score, the combined linking score, the preceding and succeeding link score

Figure 8.5 An example of a correct assignment of peak systems.

```
-----------------------------------------------

Spin System 14 | V | *SPIN-159* | SPIN-V14

  66.171 |   59.410 |  177.607 |  175.890 | 1000.000 |   35.450 |

Spin Score  6.6 | ID  0.9 | Link  7.3 | Prec   0.0 | Succ   0.6 |

-----------------------------------------------

Spin System 15 | E | *SPIN-132* | SPIN-E15

  58.785 |   66.157 |  178.957 |  177.635 |   29.480 | 1000.000 |

Spin Score    69.5 | ID  5.0 | Link    13.9 | Prec   0.6 | Succ   0.7 |

-----------------------------------------------

Spin System 16 | K | *SPIN-52* | SPIN-K16

  59.468 |   58.790 |  180.048 |  178.953 | 1000.000 | 1000.000 |

Spin Score    13.6 | ID  1.0 | Link    13.5 | Prec   0.7 | Succ   0.6 |

-----------------------------------------------

Spin System 17 | M | *SPIN-138* | SPIN-M17

  60.334 |   59.519 |  178.438 |  180.002 | 1000.000 | 1000.000 |

Spin Score    11.5 | ID  0.9 | Link    12.8 | Prec   0.6 | Succ   0.6 |

-----------------------------------------------

Spin System 18 | K | *SPIN-137* | SPIN-K18

  59.928 |   60.400 |  180.264 |  178.411 | 1000.000 | 1000.000 |
Spin Score    12.5 | ID  1.0 | Link    13.0 | Prec   0.6 | Succ   0.6 |

-------------------------------------------------

Spin System 19 | K | *SPIN-12* | SPIN-K19

  58.633 |   59.938 |  178.077 |  180.209 |   33.120 |   33.120 |

Spin Score    66.9 | ID  4.9 | Link    13.8 | Prec   0.6 | Succ   0.7 |

-------------------------------------------------

Spin System 20 | T | *SPIN-105* | SPIN-T20

  60.869 |   58.773 |  175.200 |  178.077 |   69.760 |   32.860 |

Spin Score    69.5 | ID  5.0 | Link    13.9 | Prec   0.7 | Succ   0.6 |

-------------------------------------------------

Spin System 21 | G | *SPIN-80* | SPIN-G21

  47.035 |   60.966 |  174.707 |  175.294 | 1000.000 |   70.020 |

Spin Score   131.8 | ID     10.5 | Link    12.6 | Prec    0.6 | Succ    0.5
|

-------------------------------------------------

Spin System 22 | R | *SPIN-107* | SPIN-R22

  55.170 |   47.140 |  174.597 |  174.758 |   32.860 | 1000.000 |

Spin Score    73.9 | ID  5.0 | Link    14.8 | Prec   0.5 | Succ   0.8 |

-------------------------------------------------

Spin System 23 | N | *SPIN-35* | SPIN-N23
```

```
  53.178 |   55.308 |  171.401 |  174.649 |   39.877 |   32.861 |

Spin Score     80.3 | ID   5.0 | Link     16.1 | Prec    0.8 | Succ    0.7 |

-------------------------------------------------

Spin System 24 | I | *SPIN-118* | SPIN-I24

  59.102 |   53.275 |  174.076 |  171.450 |   40.660 |   39.620 |

Spin Score    103.7 | ID   6.0 | Link     17.3 | Prec    0.7 | Succ    1.0 |

-------------------------------------------------

Spin System 25 | I | *SPIN-84* | SPIN-I25

  57.233 |   59.103 |  171.724 |  174.119 | 1000.000 |   40.660 |

Spin Score     18.4 | ID   1.1 | Link     16.0 | Prec    1.0 | Succ    0.5 |

-------------------------------------------------

Spin System 26 | V | *SPIN-100* | SPIN-V26

  58.679 |   57.330 |  175.944 |  171.786 | 1000.000 | 1000.000 |

Spin Score     14.0 | ID   1.1 | Link     12.7 | Prec    0.5 | Succ    0.6 |

-------------------------------------------------

Spin System 27 | F | *SPIN-43* | SPIN-F27

  56.719 |   58.694 |  175.996 |  175.923 |   42.470 |   32.340 |

Spin Score     36.8 | ID   5.0 | Link  7.4 | Prec    0.6 | Succ    0.0 |

-------------------------------------------------

Spin System 28 | Y | *SPIN-51* | SPIN-Y28
```

```
  52.346 |   56.818 |  173.244 |  176.027 |   40.400 |   41.960 |

Spin Score    36.3 | ID  3.8 | Link  9.7 | Prec   0.0 | Succ   0.9 |

--------------------------------------------------

Spin System 29 | G | *SPIN-103* | SPIN-G29

  46.233 |   52.447 | 1000.000 |  173.293 | 1000.000 |   40.390 |

Spin Score   128.2 | ID  10.5 | Link    12.3 | Prec   0.9 | Succ  0.3 |

--------------------------------------------------

Spin System 30 | S | *SPIN-123* | SPIN-S30

  56.081 |   46.321 |  173.603 |  172.540 | 1000.000 | 1000.000 |

Spin Score  4.0 | ID  1.1 | Link  3.6 | Prec   0.3 | Succ   0.0 |

--------------------------------------------------
```

