The Application of Artificial Intelligence Techniques to a Sequencing Problem in the Biological Domain.

Joan D Walker

A thesis submitted in partial fulfilment of the
requirements of the University of Abertay Dundee
for the degree of Doctor of Philosophy

June 1995

I certify that this thesis is the true and accurate version of the thesis approved by the examiners.

Signed .[redacted]......... Date..27/10/95

(Director of Studies)

ABSTRACT

Determining the sequence of genes along a region of DNA from the results of experimental data is a difficult task called Map Assembly. A map indicates the order of the genes and other markers called restriction enzymes. It is a time consuming activity, carried out manually by the geneticist. The data from which maps are produced contain a high degree of error, due to experimental limitations, and several feasible solutions may be constructed from the same data. Distinguishing between competing solutions relies on the geneticist's subjective judgement. Although computer applications have been developed for map assembly they have been either restricted in the amount of data that could be handled or they addressed related problems.

This thesis has investigated and developed suitable computer techniques for automating map assembly. A novel objective method for evaluating maps was devised that was based on the expert's heuristics. The method was successful in identifying optimal maps. A new search technique based on a form of genetic algorithm(GA) was developed to generate potential maps from a set of experimental data. The objective system for evaluating maps was incorporated into the GA. Optimal gene maps could be generated automatically, then merged together to produce a multi-gene map. In many cases, the sequence of genes and restriction enzymes was very close to the sequence as determined manually by the geneticist but could be produced in a fraction of the time.

ACKNOWLEDGMENTS

I am grateful to my supervisors Dr Portia File, Dr Colin Miller and Dr Bill Samson for their support and guidance throughout the project. I would like to thank other members of the Department of Mathematical and Computer Sciences for their assistance and advice, notably Ms Pat Dugard and Mr George Collie. Thanks also to my external advisors, Dr Sue Povey from the MRC, London and Dr Wendy Bickmore from the MRC, Edinburgh for giving up their time. And finally, thanks to Tom Melvin for providing hardware and maintenance and to Gavin Walker for providing hardware and for his encouragement.

TABLE OF CONTENTS

Walker, J.D., File, P.E., Samson W.B., Miller, C.J., 1992. GENIE: A Genetic Algorithm Approach to Handle Noisy Data in the Biological Domain. In: Papers of the IEE Colloquium on Genetic Algorithms for Control Systems Engineering, May 1992, London:IEE, 11/1-11/5

Walker, J.D., File, P.E., Samson W.B., Miller, C.J., 1993. The GENIE Project - A Genetic Algorithm Application to a Sequencing Problem in the Biological Domain. In: Proceedings of the International Conference of Artificial Neural Nets and Genetic Algorithms, Innsbruck 1993, edited by R.F.Albrecht, C.R.Reeves and N.C.Steele. Vienna: Springer-Verlag, 552-558

Walker, J.D., File, P.E., Samson W.B., Miller, C.J., 1994. A Hybrid Genetic Algorithm Application to a Genetics Sequencing Problem. In: Papers of the IEE Colloquium on Molecular Bioinformatics, February 1994, London:IEE, 7/1-7/12

Walker, J.D., File, P.E., Samson W.B., Miller, C.J., 1994. Building DNA Maps: A Genetic Algorithm Based Approach. In: Advances in Molecular Bioinformatics, edited by S.Schulze-Kremer. IOS Press, 179-199

TABLE OF FIGURES

TABLE OF TABLES

CHAPTER 1 - INTRODUCTION

1 Introduction

With the introduction and increasing capability of digital computers, the range and scale of problems that can be tackled by automatic means has increased dramatically. However, there are many problem solving activities which present challenges for automation for example; those which rely on judgement, reasoning or knowledge. Currently, humans are far better at solving these types of problem than computers. The field of Artificial Intelligence (AI) emerged to "study .. how to make computers do things which at the moment people do better"(Rich 1991).

A major problem solving exercise is underway in the field of genetics called the Human Genome Programme(HGP). It is an international research project scheduled to be complete in the year 2006 at an estimated cost of $3 billion. As a result of the HGP, vast amounts of data have been generated which present many challenges to researchers in the field of AI as reviewed by Hunter(1993). One of these is determining the sequence of genes and other markers on the DNA from the results of experimental, error-prone data. The process is known as "map assembly". It is a difficult, time-consuming activity that is carried out manually by the geneticist and relies exclusively on their judgement. Previous computer applications developed were either restricted in the amount of data that could be handled or addressed related problems.

The aim of this thesis was to develop computer techniques for tackling highly constrained, combinatorial optimisation problems and to evaluate the techniques in the context of the map assembly problem.

2 Map assembly

Geneticists are attempting to determine the sequence of the 100,000 or so genes along the human DNA. To work out the order of genes, many copies of DNA are broken up into fragments using substances called restriction enzymes(REs) that cleave the DNA at specific points. By using various restriction enzymes, separately and together, a number of DNA fragments are obtained. The fragments containing the genes of interest are highlighted and their lengths are calculated. An example of the experimental data (taken from Sefton et al(1990)) obtained for three genes (namely PIL, PI and AACT) is shown in table 1.1. The sequence of the genes and RE cut sites is determined by assembling the fragments together and is referred to as a "map". (A "map" can either be a "multi-gene map" which indicates the sequence of more than one gene and restriction enzyme cut sites or a "single gene map" which indicates the sequence of cut sites around one gene.) The map assembled by the geneticist using the data in table 1.1 is shown in figure 1.1. The process of assembling the map in figure 1.1 from the experimental data in table 1.1 is illustrated in figure 1.2. (A glossary of biological terms is contained in Appendix I.)

Map assembly is difficult because of the errors present in the number and lengths of the fragments due to experimental limitations. Sometimes the REs do not always cut when they should. This can produce several fragments of different lengths that overlap one another.

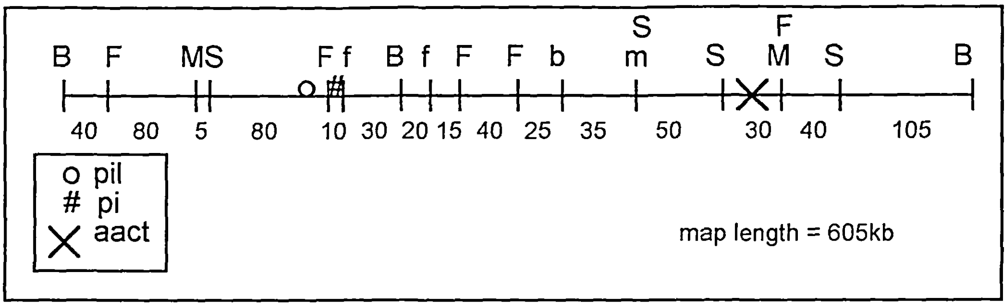


Figure 1.1 - The map published by Sefton et al(1990) showing the sequence of the three genes PIL,PI,AACT and the restriction enzyme cut sites. The numbers underneath the map indicate the distance between the cut sites. Four restriction enzymes were used, B,M,S and F. Cut sites in upper case indicate sites that always cut the DNA. Cut sites in lower case indicate sites that only cut some of the time. The map was assembled from the experimental data shown in Table 1.1 below and an example of this process is illustrated in Figure 1.2.

B			**M**			**S**			**F**		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	355	350	350	350	255	255	70	165	(190)	(195)
225	225	275	260	260	80				(10)	(135)	(175)
	(65)	230		(180)						80	135
										65	
										10	

B/M			**B/S**			**B/F**			**M/S**		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	230	125	125	70	165	(190)	135	255	255	70
225	225	105		(65)		(10)	50	105		(180)	30
130	130						10				
	(65)										

M/F			**S/F**		
PIL	PI	AACT	PIL	PI	AACT
85	80	135	70	(130)	30
(10)	65	85	(10)	80	
	10			65	
				10	

Table 1.1 - The experimental data for the PIL,PI,AACT genes. Four restriction enzymes were used B,M,S and F and were applied on their own and in combinations. Each number in the table represents a fragment of DNA that contains the gene shown. Fragments in brackets indicate suspect fragments. Assembled together by the geneticist, these fragments produced the map shown in Figure 1.1. An example of the map assembly process is illustrated in Figure 1.2.

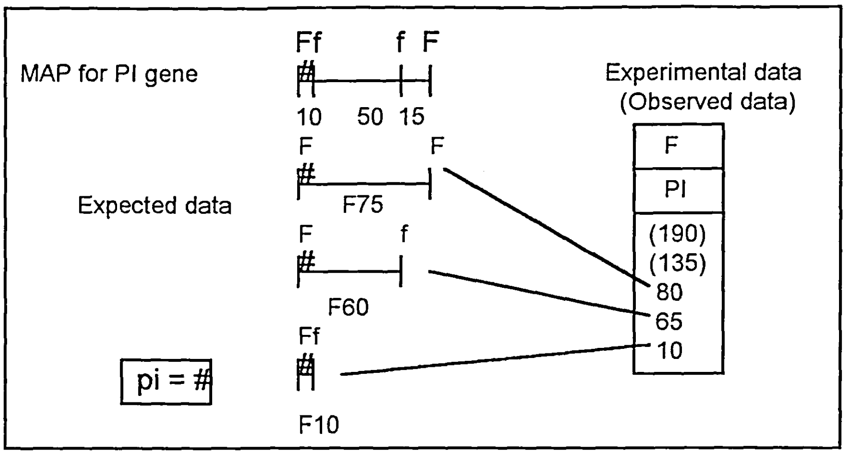


Figure 1.2 - An example of the map assembly process.

The fragments observed containing the PI gene using restriction enzyme F are shown on the right hand side of the figure (taken from Table 1.1). Note that fragments surrounded by brackets indicate "weak" fragments (ie suspect fragments) and are not used to assemble the map initially. A section of map has been extracted from Figure 1.1 illustrating the number and sequence of the F cut sites around the PI gene. The fragments expected (or derived) from the map are drawn.

The sample of total DNA used for the experiment would contain the DNA from many cells and as a consequence, there would be many fragments of DNA containing the PI gene. On some of the DNA fragments the restriction enzyme F has not always cut at the F cut sites. Such cut sites are referred to as "partial cut sites" (as explained in Chapter 2, section 2.2.1 and illustrated in Chapter 2, figure 2.4). Restriction enzyme F has two partial cut sites shown in lower case in figure 1.2. When the site is not cut, longer fragment lengths are obtained. For example, if the cut site on the right hand side of fragment length 10 had not cut, a longer fragment (F f 60) or (F F 75) would result. The fragment would be length 60 if the right hand cut site **had** cut. If the right hand cut site **had not** cut, the fragment would be length 75. As the sample of total DNA contains many fragments with the PI gene, all the different fragment lengths (10,60,75) would be generated and would all overlap one another in the region of the gene. Note that there would be other DNA fragments generated from the F cut sites shown in figure 1.2 namely fragments (f f 50) and (f F15). However, the experimental technique only highlights the fragments that contain the gene so these fragments would not be detected.

When assembling a map, the geneticist makes various judgements regarding fragment lengths as the number of observed fragments and their lengths tend to be inexact due to limitations of the experimental process. Observed fragment lengths are shortened or lengthened if necessary to make them fit into a map. In this example, the smallest fragment observed containing the PI gene (F f 10) fits exactly into the map and is the fragment that would be expected. However, the observed fragments (F f 65) and (F F 80) would be expected to be length 60 and length 75 respectively from the map.

As a consequence, there are many ways that the fragments can be assembled which can produce many different sequences of genes and cut sites. Out of these, several sequences of genes and cut sites may be feasible. To discriminate between competing maps relies on the geneticist's subjective judgement. Kearney et al(1987), Cox et al(1987), Sefton et al(1990) and Billingsley et al(1993) have all published different maps for the PIL,PI,AACT genes and RE cut sites. Although all the genes and RE cut sites have an exact position and order on the chromosomes, it is not likely that this definitive sequence can be established due to the inaccuracies in the experimental data.

3 Strategy for automating map assembly

The strategy developed in this thesis for automating map assembly is based on an approach often used by the geneticist. It was proposed that the overall problem of generating and evaluating multi-gene maps would be broken down into the simpler, yet still complex, problem of generating single gene maps. A single gene map shows the number, sequence and position of RE cut sites around one gene. Optimal single gene maps would be aligned with one another at a suitable position and would be merged to produce the complete multi-gene map. The multi-gene map shows the number, sequence and position of all the RE cut sites and the genes.

An objective mechanism for assessing maps was essential to allow existing maps to be validated and to enable an objective assessment to be made of potential new maps.

To generate potential single gene maps from a set of experimental data, some type of search strategy was required. Systematic search techniques that exhaustively tried all possibilities to arrive at the optimal solution were discounted due to the unacceptable length of time that such processes would take. Heuristic search techniques that overcome the problem of combinatorial explosion were considered. Although not guaranteed to find the optimal solution, they generally can arrive at very good ones in an acceptable length of time. One of the modern heuristic methods suitable for combinatorial problems (as reviewed by Reeves(1993a) is the Genetic Algorithm(GA). GAs (Holland(1975)) are heuristic search procedures that are based on the mechanics and analogy of natural selection. They have been shown to be effective for many difficult problems where there are a large number of possible solutions. It has been proved that a GA carries out the trade-off between adequately exploring different possibilities and exploiting the best ones found so far, in a near optimal way (Holland (1973), (1975)). The traditional GA was developed as a "weak" method (a general method that can be applied to a wide variety of problem domains requiring little or no problem specific knowledge), but has not been effective for sequencing problems as illegal potential solutions could be generated. Various researchers have investigated hybridising the traditional GA with problem specific features for sequencing problems and their results have been encouraging. It was proposed that a hybridised form of GA be developed to generate single gene maps.

Having generated optimal single gene maps, some method for aligning and merging the maps together was necessary to produce a multi-gene map.

4 Summary

Determining the sequence of genes and other markers along the DNA, known as map assembly, is a time-consuming, complex, ordering problem that relies on the geneticist's assessment of error-prone data. Many solutions may be feasible and there is no objective method of assessing maps. The strategy for automating the problem required some objective means of evaluating maps; some means of generating potential single gene maps from a set of data using a modified form of genetic algorithm; and a method for merging single gene maps together to produce multi-gene maps.

The overall aims of the project were: to investigate suitable computing techniques for automating highly constrained, combinatorial optimisation problems and to evaluate these techniques using the map assembly problem.

5 Thesis outline

Map assembly is introduced in chapter 2. The reasons why the process is difficult are examined. The complexity of the problem is analysed in terms of the number of maps possible for a given set of data. Previous computer applications are reviewed. Search techniques and in particular GAs are introduced in chapter 3. The overall strategy proposed to automate map assembly is presented. Developing an objective system for assessing maps is described in chapter 4. The results of applying the system are given in chapter 5. Developing a modified form of GA is discussed in chapter 6 and the results of using the modified GA to generate gene maps are shown in chapter 7. The results of merging individual gene maps to produce multi-gene maps are presented in chapter 8. A general discussion of the results is provided in chapter 9, the summary is in chapter 10 and conclusions are drawn in chapter 11.

CHAPTER 2 - THE MAP BUILDING PROBLEM

1 Introduction

A map shows the order and position of genes and restriction enzyme(RE) cut sites and their distances from one another. Maps are assembled from experimental data. The quality of the data depends on the type of DNA available. If the particular region of DNA of interest has been reproduced in the laboratory ("cloned") the experimental data obtained is highly accurate. When cloned DNA is not available, the total DNA of a cell must be used and additional experimental processes are necessary that complicate the process and introduce errors. The experimental techniques used and the limitations of the process are described. The number of maps possible from a set of data is considered and a formula for calculating the minimum number devised. Previous attempts of applying computing techniques to the problem are reviewed. A glossary of biological terms is contained in Appendix I.

2 DNA Mapping Techniques

Each of the 100,000 or so genes has a specific position on one of the 23 pairs of chromosomes in the human DNA. DNA is a composite structure, partly consisting of a series of four bases, namely Adenine, Guanine, Cytosine and Thymine as shown in figure 2.1. Each gene consists of a specific sequence of thousands of bases. As of August 1990, the positions of almost 5000 genes on the chromosomes had been established. There are various techniques which can be used to map genes on chromosomes - the choice of technique being influenced by the type of DNA available, either cloned DNA or total DNA. The experimental techniques of using restriction enzymes and Gel Electrophoresis are employed for both types of DNA. The additional techniques of using radioactively labelled probes and Southern Blotting are necessary for total DNA.

2.1 Restriction Enzymes

Restriction enzymes (REs) are substances that cleave the DNA when they recognise a specific sequence of bases. This produces different lengths of DNA fragments. There is a wide range of REs - each recognises a particular sequence of bases. When a piece of cloned DNA is cut by a RE, the lengths of the fragments that result always add up to equal the original length of DNA. For example, if a piece of DNA of 5000 kilobase pairs(kb) was mixed with a RE which recognised the base sequence, say, "ATCC", the strand shown in figure 2.2 would be cut into four fragments. Each fragment would consist of a different number of bases and would therefore have a different weight. The fragments could be separated in terms of weight by a technique called Gel Electrophoresis. If the fragments were put on a column of gel and an electric current applied, the fragments would move down the gel at a speed inversely proportional to their weight. The fragments would be visible as dark bands. Comparing the position of the bands with a calibrated column of gel (obtained by using standard fragments of known size), it would be possible to determine the weight and therefore the number of bases in each fragment as shown in figure 2.3. Using single REs to cut DNA is known as single digestion. Single digest fragments have been cut at each end by that RE.

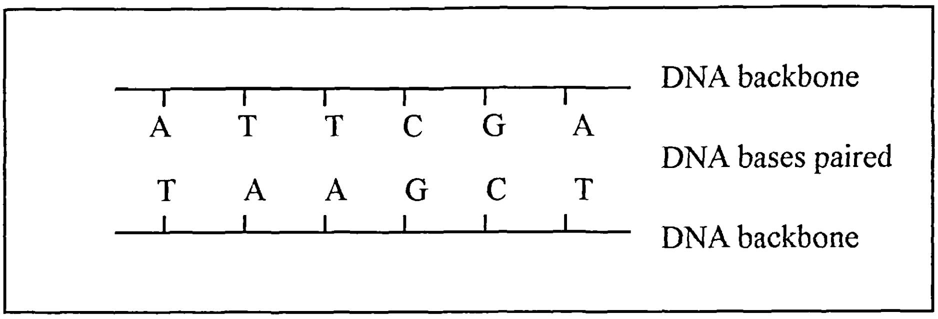


Figure 2.1 - A double strand of DNA.

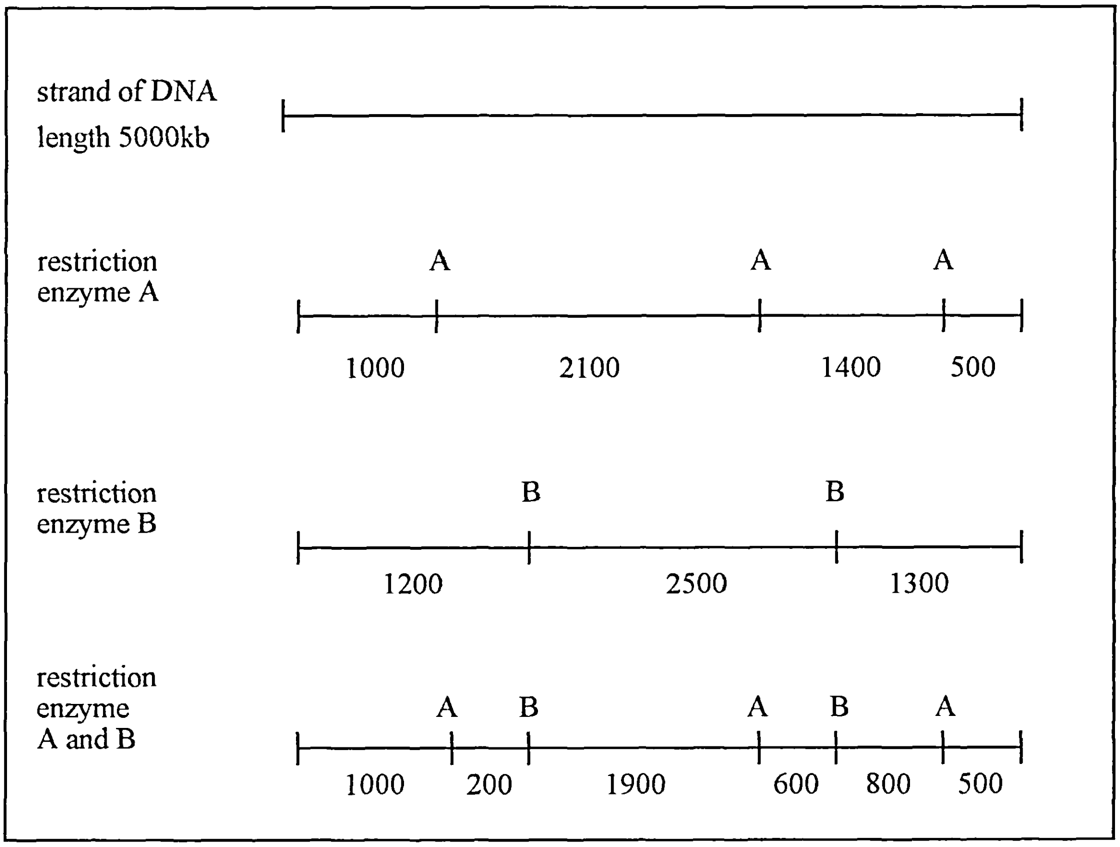


Figure 2.2 - Cleavage of DNA strand by two restriction enzymes.

When restriction enzyme **A** is used to digest DNA 5000kb, there are three **ATCC** sites that it recognises and cuts the DNA at those points, producing four single digest fragments. When restriction enzyme **B** is used to digest DNA 5000kb, there are two **CTGC** sites that it recognises and cuts the DNA at those points, producing three single digest fragments. If both restriction enzymes are applied together, six double digest fragments are produced. The lengths of the fragments are determined from gel electrophoresis. Note that the number of restriction enzyme cut sites depends on the number of fragments obtained and the sequence of cut sites must be deduced from the single and double digest results.

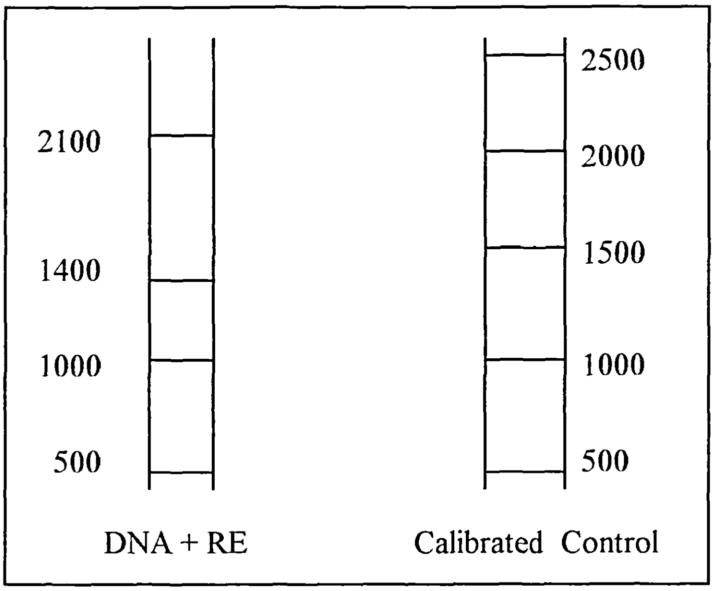


Figure 2.3 - Gel columns used in electrophoresis.

In the first column a sample of DNA of 5000kb was mixed with restriction enzyme **A** and the fragments were separated using gel electrophoresis. Four bands were visible on the gel, indicating the presence of four fragments. The fragment lengths were calculated from the calibrated control. Note that the fragments lengths add up to equal the initial length of DNA at 5000kb.

Single digests show how many occurrences there are of a particular base sequence, or "recognition site", on the DNA. However, it is difficult to determine the sequence of genes on the basis of the single digest results alone.

Using two REs to cut DNA is known as double digestion. Double digest fragments have been cut at each end by one or other of the REs. Double digest information is crucial to allow the ordering of fragments to be deduced. The key to assembling a map is to identify the common fragments. For example, in the map shown in figure 2.2, the double digest fragment of length 200 is common to the A single digest of length 2100 and the B single digest of length 1200. By using a number of different REs, a map can be assembled for a piece of cloned DNA.

2.2 Constraints using total DNA

When cloned DNA is available, the experimental method is straightforward and the sequence of the genes and RE cut sites can be deduced from the single and double digest data. However, only a very small proportion of the total human DNA has been cloned. In the absence of cloned DNA, all the DNA in a cell, the total DNA, must be used. Maps can not be assembled in the same way for total DNA as for cloned DNA for two main reasons. Firstly, REs do not always cut at their recognition sites when total DNA is used which leads to the occurrence of overlapping fragments called partial digests. Secondly, the vast amount of DNA present in total DNA can not be manipulated as easily as the small section of cloned DNA and the extra experimental processes of Southern Blotting and radioactively labelled probes are required.

2.2.1 Partial digests

All the bases in cloned DNA are in a particular state - unmethylated. When REs are used, they always cut cloned DNA at their recognition sites. The "nature" of recognition sites that are always cut are said to be "complete". In total DNA, some bases are unmethylated but others are in the opposite state - methylated. Methylation blocks the ability of a RE to cut at recognition sites and longer fragments are obtained. This phenomenon is known as partial digestion and the "nature" of recognition sites that are not always cut are said to be "partial". All fragments obtained containing a particular gene must overlap each other in the region of the gene. As there is usually more than one sequence of RE cut sites that can produce the same number of fragments, determining the correct number and sequence of cut sites is problematic as illustrated in figure 2.4.

2.2.2 Southern Blotting and Gene Probes

The sizes of fragments obtained using total DNA cannot be determined using Gel Electrophoresis alone. Due to the sheer quantity of DNA on the gels it is impossible to identify individual fragments. To isolate the fragments containing the genes of interest, a technique called Southern Blotting (Southern(1975)) is used in conjunction with radioactively labelled gene probes. Southern blotting is the process of transferring the DNA fragments from the gel to another medium, a nitrocellulose filter, to allow the fragments containing the genes to be identified.

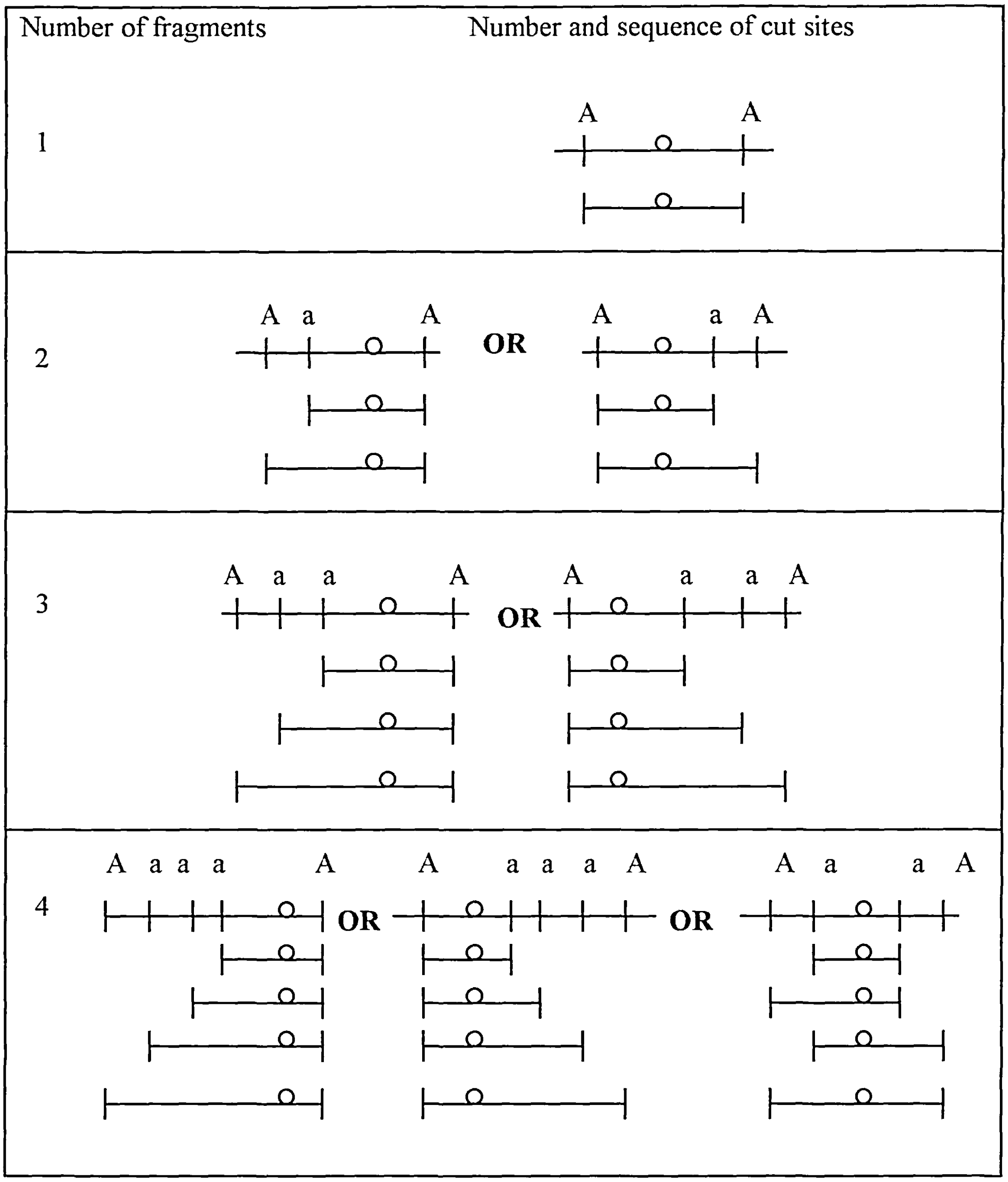


Figure 2.4 - The number and sequence of cut sites required to produce 1, 2, 3 and 4 fragments containing the same gene (shown as a circle).

"Complete" cut sites (sites that always cut), are indicated in upper case. "Partial" cut sites (sites that sometimes cut), are indicated in lower case. The presence of partial cut sites results in different lengths of fragments containing the same gene. All fragments containing the same gene must overlap in the region of the gene. The presence of a complete cut site prohibits the generation of any longer fragments. Note that four fragments can be produced from either five or four cut sites and that the sequence of cut sites around the gene are in a different order.