protein, will allow an experienced spectroscopist to determine those areas that are reliably assigned and those that are not. The linking score for each peak system are displayed in the module output. This allows the strength of the assignment for each linked sequence to be determined. The amino acid identity scores of the sequence, the length of the sequence and the amino acid sequence of the protein then allow the spectroscopist to assess whether the linked sequence is in the correct position.

8.3 BAM-2 Object Definitions

8.3.1 Alignment and Peak Systems Classes

The alignment and peak systems class definitions remain the same as those used in the BAM-1.

8.3.2 Peak System Class

The modifications to the peak system class are the removal of the chromosome-index, peaks and score attributes and the addition of the shift-array, score-prec, score-succ and linked-to attributes.

- Shifts array attribute: this is an array of floating point numbers. The floating point numbers are the chemical shifts of the peak system object.

- Identity score attribute: this is a hash of floating point numbers. The keys to the hash will be the single letter amino acid identifiers and the floating point numbers will be the identity score of the relevant amino acid. The hash will default to 1.0 if no score is found for the amino acid.

- Score-prec attribute: this floating point number records the score of the link to the preceding peak system in a sequential assignment. The sequential assignment will be an array of peak systems in the assignment object.

- Score-succ attribute: this floating point number records the score of the link to the succeeding peak system in a sequential assignment. The sequential assignment will be an array of peak systems in the assignment object.

- Linked-to-prec attribute: this list records all the other peak system objects that can precede the object in a sequential backbone assignment. This is determined using the rules contained in the alignment object. Each element of the list is itself a two element list: the first is the peak system it is linked to and the second is the linking score, a floating point number.

- Linked-to-succ attribute: this list records all the other peak system objects that can succeed the object in a sequential backbone assignment. This is determined using the rules contained in the alignment object. Each element of the list is itself a two element list: the first is the peak system it is linked to the and second is the linking score, a floating point number.

8.3.3 Assignment Class

The new assignment class is designed to represent a sequential backbone assignment. The class is designed to hold sequence of peak systems derived from the chromosome array of an individual. There will be only one instance of the alignment class. The attributes of the class are listed below.

- Peak-systems attribute: this attribute will contain the peak systems object.

- System-number attribute: this integer records the number of peak systems there are in a sequential backbone assignment.

- Array attribute: this is an array of peak systems that comprises the actual assignment the class represents.

9.0 FUTURE WORK

Future work on the project will need to concentrate on three areas: the side chain assignment module, a user interface for each module and the BAM-2. There are some secondary issues still to be addressed with the other GA modules.

9.1 SIDE CHAIN ASSIGNMENT MODULE

The side chain assignment module (SCAM) was conceived as a complement to the BAM, now the BAM-1 and BAM-2. The BAM-1 and BAM-2 would assign the backbone of a protein and the SCAM would assign the side chains of the amino acids of the protein. The development of the SCAM was not completed. The module was still undergoing initial testing when lack of time forced work on the module to stop. As the SCAM is not fully developed a brief description of the module has been placed here in the future work chapter.