Agents called probes can be manufactured that will bind to particular genes. These are radioactively labelled and when introduced on the nitrocellulose filter bind to the specific gene. A technique called "autoradiography" is used which detects the radioactivity emitted from the gene probe and enables the lengths of the fragments on which the probe resides to be determined. If there are no partial digests, the probe binds to fragments of the same length containing the gene. When there are partial digests, the probe binds to fragments of different lengths that contain the gene. In order to relate the position of fragments and genes with one another, additional probes are introduced onto the filter for each gene.

2.2.3 Limitations of the experimental process

There are several aspects of the experimental process which contribute to poor data.

Different REs operate best under different experimental conditions. For single digest experiments the ideal conditions can be created; however for double digest experiments a compromise must be reached. As a consequence, the double digest results are not generally as accurate. It is possible to lose small fragments (approximately < 50kb), particularly double digests, as they can run off the end of the gel columns. This is why the number of double digest fragments observed experimentally tends to be less than the number of double digest fragments expected from a map.

The gel columns represent the visual information from which the geneticist subjectively determines the number and lengths of fragments present. Sometimes, the gel columns become crooked due to the effect of variations in density in the gel. In addition, the shape of the fragment bands in the gel columns can vary considerably. Instead of being straight, the bands can be circular or triangular. When bands are very faint they are referred to as "weak" fragments. Although these may be proper fragments, they may be due to smudging on the gel so tend not to be used to assemble a map. A "good" map is one in which the weak fragments fit well. Errors in the lengths of fragments can also be contributed by the phenomenon of "DNA loading". Although the same amount of DNA may be used, in some cases, the DNA may be more tightly packed than others which affects the speed at which it moves down the gel.

The lengths of fragments are determined from the size markers used for the calibrated control. There is a lack of size markers available for fragment lengths exceeding 1000kb and as a result, there is more error associated with determining the lengths of longer fragments.

As a consequence of the poor quality data and due to the overlapping nature of the fragments, many feasible map solutions may be generated from the same data. Kearney et al(1987), Cox et al(1987), Sefton et al(1990) and Billingsley et al(1993) have all published different maps for the PIL,PI,AACT genes and RE cut sites. There has been no objective means of discriminating between maps.

3 Problem search space

Solving a problem can be viewed as a search through the space of possible alternatives to find the "best" solution. The "search space" for the map assembly problem is all the possible maps that can be generated from a set of data. The starting state is the observed experimental data which consists of single and double digest fragment lengths for several genes and various REs as shown in chapter 1, table 1.1. Taking the data, the geneticist assembles the fragment lengths to produce a map that is considered to represent the best fit of the data. In doing so, various decisions and judgements are made regarding the error present in the fragment lengths; the choice of RE cutting in the double digests; and the nature of the cut sites. The goal state is a map which indicates the position and sequence of the genes and RE cut sites. An example of a map generated from the data shown in chapter 1, table 1.1 was shown in chapter 1, figure 1.1.

3.1 Calculating the number of possible maps

The number of possible maps that can be generated from a set of data depends on the number of genes and REs used and the number of fragments obtained. A general procedure has been devised to calculate the minimum number of maps possible from the data for a **single gene** and this is described in table 2.1. It is shown in table 2.2 that for a simplified, restricted problem instance, the number of maps possible for a single gene is an exponential function of the input. The number of maps possible for several genes would be much greater.

3.2 Calculation of search space for test data sets

Three sets of experimental data were used in the project (shown in Appendix A): the PIL/PI/AACT data; the "perfect" PIL/PI/AACT data; and the AT/ACE data.

1. **Calculate the number and sequence of cut sites for each RE.**

Calculate the number and sequence of cut sites required to produce the single digest fragments for each restriction enzyme. The number of fragments (n) is a result of the number and sequence of cut sites (as illustrated in figure 2.4) and can be calculated from the rules listed below.

i) (n) fragments can always be obtained from (n + 1) cut sites.

ii) If (n) is prime, (n) fragments can only be obtained from (n + 1) cut sites.

iii) If (n) is divisible by p, (n) fragments can be obtained from (p + n/p) cut sites.

[The position of the cut sites in cases i) and ii) are always of the form (A * _ A) or (A _ * A) and for iii) is (A _ * _ A), where * represents the gene, _ represents any number of partial cut sites.]

2. **Determine the map options from combining the cut sites.**

The sequence of cut sites for the individual REs must be combined together to produce the sequence of cut sites around one gene. It is possible that cut sites can be arranged around the gene in more than one way - each alternative is referred to as a map **option.** The number of map options must be calculated.

Eg. If RE A produced two fragments for a gene, and RE B produced three fragments for a gene, there would be four map options possible -

A a * A + B * b b B = A a B * A b b B (option 1)
A * a A + B * b b B = A B * a A b b B (option 2)
A a * A + B b b * B = A a B b b * A B (option 3)
A * a A + B b b * B = A B b b * a A B (option 4)

The two A single digests and three B single digests could be produced by any of these map options. (Note partial cut sites are always shown in lowercase.)

3. **Determine the number of permutations per map option.**

For each map option, calculate the number of permutations per option. The only restraint was that a complete cut site could not be located nearer to the gene than any of its partial cut sites (eg if RE A produced two fragments, (a A * A) would be illegal.)

The number of permutations per option could be calculated using the formula:

$$\frac{(m+n)!}{m!\,n!} \times \frac{(p+q)!}{p!\,q!}$$

where m = number of cut sites of RE A on LHS of gene
n = number of cut sites of RE B on LHS of gene
p = number of cut sites of RE A on RHS of gene
q = number of cut sites of RE B on RHS of gene

4. **Determine the total number of maps possible.**

Add up the number of permutations for each map option.

In addition, the total number of permutations must be multiplied to allow each of the outer cut sites of each RE to be partial. (In theory, all outermost cut sites ought to be complete, however in practice, they may be partial.)

Table 2.1 - General procedure for calculating the minimum number of maps possible for a single gene.

* Consider the case of : 1 gene

2 restriction enzymes, A and B
having (q) and (s) single digest fragments
produced by (q + 1) and (s + 1) cut sites respectively
the sequence of cut sites being of the form (A * _ A)
where "_" represents any number of partial cut sites
" * " represents the gene

* Combining the sequence of cut sites for A and B, there would be 4 map options possible -

A B * a _ A b _ B
A _ a B _ b * A B
A _ a B * A b _ B
B _ b A * B a _ A

* Using the formula in table 2.1, part 3, the total number of permutations for all 4 map options is -

$$2 \text{ x } \left(\frac{(1+1)!}{1!\,1!} \text{ X } \frac{(q+s)!}{q!\,s!} \right) + 2 \text{ x } \left(\frac{(q+s)!}{q!\,s!} \text{ x } \frac{(q+s)!}{q!\,s!} \right)$$

* If Stirling's Approximation to n! is used to see what happens as (q) and (s) increase, the number of maps possible turns out to be an exponential function of the input.

(Stirling's Approximation to n! $n! \sim (2\pi)^{1/2}\, n^{n+1/2}\, e^{-n}$
details of which can be found in general statistical text books such as Fraser(1976).)

Table 2.2 - Number of maps possible for a single gene.

PIL 4,055,040	PI 57,507,840	AACT 8,847,360	TOTAL $2.06 \text{ X } 10^{21}$
PERFECT PIL 4,055,040	PERFECT PI 117,411,840	PERFECT AACT 140,820,480	TOTAL $6.70 \text{ X } 10^{22}$
AT $1.12 \text{ X } 10^{12}$	ACE 13,889,160		TOTAL $1.55 \text{ X } 10^{19}$

Table 2.3 - Number of single gene maps possible and total number of multi-gene maps.

The main data set used was the PIL/PI/AACT data from Sefton et al(1990) (also shown in chapter 1, table 1.1). Given the map for PIL/PI/AACT the "perfect" data was calculated. The data was perfect in that all the fragment lengths and the number of fragments expected were exact. The third data set was the ACE/AT data set from Sefton et al(1990). The number of possible maps for each of the single genes in each of the data sets was calculated using the procedure described in table 2.1 and the results are shown in table 2.3.

3.3 Problem characteristics

Map assembly is an example of a combinatorial, sequencing problem that requires the optimisation of fragment lengths to produce a best fit map that indicates the sequence of genes and the number and sequence of restriction enzyme cut sites. The number of maps possible for a data set was found to be at minimum an exponential function of the input.

4 Previous applications of computing techniques to map assembly

A review of the literature showed there were several programs which employed different approaches to solve the simpler problem of assembling maps for cloned DNA. There was only one application which attacked the same problem as this thesis, that of assembling maps using total DNA. Although using cloned DNA greatly simplified the problem, it was considered relevant to review the applications for cloned DNA to see if there were any useful pointers for the total DNA problem.

4.1 Applications using cloned DNA

Stefik(1978), Pearson(1982) and Hoffman(1991) have developed programs to generate DNA maps from the results of experiments using REs on cloned DNA.

Stefik(1978) used an exhaustive model-driven problem-solving approach. He generated all possible solutions using the experimental data, then evaluated them. It arrived at the correct solution in a few seconds.

Using an exhaustive search strategy with the total DNA problem would not be feasible due to the time required to generate all possible maps. Stefik's evaluation mechanism relied on all the data fitting accurately in the map. Such a mechanism would not be appropriate for the total DNA problem due to the error in the number and lengths of fragments.

Pearson(1982) developed an algorithm that generated all possible permutations for the single digest fragments and aligned them until the overlap agreed with the double digest data. To evaluate potential solutions, the expected double digest data was calculated and compared with the observed double digest data. A sum of squares was calculated and the potential map with the lowest sum of squares was considered to be correct. The program was not practical when a digest had more than seven or eight fragments due to the factorial explosion in the search.

Such a method would not be feasible for the total DNA problem due to the extremely large number of possible permutations of the single digest fragments.

Hoffman(1991) developed a support tool called COMAP to assist the user with the construction of DNA maps from digest data using cloned DNA. It was not so much suited to constructing a map from scratch, but to the fine tuning of previously assembled maps or to adding in new enzymes to already optimised maps. It provided an interactive, graphical display which showed the observed data and the data expected from a proposed map. Existing maps were evaluated by considering what would be expected from a proposed map compared to what was observed. A measure of map quality was produced by calculating a penalty function for the map where every difference between observed fragments and expected fragments was penalised.

The method COMAP used for evaluating maps was sufficient for cloned DNA as there was a one to one correspondance between the fragments expected and the fragments observed. There is not the same relationship between expected and observed data when total DNA is used. Sometimes fragments which would be expected from a map are not observed and more critically, fragments that have been observed are not expected from the map. However, it was proposed that the same principle that Hoffman used be applied to generate some system for scoring maps based on differences between the observed and expected data.

4.2 Applications using total DNA

Shifman et al(1992) and Wright et al(1992) developed a tool to allow the interactive construction, merging and storage of maps from the results of experimental data using total DNA. The program acted as a support tool that provided the geneticist with facilities to store the data; to create and display possible maps on a screen; and to store assembled maps. The tool could automatically generate possible maps using four REs and twenty double digest combinations. Maps were ranked according to the goodness of fit of the expected and observed data.

The tool tackles the same problem as this thesis; however it is limited in that it could only process up to four REs and their double digest combinations. (It is not clear how many genes the tool could handle at once). The method used to build up a map relied on exhaustive search. It was proposed that the techniques developed in this thesis would use some form of heuristic technique and therefore would not be limited in the same way.

Cinkosky and Fickett(1992) and Fickett and Cinkosky (1992) (1993) as part of the Human Genome Information Resource at Los Alamos have developed a System for Integrated Genome Map Assembly(SIGMA) to enable the building, evaluating, viewing and submitting of genome maps. It was an object-oriented, X-windows based graphical map assembly tool. Data from many types of map at different levels, from the physical map down to the base sequence map, could be entered. The tool provided the ability to integrate map data and to view map information at different levels. They use a genetic algorithm (GA) to assembly "contig" maps.

"Contig" maps are different types of maps from the restriction maps being considered here. The same approach of using a form of GA was independently proposed to be an appropriate mechanism for assembling restriction maps by this thesis.

5 Summary

Geneticists are attempting to determine the sequence of the 100,000 or so genes in the human genome. The experimental techniques used to manipulate DNA to obtain data for constructing maps are described. REs are used in various combinations to cleave the DNA. The lengths of the fragments are determined by electrophoresis for cloned DNA. When only total DNA is available, the use of Southern Blotting and radioactively labelled probes are employed. Due to limitations of the experimental processes, the fragment lengths are inaccurate. The number of maps possible for a data set was shown to be very large. Map assembly was considered to be a combinatorial sequencing problem that required the optimisation of fragment lengths to produce a best fit map. Previous attempts to apply computing techniques to map assembly were reviewed. Most relied on the use of cloned DNA. An existing computer application which tackled the same problem was restricted to handling a small number of REs, possibly because an exhaustive search strategy was utilised for map assembly.

CHAPTER 3 - SEARCH STRATEGIES

1 Introduction

This chapter considers search strategies in general and introduces a form of heuristic search strategy based on the processes of evolution - a genetic algorithm (GA). GAs and how they work are described. The idea of a non-standard, or hybrid GA is introduced. An overall strategy for tackling the problem based on a GA is proposed.

2 Search strategies

The process of searching is fundamental to finding solutions to problems. Depending on the type of problem to be solved, search can vary from a straightforward task, when the problem is well defined and there is a procedure for finding the solution, to a more complicated process when the problem is not well defined, a large number of potential solutions exist and the method of arriving at the solution is not obvious. Any search of a search space involves a tradeoff between two apparently conflicting objectives - that of adequately exploring the search space and exploiting the information found so far. There are numerous techniques described in the literature for performing search and these have been categorised in many different ways. Here, search strategies are split into two main groups as shown in figure 3.1: those strategies which are guaranteed to find the optimal solution to a problem; and in contrast, those strategies not guaranteed to find the optimal solution but seek to find near optimal solutions in a reasonable length of time. The latter type of strategies are referred to as "heuristic" methods.

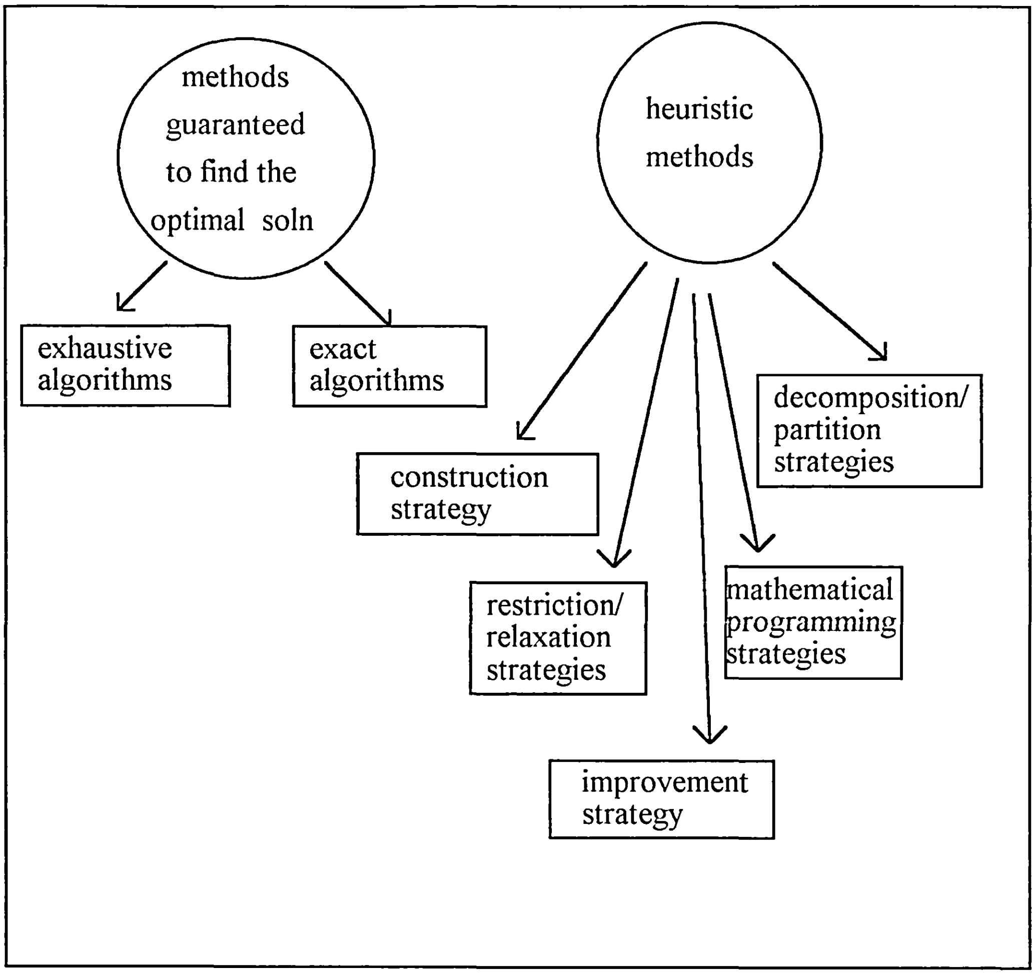

methods guaranteed to find the optimal soln
heuristic methods
exhaustive algorithms
exact algorithms
construction strategy
decomposition/ partition strategies
restriction/ relaxation strategies
mathematical programming strategies
improvement strategy

Figure 3.1 - Search Strategies

```
1. Generate an initial population

2. Evaluate the population

Repeat

        3. Reproduce and generate a new population

        4. Evaluate the new population

Until the number of trials is up
```

Figure 3.2 - The operation of a classical Genetic Algorithm

Within these two main groups, various algorithms have been developed and these are briefly reviewed in the sub-sections that follow.

2.1 Methods guaranteed to find the optimal solution

Brute-force algorithms are algorithms which guarantee to find the optimal solution by carrying out an exhaustive search of the search space. Such algorithms include any systematic form of search procedure. As these total enumeration algorithms explore the whole search space they tend to be applied mainly to small problems as the length of computing time required for problems with larger search spaces is not practical. In an attempt to overcome the effect of combinatorial explosion, various exact algorithms have been developed. The simplex algorithm was developed for linear programming problems. Other methods were based on implicit enumeration such as branch and bound methods and dynamic programming. Such algorithms find the optimal solution to problems more efficiently than complete enumeration. However, many are only efficient for small problem instances. With the increasing performance of computers, studies were conducted to measure how the computational cost of finding the problem solution varied with the size of the problem. For some problems the computational cost grew as a low-order polynomial in the problem size. However, the computational cost for other problems grew as an exponential function of the size of the problem. Cook(1971) developed the theory of "NP-completeness" which could be used to classify problems into different groups. (The Open University text(1981) provides an introduction to the theory of NP-completeness and Garey and Johnson(1979) describe many examples of NP-complete problems.) For some difficult problems, there did not appear to be any polynomial time algorithm for solving them. Karp(1972) showed that if a polynomial time algorithm could be discovered for one of these difficult problems, then polynomial algorithms would exist for the other problems in the group. It now seems likely that exact polynomial algorithms do not exist for a difficult group of problems and as a consequence, there has been a increase in interest in the development of heuristic methods.

2.2 Heuristic methods

Heuristic strategies can be categorised in several ways. Some of the main groups that emerged as a result of a survey by Zanakis(1989) are shown in figure 3.1. Many heuristic methods use various combinations of these strategies. **Constructive strategies** build up a solution to the problem one element at a time from a set of data which defines a particular problem instance. In general, a complete solution is not produced until the process is finished. The "greedy algorithm" is an example of this kind of heuristic which attempts to maximise improvement at each step. **Improvement strategies** start with a potential solution to the problem and attempt to progressively improve upon it through a sequence of changes. Local search algorithms such as hillclimbing and simulated annealing are examples of this group. **Decomposition or partitioning strategies** attempt to solve problems by breaking them down into smaller more manageable components. Components are solved either independently or by exchanging information and the result is used to build up a solution. **Mathematical programming strategies** involve using a formal mathematical model with a procedure for generating an exact solution. The procedure is altered to obtain an efficient heuristic for the problem. **Restriction and relaxation strategies** involve either reducing the problem space or expanding the problem space to produce a tractable problem.

3 Genetic algorithms

A Genetic Algorithm (GA) is a heuristic method that uses principally an improvement strategy which mimics the process of evolution. GAs were introduced by Holland(1975) as adaptive procedures based on the mechanics and analogy of natural selection. They have generated much interest in recent years and some of their recent practical applications are reviewed in Goldberg(1994). Holland recognised that natural systems were more robust than artificial systems and attempted to achieve robustness by developing an algorithm which emulated the processes of natural selection. In natural selection, the individuals which are best adapted to their environment tend to have the greatest chance of survival and reproduce more often, passing on their genes to the next population. GAs have been developed as search procedures that are population based and proceed over a number of generations. The criteria of "survival of the fittest" provides evolutionary pressure for populations to develop increasingly fit individuals. A brief review of the classical GA will be given, the reader is referred to Goldberg(1989) for a fuller introduction.

3.1 Description

In the "traditional" genetic algorithm (as defined by Davis (1991a)), a "chromosome" or potential solution to the problem is represented as a binary coded fixed length string. A number of potential solutions are generated at random to produce what is called an initial population. The "fitness" of the population is evaluated by assessing the fitness of each individual (or potential solution) in the population. A new population is produced by performing operations patterned after genetic operations such as sexual recombination (crossover) and fitness proportionate reproduction (Darwinian survival of the fittest). The more fit individuals (the better solutions) reproduce (combine together) in an attempt to generate more highly adapted individuals (solutions that are better still.) This process is repeated and each successive population is called a generation. After a fixed number of generations (trials), the fittest individual represents the solution. The steps in a classical genetic algorithm are shown in figure 3.2.

The GA is a "parallel" algorithm in that it transforms a population of individual objects into a new population. During reproduction, parents are selected to mate, the recombination operators are applied and the children are inserted into the new population. Selection is the survival of the fittest in a GA. Selecting parents to reproduce in proportion to fitness ensures that above average parents are selected to reproduce more frequently. In the traditional GA, a whole new population of individuals are created, saving the best one from the previous generation (known as Generational Replacement with Elitism) which ensures that when the best solution is found, it is not lost through disruption from crossover or mutation.

3.2 Implementation

In order to apply a GA to a particular problem, there are various issues that must be addressed such as representation; generating the initial population; selecting an objective function; choosing operators; and setting the parameters to appropriate values.

The problem must be "represented" in GA notation - traditionally a fixed length binary string representation is used. Some technique must be chosen to generate the initial population for the GA. This is a constructive process and at the simplest level potential solutions can be generated at random. During reproduction, recombination operators analogous to the biological operators of crossover and mutation are applied. Traditionally, the crossover operator involves exchanging strings at random, combining parts of good individuals in an attempt to create a more fit individual. This is illustrated in figure 3.3. The role of crossover is to provide an opportunity for the best attributes of both parent strings to be incorporated into the offspring. Mutation is a mechanism for introducing variation into the population - it tends to be applied less frequently and is considered to play a secondary role. Mutation involves changing a single value in an individual in order to generate some unexpected variation in the population and is illustrated in figure 3.4. There are various parameters that must be set for the GA such as population size, frequency of application of the genetic operators and number of generations to run. For traditional GAs, these values have been determined.

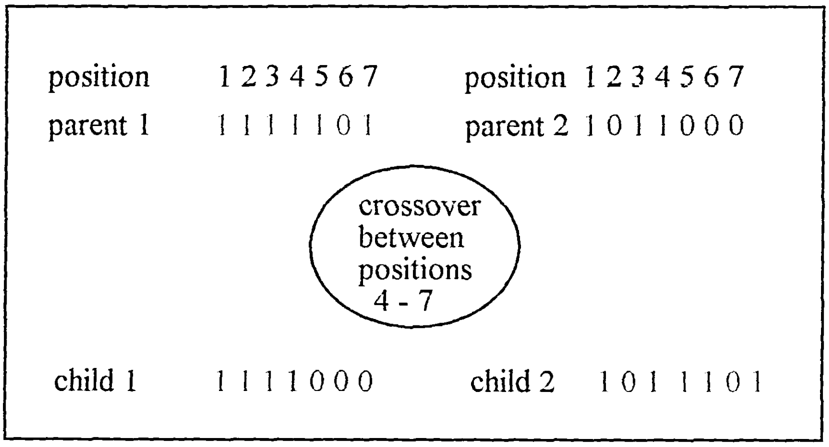
The objective function in a GA plays the role of the environment by rating potential solutions in terms of their fitness. In order to apply a GA to a problem, there must be some method of measuring the goodness of potential solutions. The evaluation function is the one area where the traditional GA requires problem specific knowledge. For some types of problems, the choice of evaluation function is obvious but for other problems it is not as straightforward. DeJong and Spears(1989) discuss some of the problems associated with developing evaluation functions for difficult problems.

3.3 Theory

There is a well established theory which has been developed by Holland to explain why GAs work and it is based on a binary representation and the notion of a schema. A brief informal overview is given here - a complete mathematical description is contained in Goldberg(1989).

A schema can be thought of as a similarity template which is used to describe subsets of strings which share similarities at different string positions. A schema is generated by introducing a "don't care" symbol "*" into the binary alphabet (0,1,*).

A schema (plural - schemata) represents all strings that match it on all positions other than "*". For example, schema *000 describes a subset of two strings namely 1000 and 0000. The number of "don't care" symbols in a schema determines the number of strings matched by the schema.



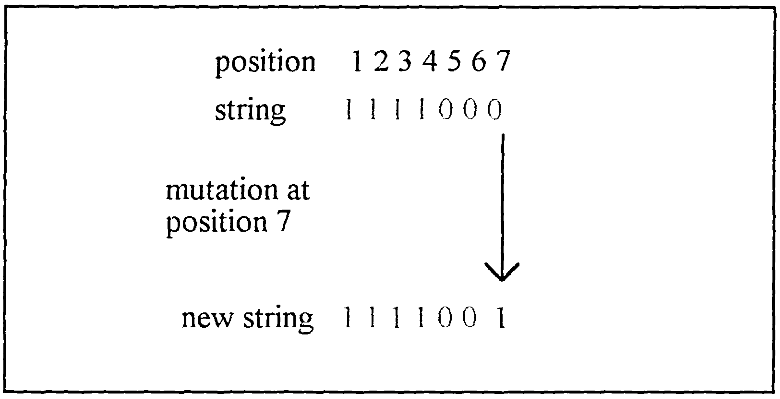
position 1 2 3 4 5 6 7 position 1 2 3 4 5 6 7

parent 1 1 1 1 1 1 0 1 parent 2 1 0 1 1 0 0 0

child 1 1 1 1 1 0 0 0 child 2 1 0 1 1 1 0 1

Figure 3.3 - The crossover operator.

(The individuals are represented in binary notation.)



position 1 2 3 4 5 6 7

string 1 1 1 1 0 0 0

new string 1 1 1 1 0 0 1

Figure 3.4 - The mutation operator.

A schema has two properties - **order** and **length.** The order of a schema is the number of fixed positions present. This is the length of the template minus the number of "don't care" symbols. The length of a schema is the distance between the first and the last fixed string positions. For example,

Schema	order	length
1*1**110	5	7
***10*11	4	4

Using the notion of a schema provides a compact way of talking about the similarities among strings. It is possible to calculate the number of schemata in a population and to look at the effect that reproduction, crossover and mutation have on them. As one string contains many schemata, it can be shown that there are a large number of schemata present in a population. The result of this is that a large number of schemata are tested in each generation and this property of GAs is referred to as "intrinsic parallelism". Providing a reproductive plan that makes reproductive chance proportional to fitness is used, reproduction allows above average schemata to grow by giving them an increased number of trials. Below average schemata die off as they are allocated with a decreasing number of trials. Crossover tends to disrupt long schemata whereas shorter length schemata have a greater chance of remaining intact. As mutation is normally applied at very low rates, it does not have much effect on schemata. The **Schema Theorem** was put forward by Holland to describe the dynamics of a GA. The Schema Theorem says that short, low-order, above average schemata will increase their representation in subsequent generations of a GA. This occurs because simple crossover does not disturb short schemata with high frequency, reproduction allocates more copies to the best schemata, and because mutation has little effect.

Goldberg(1989) refers to the short, low-order above average schemata as "building blocks" and describes the dynamics of a GA in terms of the **Building Block Hypothesis** (BBH). The BBH says that a GA seeks near optimal performance through the juxtaposition of building blocks. A GA search attempts to reduce the complexity of a problem by constructing better and better strings from the best partial solutions of past samplings. That better and better strings are created has not been proved, however, there is a large body of empirical evidence to suggest that the BBH holds for many problem classes. The consequences of the BBH are discussed in Goldberg et al(1993).

3.4 Comparison with conventional search methods

In contrast to search methods which guarantee to find the optimal solution to a problem, a GA is a heuristic method that will seek to find a near optimal solution in a reasonable length of time. A GA uses some form of construction strategy to generate an initial population of potential solutions. At the simplest level, this can involve initialising the population at random. The improvement strategy that is used takes the form of a multi-directional search. GA search differs from local search in that these improvement strategies search from point to point. One of the main problems with local search techniques is that they find the optimum point in the current neighborhood and this is dependent on the starting point. Local search exploits the best among known possibilities - exploration is restricted. GAs attempt to overcome these limitations by maintaining a database of points and by performing search in parallel. GAs are inherently performing an even wider search by virtue of their property of intrinsic parallelism. The theoretical analysis of GAs suggest they manage the tradeoff between exploration and exploitation in a near-optimal way (Holland(1973), (1975)).

GAs were developed as general-purpose search techniques. In contrast to those search methods that rely on auxillary information or assumptions regarding the search space, the only problem specific knowledge required by the GA is contained in the objective function. It is in this sense that GAs are described as being robust and they have been found to perform well across a wide variety of problem domains.

The GA method is based on a probabilistic process rather than a deterministic process. Although the GA uses probability, it is unlike a random search as the search is directed into promising regions of the search space. Random search concentrates wholly on exploring the search space with no exploitation of promising points found.

In the past it was argued that the use of recombination operators, in particular the crossover operator, distinguished a GA from other optimisation techniques. However, nowadays there is considerable overlap amongst several evolutionary algorithms such as evolutionary strategies (Rechenberg (1965)) and evolutionary programming (Fogel et al(1966)) and the boundaries between them have become fuzzy (Back and Schwefel(1993)).