The SCAM is designed to use HCCH C^{13} TOCSY or COSY experiments (Section 2.5) to perform side chain assignment, the module can use either. The task of side chain assignment, although using only one NMR experiment, is the most complex of the tasks performed by the various GA modules (Section 2.5.1). Each amino acid will have a characteristic pattern of peaks with characteristic chemical shifts. The SCAM will have to recognise each pattern of peaks while allowing for the usual spectral ambiguities (Section 2.5.2). The SCAM uses the spectrum of either a HCCH C^{13} TOCSY or COSY experiment, the amino acid sequence of the protein under investigation and a description of the ideal pattern of peaks for each amino acid. The spectrum will be in the form of a peak list, as with the other GA modules. The ideal peak patterns will be defined by the a series of alignment rules. Only four amino acids cannot be distinguished using HCCH C^{13} COSY experiments, Glutamate/Glutamine and Aspartate/Asparagine.

The classes of the module are amino acid sequence, peak, peak list, amino acid, spin system, individual, sub population and main population. The classes are the same as the classes found in other modules with the exception of the amino acid and spin system classes. The class definitions can be found in Section 9.4.

The SCAM bears some resemblance to the BAM-1 in its coding, fitness function and particularly its genetic operators.

9.1.1 CODING OF THE SCAM

The coding of the SCAM is similar to the other modules in that the chromosome is an integer array. It is different in that the conversion to an assignment is more complex. Each integer defines a peak from the appropriate[32] HCCH C^{13} spectrum. The conversion of the integer array into a spin system assignment requires a chromosome array and the *spin-systems* object, the instance of the spin system list class. The spin system objects are processed sequentially. Each spin system will read a set number of integers from the chromosome array from a specified start point. The peak number attribute defines the number of integers read and the chromosome index attribute defines the start point. Each integer read is the position of a peak object in the peak array attribute of the peak list object. The peak objects found are placed in the spin-system-peaks array of the spin system object. The spin system peaks attributes of all the spin system collectively form the spin system assignment generated by the SCAM.

9.1.2 Fitness Function of the SCAM

The fitness function of the SCAM is by far the most complex of all the modules. The fitness of a spin system assignment is the sum of the fitness of the spin system objects that form the spin system assignment. To determine the fitness of a spin system objects five factors are considered:

- Blank Peaks
- Alignment Distance
- Alignment Constraints
- Position Constraints
- Shift Constraints

The blank peak factor simply determines the ratio of blank peaks to real peaks. If there are four peaks in a peaks system and one of them is a blank peak the blank peak factor will be 0.75. The factor is used to represent the fact that the more peaks in a spin system the greater the probability of it being a genuine spin system.

The alignment distance factor is calculated using the alignment rules. The factor calculates the distance from the ideal position of each peak in the spin

[32] TOCSY of COSY.

system. For each peak there is a tolerance in each dimension. The tolerances define the maximum difference that can exist between the chemical shifts of peaks that are considered to be aligned. Each alignment rule defines two peaks of a spin system and two chemical shifts of those peaks. If the two chemical shifts are within the appropriate tolerance[33] then the absolute distance between them is calculated. If the distance between the chemical shifts of the peaks is greater than the tolerance then the tolerance becomes the distance. The tolerance is considered to be the maximum distance. If one or both of the peaks referred to by the alignment rule are blank peaks then the distance is zero. Blank peaks are considered to give perfect alignments. The distance is then added to the appropriate distance total. There are three distance totals; one for each dimension of a spectrum.

Once all the rules have been evaluated each total distance (d_1, d_2 and d_3) is divided by the appropriate attribute of the amino acid object of the spin system, i.e. amino-acid-d1-factor (f_1), amino-acid-d2-factor (f_2) or amino-acid-d3-factor (f_3). The results of the three calculations are added together and subtracted from 1. The calculation is shown in Equation 9.1 where A_d is the alignment distance, d the total distance for a dimension and f the factor for a dimension. The calculation of the dimension factors is shown in Equation 9.2 where T is the tolerance for the dimension, N_T is the total number of alignment rules and N is the number of alignment rules that concern the dimension.

$$A_d = 1 - \left(\frac{d_1}{f_1} + \frac{d_2}{f_2} + \frac{d_3}{f_3} \right) \qquad \text{Equation 9.1}$$

$$f = \frac{T \times N^2}{N_T} \qquad \text{Equation 9.2}$$

An example of an alignment rule is (0 3 d2 d2 -6). The d2 tolerance will be used and the absolute difference between the two d2 chemical shifts of the 1st and 4th peaks in the spin system will be added to the d2 total distance variable (d_2 in Equation 9.2).

The alignment constraint factor (A_c) is calculated at the same time as the

[33] The appropriate tolerance will be the one that matches the first chemical shift defined in the alignment rule, e.g. rule (3 4 d1 d1 -7) will use the d1 tolerance.

alignment distance factor. The factor is the sum of the violation numbers (V) from the alignment rules that were false added to the number of alignment rules that prove to be true (N_t) divided by the number of alignment rules (N), see Equation 9.3. An alignment rule is considered to be true when the difference between the two specified chemical shifts is less than the tolerance for the specified dimension. The violation number is the -6 in the example give previously. The violation number is used to increase the impact of violating the alignment rule. Taking the example rule negative impact on violating the rule is 1 with out the violation number and 7 with it. The alignment constraint factor is used to determine what is a valid spin system while the alignment distance factor determines how good a spin system it is.

$$A_c = \frac{N_t + V}{N} \qquad \text{Equation 9.3}$$

The position constraint factor (P in Equation 9.4) is calculated using the constraint list attribute of the amino acid object of the spin system. The constraint rules state the relative positions of the peaks in a spin system. An alignment rule defines two peaks of a spin system and a chemical shift of each peak. The specified chemical shift of the 1st peak defined should be greater than that of the 2nd. The factor is calculated by adding the number of rules that are true (N_t) to the sum of the violation numbers (V) of the rules that are false and dividing by the number of rules (N), Equation 9.4.

$$P = \frac{N_t + V}{N} \qquad \text{Equation 9.4}$$

The shift constraint factor (S in Equation 9.5) is used to determine if the peaks of the spin system are in the correct region of the spectrum. The factor is calculated using the shift array attribute of the amino acid object of the spin system. The shift array defines an ideal range for each chemical shift of a peak. The chemical shift of each peak is compared against the appropriate ranges given in the shift array. If the chemical shift of the peak (s) falls within the defined range the shift constraint factor for that shift (s_f) is 1.0. A second range is calculated using the range given in the shift array. Twice the distance between the two chemical shifts of the range (r) is calculated. The distance is added to the upper chemical shift (u) of the range and subtracted from the lower (l) to give outer upper (u_o) and lower limits (l_o) (Equations 9.5, 9.6 & 9.7). If the chemical shift is on the upper or lower limit of the range the shift constraint for that shift is 1.0 and 0.0 if it is on the outer lower or upper limit.

When the chemical shift of the peak falls between a limit and an outer limit its score is linearly proportional to its distance from the limit (Equations 9.8 & 9.9). If the chemical shift of the peak is out side of the range of either the outer limits them the shift constraint factor of the shift is ascribed a value of -3. The -3 is used to negate the impact of the other chemical shifts of the peak.

$$r = 2(u - l) \qquad \text{Equation 9.5}$$

$$u_o = u + r \qquad \text{Equation 9.6}$$

$$l_o = l - r \qquad \text{Equation 9.7}$$

$$s_f = \frac{s - u}{r} \qquad \text{Equation 9.8}$$

$$s_f = \frac{l - s}{r} \qquad \text{Equation 9.9}$$

The shift constraint factor is the average of the shift constraint factors for each shift.

The five factors are then combined using Equation 9.10 to give the fitness of a spin system (F_s). The alignment constraint factor receives the bulk of the importance in the alignment from the impact of the alignment rules. The constraint factor is important in determining when the peaks of spin system object are aligned well enough to form a valid spin system. The distance component is used to determine the quality of valid spin systems. The position constraints insure along with the alignment constraints that the peaks form the pattern of peaks characteristic of the appropriate amino acid. The shift constraints insure that the pattern peaks are in the appropriate region of the spectrum. The blank peak factor insures that the more complete the pattern of peaks in the spin system object the greater its fitness.

$$F_s = BPS((A_d \times 0.2) + (A_c \times 0.8)) \qquad \text{Equation 9.10}$$

9.1.3 Genetic Operators of the SCAM

When the work on the SCAM was stopped there were 7 genetic operators being tested, though not concurrently. The operators were:

- Crossover

- Mutate
- Mutate Shift
- Mutate Aligned
- Mutate Peak Ideal Peak
- Mutate Ideal Peak
- Mutate Ideal Spin System

The crossover operator is identical in function to that of the other GA modules and is a permanent part of the SCAM. The other operators were still being evaluated when work stopped. All the operators operated on the peaks of a spin system object.