4 Non-standard genetic algorithms

The standard or traditional GA is one which uses binary fixed length string representation, one-point crossover and mutation, generational replacement normally with elitism and some fitness normalisation process. The traditional GA has the theoretical underpinnings of the Schema Theorem. A central goal in GA research has been to develop an algorithm that is robust and can perform well across a variety of problem domains with no problem specific knowledge. There are many researchers who are still pursuing this goal and working with the traditional GA. There has also been considerable interest in the application of GAs to real-world problems. From an AI perspective, the standard GA can be classed as a "weak method" as it makes few assumptions about the problem domain and is widely applicable. As a weak method, the traditional GA is unlikely to be the best algorithm to use for any particular problem. Various modifications to the traditional GA have been proposed and the concept of a "hybrid GA" (HGA) has been suggested by a number of authors ((Bosworth et al(1972), Bethke(1981), Goldberg(1983)) and Davis(1991a). Some research has been conducted on HGAs and has shown that such an approach is successful for combinatorial optimisation problems. Goldberg and Lingle(1985), Oliver et al(1987), Whitley et al(1989)(1991) and Grefenstette et al(1985) have applied GAs to the Travelling Salesman Problem. Syswerda(1991) applied an HGA to the NP-hard problem of schedule optimisation. A review of some of the differences between the traditional GA and HGA is provided in the next section.

4.1 Representation

It was recognised that the binary string representation was not the most suitable for certain types of problems. In some cases, the most natural representation involved more complex data structures and by linearising the data structure into a string representation the window by which the system observed the world was limited. Davis(1991b) and Michalewicz(1993) reported that for numerical optimisation problems that required a high degree of precision, real number representation gave solutions with greater accuracy and in fewer generations than equivalent implementations using binary. Other representations have used character strings, gray coding, integers and matrices. Goldberg et al(1989),(1990) investigated using variable length strings developing the "messy GA".

4.2 Operators

Applying the traditional GA to sequencing problems presented difficulties as illegal sequences could be generated. For example, in the case of the Travelling Salesman Problem, there are constraints placed on the symbol string that represents a tour of the cities, in that no city can appear more than once. The classical recombination operators which worked well when solutions were coded as bit strings did not work when the solutions were coded as sequences. As crossover and mutation rearranged symbols independently of each other, this could lead to cities appearing more than once or not at all. Various approaches have been taken to enable the application of the traditional GAs to constraint problems. Two main methods have been used - the first involves using a penalty function in the objective function. Potential solutions which violate constraints are generated, but are heavily penalised. There are disadvantages to this approach and it is generally considered feasible only if the number of constraints is small. The second method involves the use of specialised decoders and repair algorithms but these are often computationally costly to run. A third approach is to modify the traditional GA in terms of the problem representation and to devise new operators.

A variety of operators have been developed for sequencing problems (Goldberg and Lingle(1985); Davis(1985); Oliver et al(1987); Davis(1991a); Whitley et al(1989); Whitley et al(1991); Fox and McMahon(1991)). Evidence suggests that the effectiveness of different types of operator depends on the nature of the problem (Starkweather et al(1991)). All ordering problems are not similiar - in some, the relative order of tasks is important and in others, adjacency is important. Operators differ in the kind of information they attempt to preserve during recombination. In general, it is recommended that potential solutions should be broken up in a way that is natural for the problem. If there are heuristics that can be applied, it is often useful to incorporate these into the operator set to provide domain based guidance to the search process (for example: Bosworth(1972); Grefenstette(1987)).

4.3 Theory for non-standard genetic algorithms

Holland's Schema Theorem is well developed for problems that can be encoded as a binary string and use the standard operators. However, for many problems binary representation is not a natural coding and for ordering problems various different operators have been developed. Several authors have attempted to re-examine and generalise schema analysis and the notion of implicit parallelism to provide a theoretical foundation for non-standard GAs (Goldberg and Lingle(1985), Oliver et al(1987), Goldberg(1989), Antonisse(1989), Whitley et al(1989), Vose and Liepins(1991), Radcliffe(1991a, 1991b, 1993, 1994), Radcliffe and George(1993), Eshelman and Schaffer (1992,1993)).

In terms of representation, Antonisse(1989) provided an alternative interpretation of the schema notation which overturned the binary coding constraint. This allowed the use of alternative codings for a problem using a higher cardinality alphabet. Recent results from Radcliffe and Surry(1994a) provide further evidence that casts doubts on the appropriateness of low cardinality representations. For real-coded GAs, Eshelman and Schaffer(1993) introduced "interval schemata" as a tool for analyzing their performance.

Goldberg and Lingle(1985) extended the concept of schemata for ordering problems by defining o-schemata (ordering schemata). They used the notion of o-schemata to analyse the effect of a new operator PMX (Partially Mapped Crossover) on the performance of the GA for ordering problems and demonstrated that PMX preserved ordering building blocks (low order, o-schemata) with a high probability. This gave the desirable result of an operator which searched among both ordering and allele combinations (o-schemata and a-schemata) that lead to good fitness. Oliver et al(1987) proposed a modified definition of o-schemata. They found that the performance of operators for ordering problems determined using o-schema analysis depended on the type of ordering problem. Goldberg(1989) described how a family of definitions of an o-schema could be developed to meet the requirements of different types of ordering problems (for example, when absolute or relative position mattered).

Whitley et al(1989) investigated operators for ordering problems in the context of the Traveling Salesman Problem(TSP) and devised the "edge recombination operator". They proposed that the important information in the TSP was not the position of a particular city but the critical links between cities. Their edge recombination operator searched for critical edge recombinations. They showed that as an underlying binary representation existed for edge information, the operator could be related to Holland's Schema Theorem and there was no need for any new notion of a schema.

Vose and Liepins(1991) analysed schema disruption and formalised a "building block hypothesis" based on the interaction of the crossover operator with schemata. They argued that the building blocks should be determined from considering the interaction of the crossover operator with schemata. They noted that with the increasing application of GAs to combinatorial problems, non-standard crossover operators were being developed and that sometimes the "building blocks" of the problem were not clear.

Radcliffe(1991a,1991b,1993) has extended the notion of intrinsic parallelism and the associated Schema Theorem to general non-string representations through the introduction of arbitrary equivalence relations. Radcliffe calls his general version of a schema a "forma" (plural "formae"). A forma is used to describe subsets of chromosomes that are similar in some general way. His focus is on finding sets of formae that characterise the regularities in the particular problem or class of problems under consideration and developing operators that manipulate these to good effect. To this end, Radcliffe has proposed several "design principles" for devising useful equivalence relations, representations and recombination operators. Applying his design principles, Radcliffe has developed a family of representation independent operators. Using Radcliffe's approach, new recombination operators can be designed that possess desirable properties such as "respect" (if both parents have the same characteristic, the offspring should inherit that characteristic), "proper assortment" (if both parents have different characteristics, it should at least be possible that both characteristics can be inherited by the offspring) and "strict transmission" (if one parent has one of two characteristics, the offspring must inherit one of the two characteristics). As these properties may not always be compatible various trade-offs must be made between them as described in Radcliffe(1993) and Radcliffe and George(1993).

5 Problem solving strategy

The map building problem is difficult as it relies on error-prone data; there can be more than one solution feasible; there are a large number of possible maps; and the method used to arrive at a solution is based on subjective judgement. The approach the expert takes to handle the problem complexity was examined. The geneticist often concentrates on assembling potential single gene maps and then postulates how well the single gene maps could be fitted together and maps are built up incrementally. The human approach is a form of problem decomposition using a bottom-up strategy to make the process manageable and subsequently a merge operation is carried out which involves aligning the single gene maps with one another over fragments common to pairs of maps. It was proposed that a similiar strategy be adopted to automate map assembly. It was proposed that a form of GA be developed to determine the sequence of cut sites for single gene maps from the results of experimental data. As standard GAs have not been effective for sequencing types of problems (illegal sequences could be generated), a modified GA was proposed that would incorporate the objective system for assessing maps. Once several "acceptable" maps have been developed for individual genes a merge process would be carried out to merge maps on common fragments. The merged map would determine the sequence of all the genes and cut sites.

An objective system for assessing maps would be developed - techniques would be used to collect and process the geneticist's expert knowledge.

6 Summary

Search strategies were discussed. The traditional GA was introduced and ways of modifying the GAs to handle complex problems were considered. The strategy for automating map assembly was presented that consisted of developing a method of objectively assessing maps; developing a modifed form of GA for generating potential single gene maps from experimental data; and developing a system for aligning single gene maps together to generate a complete multi-gene map.

CHAPTER 4 - DEVELOPING AN OBJECTIVE SYSTEM FOR ASSESSING MAPS

1 Introduction

Evaluating maps relies on the subjective judgement of the geneticist. Obtaining the data to assemble maps from is time-consuming and problematic and the data itself tends to be inexact. Once a map has been assembled that appears to represent a good fit of the fragments, there is no mechanism for objectively assessing the map.

To develop an objective system for assessing maps, four main tasks were identified. Firstly, a knowledge elicitation process was conducted to determine what were the "good" and "bad" features of maps. Secondly, alternative maps for the same genes were analysed to highlight discrepancies between maps. Thirdly, using the map information, a questionnaire was devised to enable maps to be scored. Lastly, a general mechanism for evaluating single gene maps was developed.

2 Knowledge Elicitation Stage

A series of informal interviews and discussions were held with the geneticists in an attempt to identify what they regarded as general features or characteristics of "good" and "bad" maps and to find out what rules of thumb or heuristics they used when carrying out map assembly. The heuristics and techniques used by the geneticist are summarised in table 4.1.

1 Take into account the strength of fragments.

The strength of fragments on the gel lane can vary considerably. Have the most confidence in the strong fragments that show up distinctly.

2 Allow for variation in length of fragments.

Some allowance must be made for fragment lengths as they can vary due to experimental limitations Allow +/- 10% on fragments up to 1000kb and +/- 20% on fragments between 1000 to 2000kb.

3 Identify common fragments in a dataset.

For a gene. All fragments containing a particular gene must overlap one another in region of the smallest length fragment containing the gene.

Between single and double digest fragments of the same gene. If a double digest fragment is the same length as a single digest fragment, it is likely they represent the same fragment.

Between genes. If the same length of fragment is obtained for two or more genes using the same restriction enzyme, it is possible that it is a common fragment and contains both genes.

4 Identify partial digests

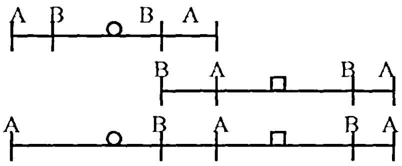
Look at the number of fragments obtained using a particular restriction enzyme. If more than one fragment is obtained for a gene, there must be partial digests. (Refer to figure 2.4 for examples.)

5 Determine relationships between fragments

The relationship between fragments can be determined using the double digest results.

6 Identify cut sites common between genes.

If two genes are close on the DNA, it is likely that the cut sites from the single gene maps will overlap. For example



7 Consider the number of restriction enzymes and length of the map.

If several potential maps have been assembled from the same data, compare the number of cut sites and length between the maps. The map with the least cut sites and shortest length is likely to be a good map as this indicates that the data has fitted well into the map.

8 Assemble single gene maps.

Take each gene in turn and develop single gene maps. Merge the maps together on a common cut site.

9 Assemble a map by considering pairs of genes.

Select a pair of genes and examine the single and double digest information for the pair and generate a tentative solution. Repeat the process for other pairs of genes then merge the gene pair maps together.

10 Assemble a map from the smallest fragment.

Choose the smallest strong fragment containing a gene as the starting point. All other fragments must overlap it.

Table 4.1 - The geneticist's heuristics used for map assembly.

3 Analysis of published maps stage

The second exercise involved analysing published maps and the data from which they were generated to gain insight into the error tolerated in maps. Four different maps have been published for the PIL,PI,AACT genes. Kearney et al(1987) reported that the genes were ordered PIL-PI-AACT, whereas Cox et al(1987) reported the genes were ordered PI-PIL-AACT. Both groups revised their original maps: the Kearney group published a revised map in Sefton et al(1990); and the Cox group published a revised map in Billingsley et al(1993). Although the order of the genes in the revised maps agreed (PIL-PI-AACT), there were many differences in the order and number of the RE cut sites. The Sefton map and Billingsley map were analysed in detail as was a third map assembled by the author using the Sefton et al(1990) data (the Proposed Map) that appeared to represent a good fit of the data. The three maps are shown in figure 4.1. Taking each map in turn, the fragments that would be expected were calculated and discrepancies between the observed data (the experimental data) and the data expected from the map were noted. The number of discrepancies found for each map is summarised in table 4.2. (The detailed analysis for each map is contained in Appendix B.) The nature of the discrepancies is summarised below.

* Fragments observed but not expected from the map.

* Fragments expected from the map but not observed.

* Different lengths of fragments expected from the map compared to the observed fragment lengths.

* Differences in the nature and number of cut sites in the map to allow the observed fragments to be fitted.

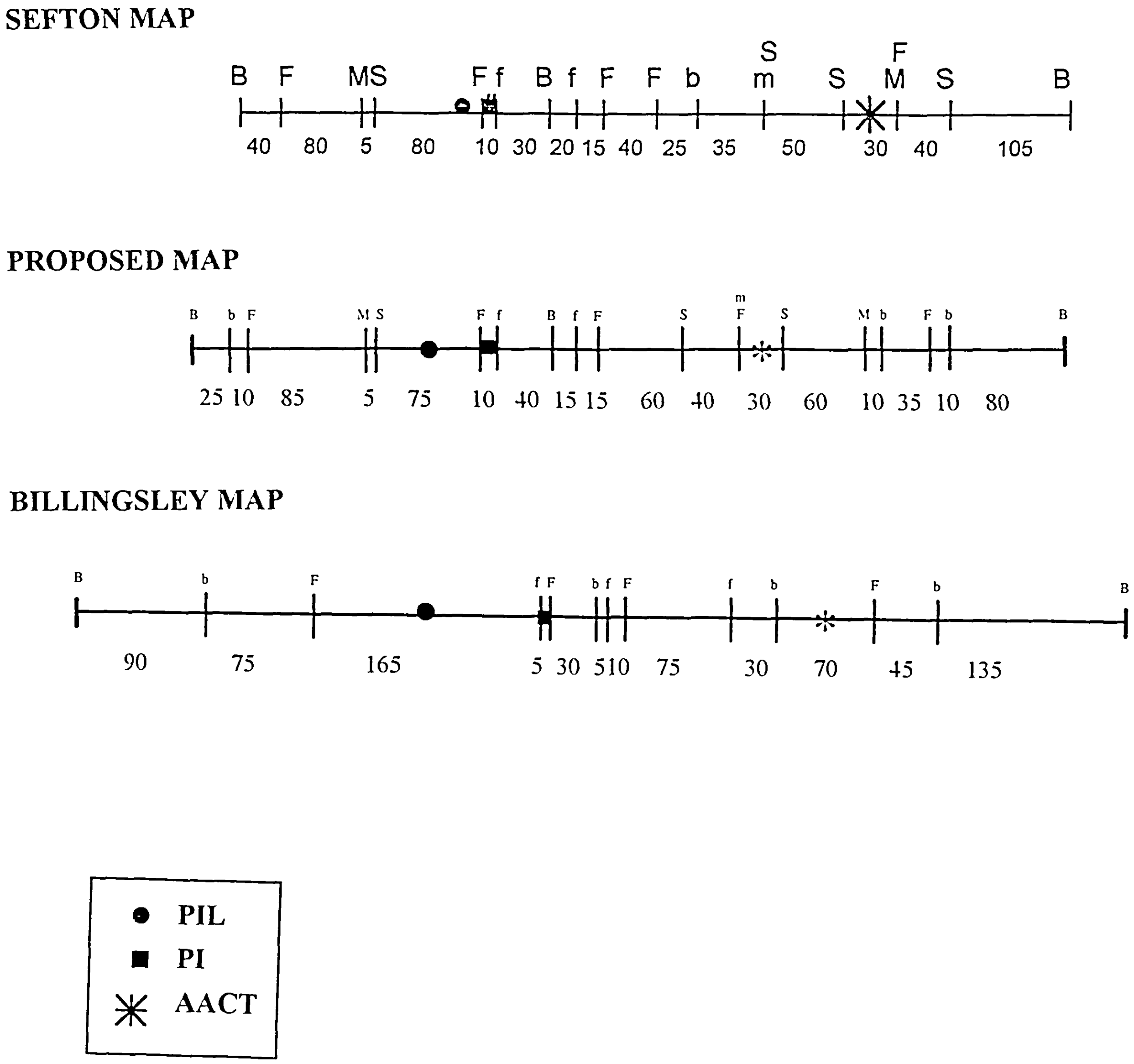


Figure 4.1 - Alternative maps for PIL,PI,AACT.