The mutate, mutate shift and mutate aligned operators all randomly chose an individual and then randomly chose a spin system from the individual. A peak from the spin system is randomly chosen and then replaced with another randomly chosen peak. The operators vary in the place from which the replacement peak is chosen. The mutate peak operator chooses a peak from the peak array attribute of the peak list object. The mutate shift operator chooses a peak from the peak array of the spin system amino acid object. Specifically the peak is chosen from the element of the array that occupies the same position as the peak. The mutate aligned operator chooses the peak from the either the peak-aligned-cross or peak-aligned-diagonal array of another peak of the spin system. The array used is dependent on the type of peak that has been selected for replacement; whether it is a diagonal or cross peak. Each element of the arrays will be an array of peaks; the peaks that align with the peak object the attribute belongs to. The element used is the one that corresponds to the peak being replaced, e.g. if the 4th peak in a spin system is being replaced then the replacement peak is selected from the 4th element of either peak-aligned-cross or peak-aligned-diagonal array.

The mutate ideal peak, mutate peak ideal peak and mutate ideal spin system operators all operate in a similar fashion. The operators randomly select the position of a peak in a spin system. Then, using the other peaks of a spin system and the alignment rules, an ideal position in the spectrum is calculated for a peak to be placed at the selected position in the spin system.

The mutate ideal peak operator randomly selects an individual and then a region on the chromosome array of the individual that encodes the peaks of spin system object is randomly selected. The peaks encoded by the region are found. One of the peaks is randomly chosen for replacement. The alignment rules and the other peaks are used to find the ideal position in the spectrum for a peak to replace the selected peak. The peak closest to the ideal position is then found and the peak is substituted for the selected peak. The mutate peak ideal spin system operator, instead of randomly choosing a peak, systematically replaces all the peaks in a spin system. The mutate peak ideal peak operator differs in that only one other peak in a spin system is used to calculate the ideal position; the operator has been superseded by the mutate aligned operator.

9.2 FUTURE WORK ON 2D-SAM, 3D-SAM, BAM-1 AND BAM-2

The 2D-SAM and 3D-SAM are complete. The only issues that could yet be explored is testing the modules on other proteins. The BAM-1 and BAM-2 similarly needs only further testing.

9.3 USER INTERFACE

Essentially the 2D-SAM, 3D-SAM, BAM-1 and BAM-2 have no user interface. To run any of the modules a command is entered into a UNIX shell e.g. "bam1 -l "run.lisp" > results/record &". This must be run from a directory that has the BAM-1 binary and run.lisp files. The run.lisp files contains all the user defined parameters. The current directory must also have two sub directories: data and results. The data directory must contain all the necessary data files and the results directory must contain a file called machine-name, this file is used to name all the results files. If any of these elements are missing the module will not run correctly if at all. One of the modules, BAM-1, has been used by a person other than myself. If the modules are to be used they must be made user friendly; ideally a graphical user interface would be developed. Alternatively the modules could be incorporated into one of the existing NMR packages. The BAM-1 can produce a file that acts as input to the NMR analysis application NMR Compass.

As part of the user interface problem is the presentation of the results of the various GA modules. Currently all the modules produce their results in a text file. The file is the assignment generated by the module with an estimate on how good each element of the assignment is. This does convey all the essential information but is not always what the spectroscopist wants from the module. Further work, in close collaboration with the NMR spectroscopists of the centre, is required to make the most of the assignment generated by the module.

9.4 SCAM OBJECT DEFINITIONS

This section contains the class definitions of the SCAM.

9.4.1 AMINO ACID CLASS

The amino acid class encodes the ideal pattern of peaks for an amino acid. There will be as many instances of the amino acid class as there are amino acids. Each amino acid object will have one or more corresponding spin systems. If there are four Lysines in the amino acid sequence of a protein there will be four spin systems that have a lysine amino acid object as their spin-system-amino-acid attribute. The attributes of the amino acid class are listed below.

- name attribute: this is the full name of the amino acid, e.g. “Tyrosine”.
- name-3l attribute: this is the three letter abbreviation of the amino acid, e.g. “Tyr”.
- name-1l attribute: this is the one letter abbreviation of the amino acid, e.g. “Y”.
- peak number attribute: this is the number of peaks an amino acid generates in the spectrum.
- d1-factor attribute: this is the sum of the tolerances of the alignments of the peaks of the amino acid in the d1 dimension.
- d2-factor attribute: this is the sum of the tolerances of the alignments of the peaks of the amino acid in the d2 dimension.
- d3-factor attribute: this is the sum of the tolerances of the alignments of the peaks of the amino acid in the d3 dimension.
- constraint number attribute: this is the number of constraints in the constraint list.