The published map from Sefton et al(1990) (length=605kb); the author's Proposed map (length=605kb); and the published map from Billingsley et al(1993) (length=736kb). (Not to scale.)

	PIL	PI	AACT	TOTAL
Sefton map	6	5	9	20
Billingsley map	8	8	7	23
Proposed map	5	4	6	15

Table 4.2 - Summary of the number of discrepancies found between the observed and expected data in the Sefton, Billingsley and Proposed maps for the PIL,PI,AACT genes.

	PIL	PI	AACT
Sefton map	7 minor 1 significant 1 critical **28 - unacceptable**	6 minor 1 critical **24 - unacceptable**	2 minor 1 minor - significant 2 serious **21 - unacceptable**
Billingsley map	5 minor - serious **5 - adequate**	1 minor - serious **1 - good**	2 minor 1 minor - serious **3 - good**
Proposed map	3 minor **3 - good**	2 minor 2 significant **8 - acceptable**	2 minor 2 significant - serious **8 - acceptable**

Table 4.3 - Summary of gene map scores.(The total scores for each gene map are expressed in units of minor problems and can be calculated as follows - score 1 for a minor problem; 3 for a significant problem; 9 for a serious problem; 18 for a critical problem.)

Questionnaire	Evaluation		
Section	1	2	3
1. Fragment lengths	Y	N	Y
2. Nature of cut sites	N	Y	N
3. Weak fragments	N	N	N
4. Length of single digests	Y	N	Y
5. Yield of double digests	N	Y	N
6. Length of double digests	N	N	Y

Table 4.4 - Evaluation coverage of the questionnaire.

4 Quantifying map features

The geneticist's heuristics provided information regarding map characteristics as did the analysis of the PIL/PI/AACT maps. Some mechanism was required to utilise the information to generate an objective system for assessing maps. It was proposed that a questionnaire be devised to measure the relative importance of the various map features.

4.1 Developing a questionnaire

The approach to questionnaire design as described in Oppenheim(1966) was followed. Several general areas used to assess maps formed the basis of the questionnaire and the geneticist was asked to rate various problem situations described in each area. Every effort was made to ensure that questions were worded to prevent bias and to minimise ambiguity. In order to measure the geneticist's opinion of each map problem and of its overall importance to a map, a rating system was devised. The geneticist was given five ratings which could be applied to each problem - "no problem", "minor problem", "significant problem", "serious problem" or "critical problem". (It was considered that any fewer in the rating scale would not adequately represent the range of problem types and any more would generate a difficulty in discriminating between them.) Once all the problems were rated, the geneticist was asked to specify how many problems would be tolerated in an "ideal", "good", "acceptable" and "unacceptable" map.

The six areas the questionnaire examined are briefly summarised here. Section 1 considered fragment lengths. Errors were present in the observed fragments due to experimental limitations - the amount of error tolerated in fragments of different lengths was established. Section 2 examined the importance of the nature (partial or complete) and number of the cut sites relative to the number of fragments observed. The role of weak fragments in map assembly was considered in Section 3. Section 4 assessed how single digest fragments fitted into a map in relation to other single digest fragments. The effect of differences in the number and type of double digests observed and expected was measured in Section 5. Section 6 determined the number of minor, significant, serious and critical problems that could be tolerated in an "ideal", "good", "adequate" or "unacceptable" maps.

The questionnaire was completed by the external advisors, Povey and Bickmore, and is contained in appendix C.

4.2 Analysis of questionnaire results

The use of a rating system in questionnaires does invite possible errors (Oppenheim (1966)). One of the main problems lies in the ease with which ratings can be influenced by variables of which the rater is unaware. Ratings may differ due to the raters having different "frames of reference". This is a general problem that can lead to much misunderstanding. To minimise the problem, the frame of reference the rater was expected to use was described as far as possible in each question. On reviewing the questionnaire results, it became clear that the external advisors did have different frames of reference which appeared to be due to the different types of work each was involved with. For the region of chromosome that the Bickmore group was interested in, there was information available in the literature which influenced the design of their experiments and provided them with benchmarks along the chromosome. As a result, it was not necessary to carry out all combinations of double digests. For the particular region of chromosome that the Povey group was interested in, there was little additional information available in the literature to draw upon. The experiments carried out by the Povey group were self-contained in that all the information required to generate the map was determined as a result of carrying out the single digests and all combinations of double digests. It was proposed that the system for automating map assembly should be self-contained in the first instance and therefore the scores applied by the Povey group were used as the basis for the scoring system.

4.3 Scoring maps based on questionnaire results

The Povey group's answers to the questionnaire were applied to the problems in the three PIL/PI/AACT maps shown in figure 4.1 which were described in full in Appendix B, to see which map scored best. The scoring of the discrepencies is shown in detail in Appendix D and the map scores are summarised in table 4.3. A score of 0 indicated an ideal map; 1-3 indicated a good map; 4-9 indicated an acceptable map; and 10+ indicated an unacceptable map. Overall, the Billingsley map appeared to score best, however, if it had been possible to include the MluI restriction enzyme information for the Billingsley map, the score may have been different. Both the Billingsley map and the Proposed map were classed as acceptable maps. The Sefton map contained critical problems (as illustrated in Appendix E) and would not be considered acceptable.

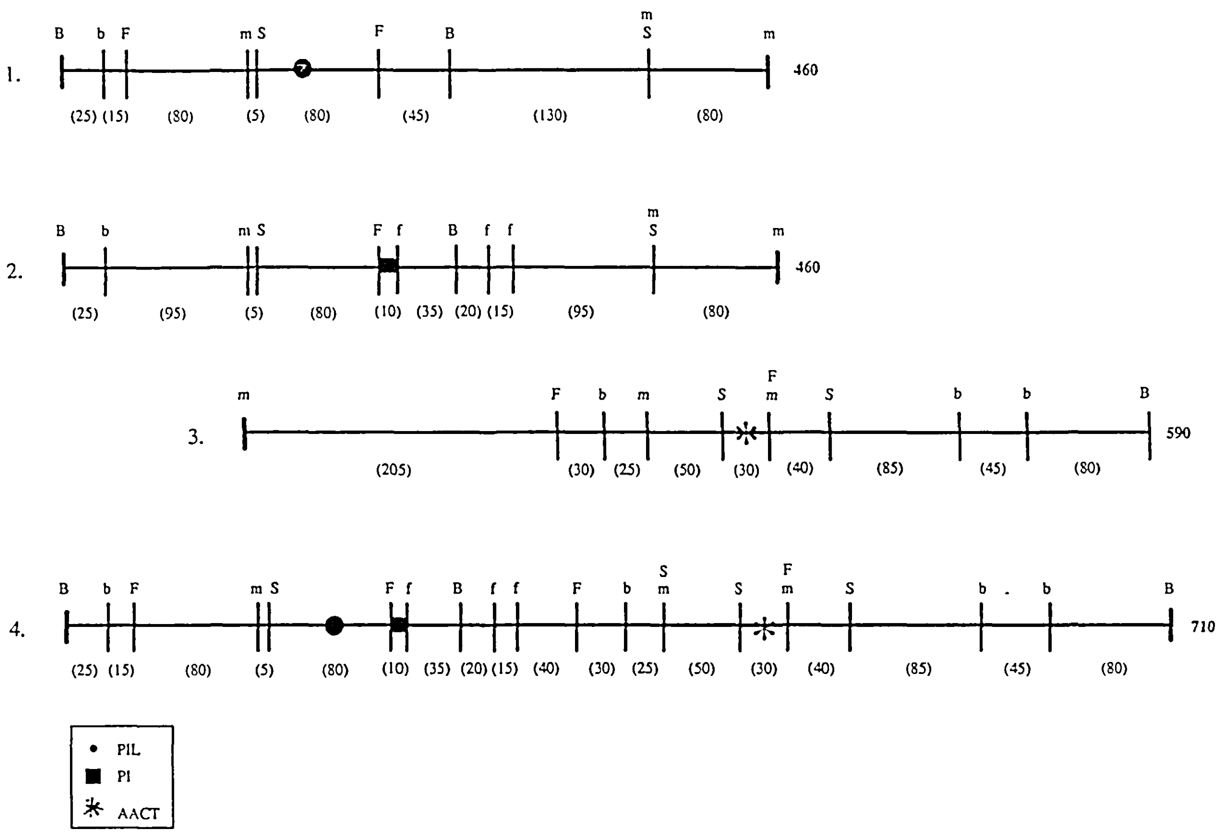


Figure 4.2 - The Sefton revised published map.

The complete revised multi-gene map and single gene maps for PIL,PI,AACT are shown.

As a result, the Sefton group revised aspects of their published map. Three extra restriction enzyme cut sites were added and three restriction enzyme cut sites were changed from being complete cut sites to partial cut sites. The order of the genes was unaltered. The revised published map is shown in figure 4.2. The changes made to the original published map and the scoring of the revised map are shown in detail in Appendix E. The revised map was scored and was now classed an acceptable map containing no critical problems.

5 Developing a general mechanism for scoring maps

As the strategy for tackling map assembly involved assembling single gene maps, a general mechanism for objectively assessing such maps was required. (The method of evaluating multi-gene maps is described in chapter 8.)

It was proposed that the evaluation mechanism would take as input the observed experimental data for one gene and a potential map assembled from that data, and would generate a score for the gene map. Different ways of evaluating maps were considered based on the responses to the questionnaire and three main evaluation methods emerged. The first evaluation assessed a gene map on how well the observed single digest results fitted. The second evaluation assessed a gene map on how well the double digest results fitted. The third evaluation assessed a gene map on how well all the observed data fitted into the gene map. The broad areas of the questionnaire that each evaluation covered are shown in table 4.4. Each evaluation was not considered appropriate as the sole means of assessing maps. For this reason, a gene map was awarded a score from each evaluation which were summed together to produce a total score. The total score was expressed in units of minor problems. Depending on the score of a map (or in other words, the number of minor problem equivalents present in the map), the map was classifed as "ideal", "good", "adequate" or "unacceptable". An "ideal" map was a map which scored zero and therefore had no minor problems. The number of minor problems tolerated in "good", "adequate" and "unacceptable" maps depended on the size of the map. For large gene maps, a "good" map had up to three minor problems, an "adequate" map had between four and nine minor problems. More than nine minor problems and the map was considered unacceptable.

Each of the three evaluations are described in the following sections. Examples of applying the evaluations are given in the Appendix F.

5.1 Evaluation 1 - Fit of single digest data

The first method of evaluating a gene map was based on how well the observed single digest fragments fitted into the map. The single digest fragments were the fragments obtained using a single restriction enzyme. Each single digest fragment was the result of a restriction enzyme cutting the DNA on either side of the gene.

Evaluation 1 took as input a gene map and the observed single digest data for each restriction enzyme and "assigned" the lengths of each of the observed single digest fragments to the single digest fragments that would be expected from the map. As all single digests obtained for a particular gene must overlap in the region of the gene, it was likely that some single digests would be nested within other single digests in the map. The pairs of restriction enzyme cut sites producing each single digest were checked to see if any pair of cut sites were nested within another pair of cut sites. Where this was the case, the observed fragment lengths that had been assigned to each digest were examined. If a nested single digest was found to be bigger than the outer fragment, this indicated a problem with the map. A map score was awarded which reflected the difference in fragment sizes.

The map score was based on the amount by which the nested fragment was larger in relation to the outer fragment and was expressed in units of minor problems. The difference between the nested fragment and the outer fragment was calculated, taking into account the possibility that each of the fragment lengths might have been out by up to 10% (for fragments <1000 kb). In addition to the 10% allowable error, if the nested fragment was greater than the outer fragment by

0 or less	- no problem	score = 0
up to 10%	- minor problem	score = 1
11 - 24%	- significant problem	score = 3
25 - 49%	- serious problem	score = 9
50% +	- critical problem	score = 18

The score indicated how well the observed single digest data fitted into the map.

In some maps, it was difficult to assign the lengths of the observed fragments to the fragments expected from a map because of the number of cut sites producing the fragments, and to differences between the number of fragments expected from a map and the number of fragments observed. For example, if the number of cut sites in a map for a particular restriction enzyme, say "B", was even and of the form "B b * b B" (where "*" indicated the gene, "B" was the outer complete cut site and "b" indicated a partial cut site), four fragments would be expected. If all four fragment were observed, how would they be fitted to the cut sites ? The largest observed fragment could be fitted to the "B * B" cut sites and the smallest observed fragment could be fitted to the "b * b" cut sites. However, it would be difficult to say which of the two remaining observed fragments could be fitted to "B * b" and "b * B" without knowing the distance between the cut sites. So, even when the same number of fragments were expected as observed, there would be uncertainty surrounding the assignment of fragment lengths.

The experimental results for single digest fragments were more accurate than for double digest fragments because of the optimal operating conditions that could be created for a single restriction enzyme. As inconsistencies in the number of fragments observed and expected were not considered frequent, it was proposed the evaluation would be applied only to maps that had the same number of single digests observed as expected. (This restriction was not applied to double digests as described in section 5.3.)

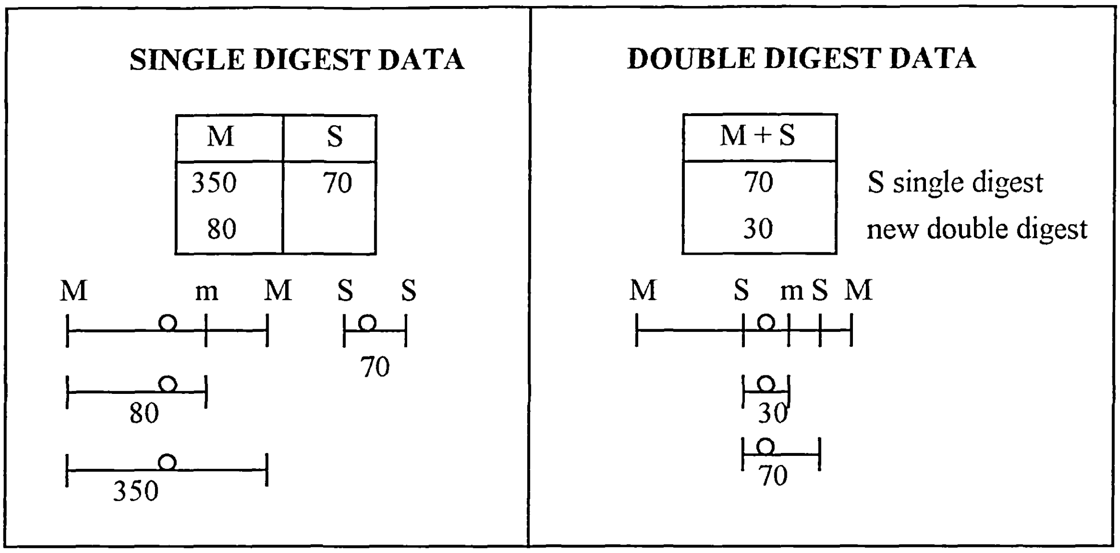
Given the same number of single digests expected as observed, when there was a matching problem between the observed and expected fragments, the length of the largest observed fragment was assigned to the cut sites that resulted in the widest position in the map. As a preliminary measure, the lengths of the remaining fragments were assigned on a "first-come first-serve" basis. However, it was proposed that this method be reviewed at a later date as it could lead to a good match being excluded.

5.2 Evaluation 2 - Fit of double digest data

The second method of evaluating the quality of a gene map was based on determining the number and type of double digests expected from a map compared to the number and type of double digests observed from the experimental data.

The double digest fragments were the fragments obtained when two restriction enzymes were used to chop up the DNA. Each double digest fragment was the result of one or other of the restriction enzymes cutting the DNA on either side of the gene. A double digest fragment could be classified usually as either a single digest fragment or as a new double digest fragment. The double digest fragment would be considered a single digest if the length the double digest was the same as the length of a single digest (indicating that the other restriction enzyme had not cut within it). A double digest whose length was different from both single digests would be considered a new double digest. This is illustrated using by example in figure 4.3.

Evaluation 2 took as input a gene map and the observed double digest fragments. For each of the observed double digest fragments, the following procedure was carried out. (Weak fragments were ignored.)



SINGLE DIGEST DATA

M	S
350	70
80	

DOUBLE DIGEST DATA

M + S	
70	S single digest
30	new double digest

Figure 4.3 - Types of double digest.

The length of the single digest fragments for two restriction enzymes, **M** and **S** are shown. The lengths of the double digest fragments, when both restriction enzymes were used together is shown. The **MS** double digest of length 70 is likely to be the S single digest fragment as they are the same length. The small **MS** double digest of length 30 is likely to be a fragment cut at one end by **m** and the other by **S** as there were no single digests of that length. A map showing the number and sequence of the **M** and **S** cut sites that would generate the fragment lengths is shown (not to scale).

1. From the table of observed data, each double digest fragment was classed as either a new fragment or as a single digest fragment. The likely ordering of the restriction enzyme cut sites around the gene was determined.

2. The ordering of the double digests fragments expected from the map was calculated and this was compared with the ordering of restriction enzyme cut sites of the observed double digests that was calculated in step 1.

3. Based on the comparison described above, a score was awarded for the map as follows. The percentage of the type of double digest fragments observed which were expected was scored.

		SCORE
95%+	- no problem	= 0
91 - 94%	- minor problem	= 1
90%	- significant problem	= 3
<90%	- critical problem	= 18

The percentage of the type of double digest fragments expected which were observed was scored.

		SCORE
70%+	- no problem	= 0
61 - 69%	- minor problem	= 1
60%	- significant problem	= 3
51-59%	- serious problem	= 9
<50%	- critical problem	= 18

(Note that the percentage "bands" used were based directly on the geneticist's responses to the questionnaire. Rather than introduce possible errors by attempting to even out the bands, it was proposed that they be left intact.)

Both the scores were added together to give a total score for a gene map on evaluation 2.

5.3 Evaluation 3 - Fit of total data

Evaluation 3 considered the gene map as a whole and awarded a score based on how well all the lengths of the observed fragments (both single and double digests) fitted into the map.

Evaluation 3 took as input a gene map and the observed data. The map was represented as a series of simultaneous equations where the intervals between the restriction enzyme cut sites were variables whose values were unknown. Each observed fragment length was "assigned" to an appropriate expected fragment from the map. The length of the expected fragment was the sum of the intervals from the position of the left-hand restriction enzyme cut site to the position of the right-hand restriction enzyme cut site in the map. Treating fragments in this way enabled equations to be generated. If the series of simultaneous equations could be solved, the lengths of the intervals between the cut sites could be determined. If all the intervals between the cut sites were positive, this indicated that the observed data fitted well into the map. If any interval was negative, this suggested that the restriction enzyme cut sites on either side of the interval were in the wrong order and that the sequence of restriction enzyme cut sites in the map was not correct. Such a map should not be feasible but as fragment lengths were allowed to vary (by up to 10% for fragments < 1000kb), some allowance for small negative intervals was taken into account. If an interval between cut sites was zero, this indicated that the cut sites were coincident. (Coincident cut sites arise when the recognition site of one restriction enzyme is contained within the recognition cut site of another. For example, a restriction enzyme which recognised "ATC" would cut the DNA one base before a restriction enzyme which recognised "GTCTATCG".)

If it was not possible to assign a minimal number of the observed digest lengths to the expected digests in the map, a situation arose where there were more unknown intervals than there were equations. As a result, the set of equations could not be solved. (The problem instance is said to be "under-determined".) In such cases where the observed data did not fit well into the map, the map was not considered feasible and was awarded a poor score. If there were more equations than there were unknown intervals, a transformation was applied to obtain the same number of equations and unknown variables (as described in Appendix F.)

In many maps, attempting to assign the lengths of the observed fragments to the fragments expected from a map was problematic. This was due to two factors - the number of cut sites producing digests and to differences between the number of fragments expected from a map and the number of fragments observed. The way in which the number of cut sites present in a map could create problems for the assignment of observed fragment lengths to expected fragments was the same problem that arose in evaluation 1, when fitting the single digest lengths, as described in section 5.1. Even when the same number of fragments were expected as observed, there were problems in assigning their lengths. If more or less fragments were observed than expected, it became even more difficult to calculate the position of any of the fragments. Inconsistencies between the number of fragments observed and expected was one of the characteristics of the map assembly problem, more commonly associated with double digests than single digests. The method for dealing with single digests was described in section 5.1. When there was a matching problem between the observed and expected data for the double digests, the decision was made to try out several different ways of fitting the observed double digests and to calculate the score for the map. The best score generated from the different possibilities was the one that was awarded to the map. If there were several options which resulted in an "ideal" scoring map, the map with the shortest length was considered the best map, based on the genticist's heuristic described in section 2 that the shorter the map, the better the fit of the data.

The score awarded by evaluation 3 was based on the number and size of any negative intervals in a map. If there were no negative intervals present, the observed data was considered to fit well and the map was awarded the best score. To accomodate errors in fragment lengths small negative intervals were tolerated. The size of the negative interval in relation to the length of the map was calculated and a score was awarded as shown below. If the map had more than one negative interval, the map was awarded a poor score.

Number of negative intervals	Size	Score
0	not applicable	0
1	< 1% of the map length	1
	< 2% of the map length	3
	> 2% of the map length	9
>1	not applicable	18

Calculating the distances between the cut sites in the map by solving the series of equations, produced a series of residual vectors that indicated the amount of error involved in solving the equations. As a preliminary measure, the residual information was not used to contribute to the map score.

6 Summary

An objective system for assessing maps was developed. A knowledge elicitation stage was carried out to identify and capture the expert's subjective judgements and different maps for the PIL/PI/AACT genes were analysed to gain insight into the type of discrepancies present in maps. A questionnaire was developed to estimate the relative importance of different map features to the overall quality of a map. A general mechanism for scoring maps that consisted of three types of evaluations was developed. The mechanism took as input a gene map and a set of observed data, applied three evaluations to generate a score for the gene map and classified the map as "ideal", "good", "adequate" or "unacceptable".

CHAPTER 5 - MAP EVALUATION RESULTS

1 Introduction

The general system developed for objectively evaluating single gene maps was described in chapter 4. The system comprised three evaluations that assessed various map features and awarded a score indicating how well the observed data fitted into the map. The total score classed a map as "ideal", "good", "adequate" or "unacceptable". Maps in the "ideal", "good" or "adequate" categories were all considered to be acceptable maps to the geneticist.

The results of testing the evaluation mechanism are described in this chapter. The evaluation mechanism was applied to a test list of maps generated using three different data sets. These included maps that contained no errors and maps the geneticist considered were optimal. To determine the effectiveness of the evaluation mechanism in identifying acceptable maps, all the maps possible for a particular data set were systematically generated. Each map was evaluated and categorised. The total number of maps belonging to each category was established. All the maps in the "ideal" category were examined in detail. As a result of these activities, it was proposed that the map evaluation mechanism was a plausible system for assessing gene maps.

2 Testing the evaluation mechanism

A test list of maps was compiled to test the ability of the evaluation mechanism in identifying optimal maps. The list contained maps generated using three different data sets - the first data set consisted of ideal data, the second and third consisted of experimental data. The three data sets are contained in Appendix A and are described in more detail below. The list of maps is contained in Appendix G.

To test the evaluation mechanism, it was proposed that the evaluation mechanism be applied to "perfect" maps generated from "perfect" data, ie where all fragments were present and all fragment lengths were accurate. A "perfect" data set was generated from the Sefton revised map for PIL/PI/AACT by calculating the "perfect" data expected from each gene map. The aim was to determine whether or not the evaluation mechanism was successful in handling the ideal case.

The second data set was the observed experimental data for PIL/PI/AACT taken from Sefton et al(1990). (As the data was experimental data, it contained errors.) The maps in the test list for this data set were those assembled manually by the geneticist and represented, in their opinion, the optimal maps for the data. Maps assembled manually by the author and proposed as optimal maps were included also.

The third data set used was the observed experimental data for AT/ACE taken from Sefton et al(1990). The maps were found to contain certain problems in the number of cut sites used to generate the single digests. Two minor amendments were made to the maps. Nevertheless, it was considered that the maps represented near-optimal maps.

It was not unusual for maps to contain coincident cut sites - ie obtained when the recognition site of one restriction enzyme was contained within the recognition site of another. As maps were input to the evaluation mechanism sequentially, each different map permutation was entered to accomodate coincident cut sites. For example, if the cut sites "B" and "S" overlapped in a map, two permutations would be evaluated - the map with permutation "BM" and the map with permutation "MB".

2.1 Results of applying the evaluation mechanism to the test list of maps

Each of the maps in the test list were input to the evaluation mechanism along with the appropriate data set and the maps were awarded a score. Map scores were expressed in units of minor problems. Depending on the score of the map, the map was categorised as an "ideal", "good", "adequate" or "unacceptable" map. An "ideal" map contained no problems, a "good" map tolerated between one and three problems, an "adequate" map tolerated between four and nine problems. Any map with more than nine problems was regarded as an "unacceptable" map. Note that a map that was "ideal", "good" or "adequate" was considered an acceptable map by the geneticist.

The results of applying each evaluation separately to the test list of maps are shown in table 5.1. Based on evaluation 1 alone, all the maps in the test list were either "ideal" or "good". Using evaluation 2 alone, all the maps in the test list were acceptable apart from the maps proposed by the author using data set 2. All the maps in the test list scored as acceptable using evaluation 3 alone. When all three evaluations were summed together to produce a total score, all maps apart from those proposed by the author were considered acceptable.

Map Identification	Map	Map Score				Classification
		E1	E2	E3	E123	
PPIL-1	B b F m S * F B S m m	0	0	0	0	IDEAL
PPIL-2	B b F m S * F B m S m	0	0	0	0	IDEAL
PPI1	B b m S F * f B f f F S m m	0	0	0	0	IDEAL
PPI2	B b m S F * f B f f F m S m	0	0	0	0	IDEAL
PAACT1	m B F b m S * F m S b b B	0	0	0	0	IDEAL
PAACT2	m B F b m S * m F S b b B	0	0	9	9	ACCEPTABLE
PIL-PUBR1	B b F m S * F B S m m	0	0	3	3	GOOD
PIL-PUBR2	B b F m S * F B m S m	0	0	0	0	IDEAL
PIL-PROP	B b F M S * F B S m M	0	18	3	21	UNACCEPTABLE
PI-PUBR1	B b m S F * f B f f S m m	0	0	0	0	IDEAL
PI-PUBR2	B b m S F * f B f f m S m	0	0	0	0	IDEAL
PI-PROP1	B b M S F * f B f F m S M	1	18	0	19	UNACCEPTABLE
PI-PROP2	B b M S F * f B f F S m M	1	18	0	19	UNACCEPTABLE
AACT-PUBR1	m F b m S * F m S b b B	0	0	0	0	IDEAL
AACT-PUBR2	m F b m S * m F S b b B	0	1	0	1	GOOD
AACT-PROP1	M B S m F * S M b F b B	0	18	0	18	UNACCEPTABLE
AACT-PROP2	M B S F m * S M b F b B	0	18	0	18	UNACCEPTABLE
AT1	S B s m b f * f B f f f s S f m F m m M	0	0	0	0	IDEAL
AT2	S B s m b f * f B f f f s S m f F m m M	0	0	0	0	IDEAL
ACE1	M F f f B S * f m b F S m B m M	0	1	0	1	GOOD
ACE2	M F f f B S * m f b F S m B m M	0	1	0	1	GOOD
ACE3	M F f f B S * f m b F m S B m M	0	1	1	2	GOOD
ACE4	M F f f B S * m f b F m S B m M	0	1	0	1	GOOD

Table 5.1 - Scores awarded to the test list of maps by evaluations 1, 2 and 3.

The test list contained three blocks of maps as shown. The first block contained the perfect maps for PIL,PI,AACT generated from the perfect data. The second block contained the geneticist's optimal maps (map identification PUBR) and the author's proposed maps (map identification PROP) for PIL,PI,AACT generated from the Sefton et al(1990) data. The third block contained the geneticist's optimal maps for ACE/AT generated from the Sefton et al(1990) data. Where there were coincident cut sites present, all the map permutations were evaluated (the map permutation number is the last figure in the map identification.)

3 Investigating the specificity of the evaluation mechanism

The number of maps possible for the PIL/PI/AACT genes, using the experimental data in Sefton et al(1990), was calculated in chapter 2. There were 4,055,040 maps possible for PIL; 57,507,840 maps for PI; and 8,847,360 maps for AACT. The revised published maps that the Sefton group assembled using the observed data were classified as "ideal" maps using the evaluation mechanism. How many other maps out of all the possible maps were also "ideal"? If there were very many, the evaluation mechanism would need to be revised to be more specific. In order to assess the specificity of the evaluation mechanism, it was proposed that every possible map for each of the PIL/PI/AACT genes be enumerated using the Sefton data. Each map had the three evaluations applied separately and together. Maps were categorised as "ideal", "good", "adequate" or "unacceptable". The results of this exercise are shown in table 5.2. (One of the reasons for opting for a heuristic method rather than a systematic method was the length of time required for systematic approaches as shown in table 5.3.)