- shift array attribute: this is a three dimensional array of floating point numbers. The first dimension of the array is the number of peaks the amino acid generates in a spectrum, the peak-number attribute. The second dimension of the array is the number of dimensions in the spectrum, a HCCH COSY spectrum has three dimensions. The third dimension of the array defines the lower and upper limits of a range of chemical shifts for the specified dimension of the specified peak. For example in a HCCH COSY spin system with four peaks the array will be 4 x 3 x 2. The array gives the expected position of the peaks of the amino acid in the spectrum.

- type array attribute: this is a one dimensional array. The array length is defined by the number of peaks the amino acid generates in the spectrum. The array symbols are either C or D. The symbol C indicates a cross peak and the symbol D indicates a diagonal peak. The array defines where a diagonal and cross peaks should be placed in the spin-system-peaks attribute of a spin system object.

- alignment list attribute: this is a list of alignment rules. Each rule will itself be a list, e.g. (0 3 d2 d2 -6). The first and second elements of the list refer to two peaks of the spin system. The third and fourth elements define which chemical shifts of the defined peaks must align. The fifth element of the list is the violation number. This number is used by the fitness function when the alignment rule is violated.

- constraint list attribute: this is a list of constraint rules. Each rule is itself a list, e.g. (2 1 d1 -4). The first and second elements of the list define the peaks the constraint applies to. The third element of the list defines the chemical shift the constraint applies to. The fourth element is the violation number, which has the same function as before. The constraint defined by the example rule is that the d1 chemical shift of the third[34] peak in the corresponding spin-system-peaks attribute must be greater than the d1 chemical shift of the second peak in the corresponding spin-system-peaks attribute.

- peak array constraint attribute: this is an array of arrays of peak objects. The size of the first array will be defined by the peak number attribute of

[34] All arrays in lisp start at 0. The peaks referred to by the rules are in an array of peak objects and therefore the 2 in the rule refers to the third peak in the spin system.

the amino acid. Each element of the array will contain an array of peak objects. The array will be all the peak objects that conform to the relevant information in both the amino-acid-shift-array and amino-acid-type-array attributes. The first element of the array will be the peak objects that conform to information in the first element of both the amino-acid-shift-array and amino-acid-type-array attributes. The amino-acid-shift-array will define a chemical shift range for the three chemical shifts of the peak objects. The amino-acid-type-array will define whether a peak is a cross or diagonal peak. The purpose of the array is to provide all the peaks that can be placed at each position in the in the spin-system-peaks array of the appropriate spin system.

9.4.2 Spin System Class

The spin system class forms the spin system assignment of a protein. There will be as many instances of the class as there are amino acids in the protein. Each spin system object will have a corresponding amino acid object; the amino acid object will be an attribute of the spin system. The attributes of the spin system are listed below:

- id attribute: this identifies the spin system to the system.
- type attribute: this is the amino acid the spin system is generated from.
- amino acid attribute: this is the appropriate amino acid object for the spin system.
- chromosome index attribute: this is the position of the first integer in the chromosome array of an individual object that defines the first peak in the spin system.
- peak number attribute: this is the number of peaks in a spin system:
- peaks attribute: this is an array of peak objects; of length spin-system-peak-number. The array holds the peaks of the spin system.

All the spin system objects are collected into one object. The object is the spin systems object. The object has just three attributes. The first states the number of spin system in contained in the object. While the second and third are a list and array of the spin system objects respectively.

BIBLIOGRAPHY

1 A.E. Derome, Modern NMR Techniques of Chemistry Research, Pergamon Press (1987)

2 K. Wüthrich, NMR of Proteins and Nucleic Acids, Wiley, 1986.

3 G.C.K Roberts, NMR of Macromolecules: a practical approach, Oxford University Press (1993).

4 I.D. Campbell, *Biochemical Society Transactions* 19, 243-248 (1991)

5 M. Billeter, V.J. Basus and I.D. Kuntz, *Journal of Magnetic Resonance* 76, 400-415, (1988).

6 C.D. Eads and I.D. Kuntz, *Journal of Magnetic Resonance* 82, 467-482 (1989).

7 F.J.M van de Ven, *Journal of Magnetic Resonance* 86, 633-644 (1990).

8 C. Cieslar, T.A. Holack and H. Osckinat, *Journal of Magnetic Resonance* 89, 184-190 (1990).

9 G.J. Kleywegt, R. Boelens, M. Cox, M. Llinas and R. Kaptien, *Journal of Biomolecular NMR* 1, 23-47 (1991).

10 H. Osckinat, T.A. Holack and C. Cieslar, *Biopolymers* , Vol. 31, 699-712 (1991).

11 G.J. Kleywegt, G.W. Vuister, A. Padilla, R.M. Knegtel, R Boelens and R Kaptien, *Journal of Magnetic Resonance* series B 102, 166-176 (1993).

12 R. Bernstein, C. Cieslar, H. Osckinat, J. Freund and T.A. Holack, *Journal of Biomolecular NMR*, 3, 241-251 (1993).

13 P. Castasti, E. Carrara and C. Nicolini, *Journal of Computational Chemistry*, Vol. 11, No. 7, 805-818 (1990).

14 Č. Yu, J. Hwang, T. Chen and V. Soo, *Journal of Chemical Information*

and Computer Sciences 32, 183-187 (1992).

15 R. Wehrens, C. Lucasius, L. Buydens and G. Kateman, *Analytica Chimica Acta* 277, 313-324 (1993).

16 C. Ceislar, G.M. Gore and A.M. Gronenborn, *Journal of Magnetic Resonance* 80, 119-127 (1988).

17 D.S. Garret, R. Powers, A.M. Gronenborn and G.M. Clore, *Journal Magnetic Resonance* 95, 214-220 (1991).

18 S.A. Corne, and A.P. Johnson, *Journal of Magnetic Resonance* 100, 256-266 (1992).

19 K. Wüthrich, *NMR of Proteins and Nucleic Acids*, Wiley, 1986.

20 G.C.K Roberts, *NMR of Macromolecules: a practical approach*, Oxford University Press (1993).

21 S.J. Nelson, D.M. Schneider and A.J. Wand, Biophysical Journal 59, 1113-1122 (1991).

22 C. Redfield, In *NMR of Macromolecules: a practical approach* (G.C.K. Roberts Ed.), 88-97 (1993).

23 G.M Clore and A.M. Gronenborn, Annual Review of Biophysics and Biophysical Chemistry 20, 29-63 (1991).

24 D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989).

25 Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (2nd Ed.), Springer-Verlag (1994).

26 S. Forest, *Science* 261, 872-878 (1993).

27 D. Beasley, D.R. Bull and R.R. Martin, *University Computing* 15(2), 58-69 (1993).

28 D. Beasley, D.R. Bull and R.R. Martin, *University Computing* 15(4), 170-181 (1993).

29 P. Ross and D. Corne, *University of Edinburgh, Department of Artificial Intelligence, Genetic Algorithms Research Group*, Paper No. 94-007.

30 D.E. Goldberg, K. Deb and J.H. Clark, *University of Illinois, Illigal Report No. 91010*, (1991).

31 D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 106-120 Addison-Wesley (1989).

32 D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 122-125 Addison-Wesley (1989).

33 Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (2nd Ed.), 63-65, Springer-Verlag (1994).

34 S.W. Mahfoud, *Parallel Problem Solving From Nature*, 2, R.Manner and B. Manderick (Eds), 27-36 Elsevier Publishers (1992).

35 D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 106-120 Addison-Wesley (1989).

36 R.S. Pressman, *Software Engineering: A Practitioners Approach* (3rd Ed.), Chapter 8 239-264, Chapter 12 & 13 397-458, McGraw-Hill (1992).

37 Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (2nd Ed.), 95-104 Springer-Verlag (1994).

38 L.Y. Lian, et al, *Biochemistry* 30, 5335-5340 (1991).

39 M.D. Carr et al, *Biochemistry* 30, 6330-6341 (1991).

40 C. Redfield, In *NMR of Macromolecules: a practical approach* (G.C.K. Roberts Ed.), Chapter 4 71-99 (1993).

41 M.D. Carr et al, *Biochemistry* 30, 6330-6341 (1991).

42 I. G. Barsukov et al, in preparation.