Using all three evaluations together there were 112 "ideal" maps possible for AACT, 208 "ideal" maps for PI and 96 "ideal" maps for PIL. (As would be expected, the Sefton revised published maps were amongst the ideal maps.) When the "ideal" maps were examined in more detail, it was found that many maps had the same number of cut sites in the same order. The only difference between the maps was the nature of the cut sites (ie whether or not they were complete cut sites or partial cut sites). When the nature of the cut site was ignored, maps reduced to a basic form which was referred to as a "template". A template specified a particular number and sequence of cut sites around the gene, but ignored their nature. (The number of cut sites depended on the number of fragments in the data set, as illustrated in figure 2.4.) It was found that the 112 "ideal" maps for AACT reduced to 8 "ideal" templates, shown in figure 5.1; the 96 "ideal" maps for PIL reduced to 7 "ideal" templates, shown in figure 5.2; and the 208 "ideal" maps for PI reduced to 11 "ideal" templates. As 4 of these templates were identical(due to coincident cut sites), 7 unique templates are shown for PI in figure 5.3. Considering only templates reduced the number of possible maps for PIL from 4,055,040 to 15,840; for PI from 57,507,840 to 224,640; and for AACT from 8,847,360 to 34,560.

The "ideal" templates for each of the genes had the same number of cut sites but the sequence of the cut sites differed markedly between templates illustrating just how many alternative ways the same fragments could be assembled to give templates with unique sequences and varying lengths.

	AACT	PIL	PI
Number of maps in the search space	8,847,360	4,055,040	57,507,840
EVALUATION 1			
Ideal maps	1,439,744	1,233,920	11,161,600
Good maps	0	0	0
Adequate maps	238,080	584,192	3,206,144
Unacceptable maps	7,169,536	2,236,928	43,140,096
EVALUATION 2			
Ideal maps	2,800	1,928	13,392
Good maps	29,152	22,376	171,224
Adequate maps	169,488	44,992	483,208
Unacceptable maps	8,645,920	3,985,744	56,840,016
EVALUATION 3			
Ideal maps	28,288	16,896	14,336
Good maps	17,024	12,800	26,624
Adequate maps	376,384	179,712	336,384
Unacceptable maps	8,425,664	3,845,632	57,130,496
EVALUATION 1, 2, 3			
Ideal maps	112	96	208
Good maps	528	576	672
Adequate maps	2,568	1,240	2,640
Unacceptable maps	8,844,152	4,053,128	57,504,320

Table 5.2 - Results of categorising all the maps possible for the PIL/PI/AACT data set

	PIL	PI	AACT
IBM compatiblc PC (386 processor)	10.25 days	146.5 days	22.5 days
DEC Alpha AXP workstation	1.75 hrs	1 day 4 hrs 33 mins	4 hrs 53 mins

Table 5.3 - Computing time required to enumerate and evaluate all possible maps in the PIL/PI/AACT data set

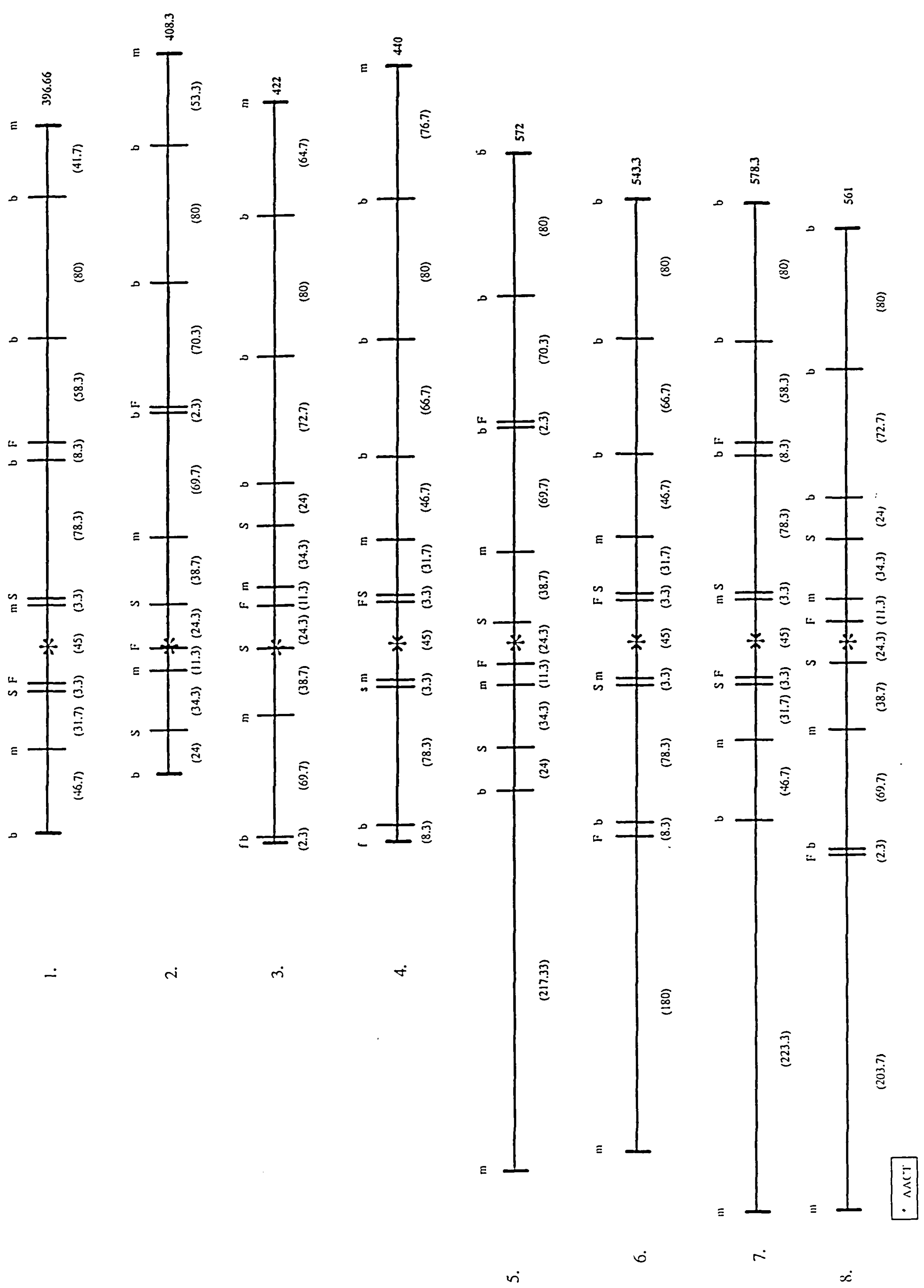


Figure 5.1 - All AACT "ideal" templates

These templates were found by generating and evaluating all 8,847,360 maps possible for AACT. All the templates have the same number of cut sites (11) and range in length from 396.6 to 578.3 kb.

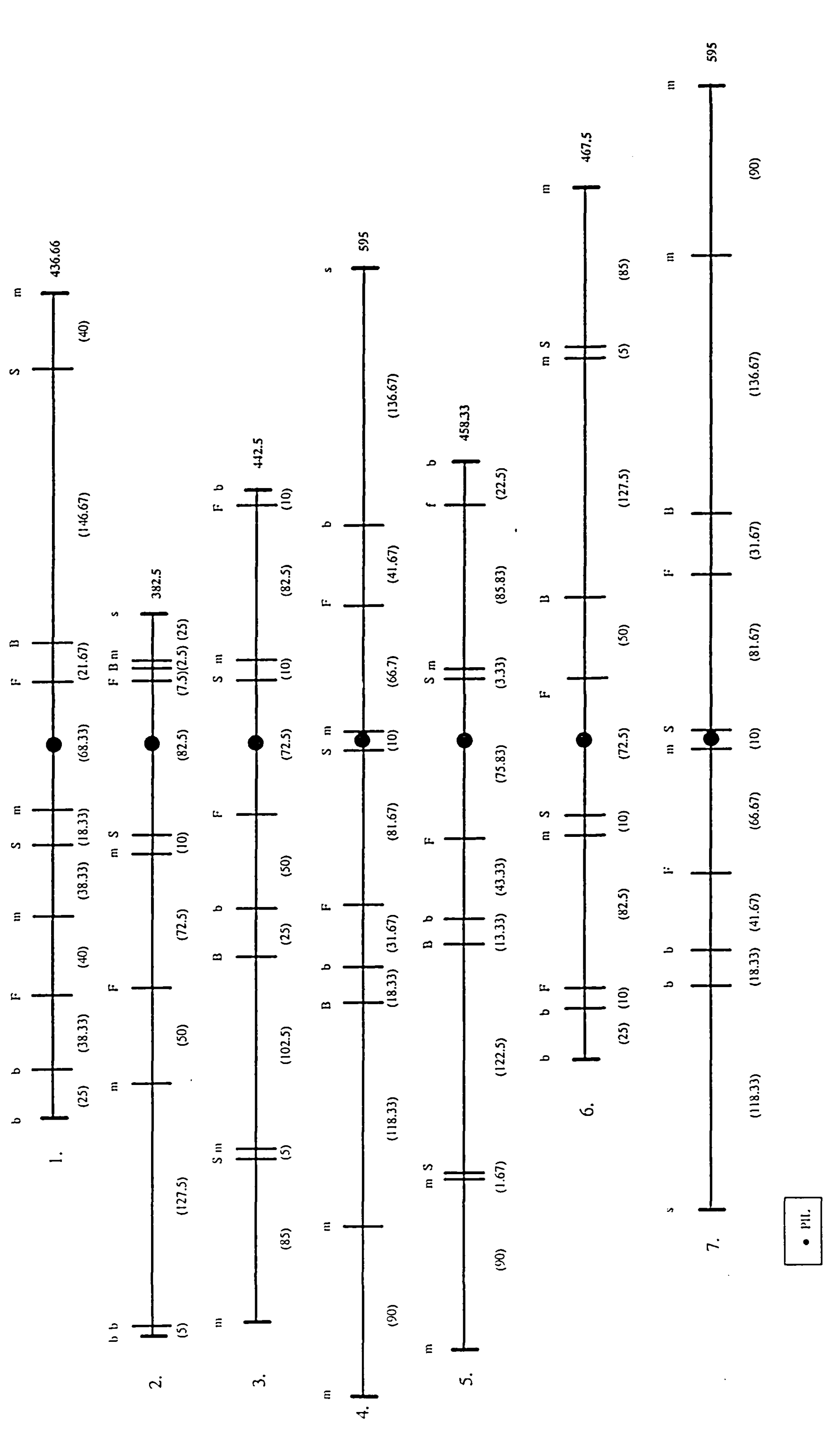


Figure 5.2 - All PIL "ideal" templates

These templates were found by generating and evaluating all 4,055,040 maps possible for PIL. All the templates have the same number of cut sites (10) and range in length from 382.5 to 595 kb.

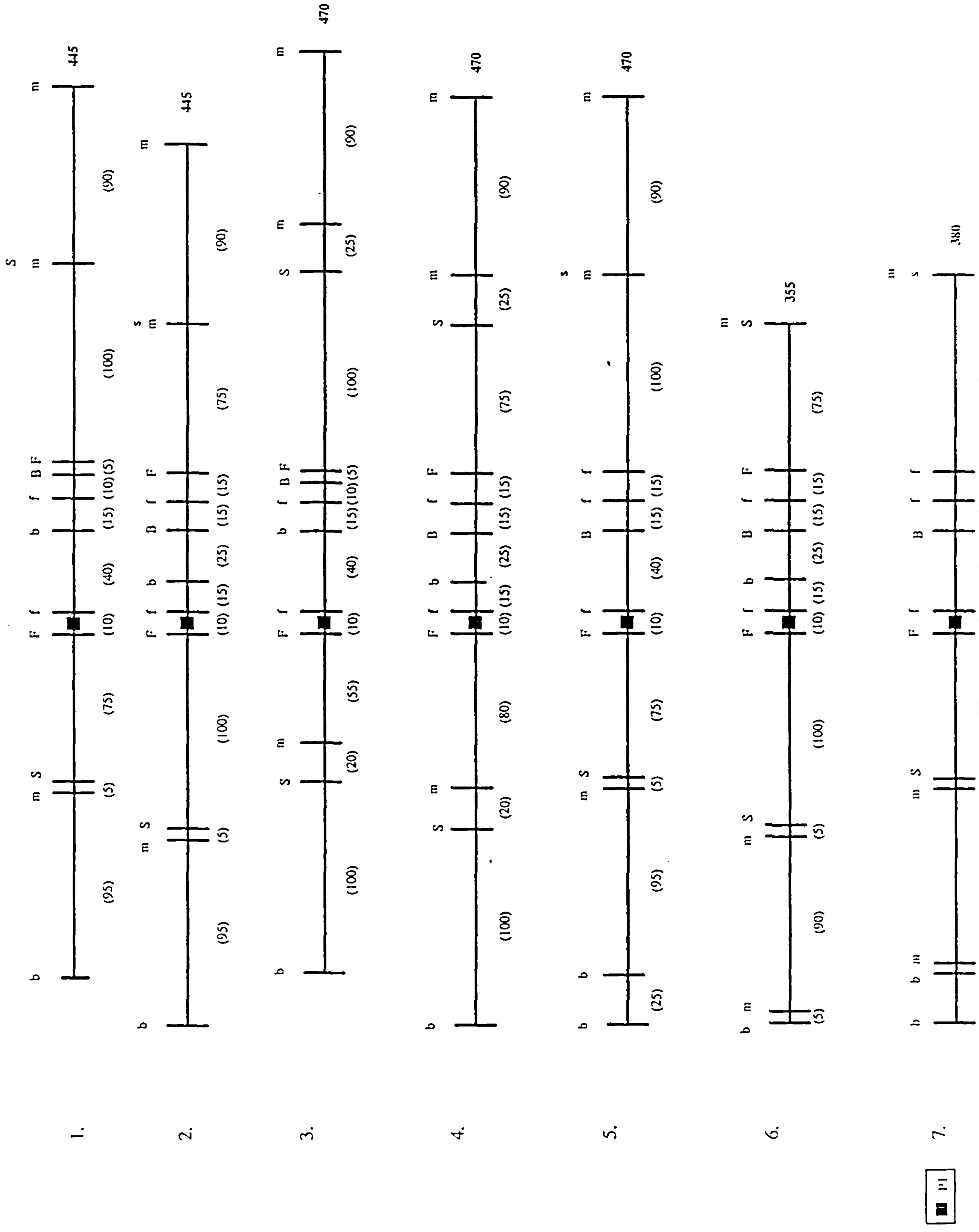


Figure 5.3 - All PI "ideal" templates

These templates were found by generating and evaluating all 57,507,840 maps possible for PI. All the templates have the same number of cut sites (12) and range in length from 355 to 470kb.

Examining the ideal templates in more detail indicated that they consisted of different cut site "options". (An "option" was the term given to the alternative ways that the restriction enzyme cut sites could be combined around a gene, as described in chapter 2, table 2.1.) For example, when the PIL templates in figure 5.2 were analysed, the first five maps had the B and M cut sites arranged as "bbmm*bm" and the remaining two maps had the B and M cut sites arranged as "bbm*mmb". The "ideal" templates ranged in length from 382.5kb to 595kb. In the AACT "ideal" templates shown in figure 5.1, four templates had the option "bm*mmbbb" and four had the option "bmm*mbbb". The "ideal" templates ranged in length from 396.6kb to 578.3kb. In the PI "ideal" templates shown in figure 5.3, there were four options - "fbm*bbmmfff"; "fbbm*bmmfff"; "bmmf*fffbbm" and "bbmmf*fffbm". The "ideal" templates ranged in length from 355kb to 470kb.

The percentage of "ideal" maps out of all the possible maps (using all three evaluations together) for PIL, PI and AACT was calculated to be 0.002%, 0.0004% and 0.001% respectively.

4 Discussion

When each of the evaluations was applied to the maps in the test list, only the Proposed maps (as generated by the author for data set 2) were classed as "unacceptable". Although the observed data fitted well into the Proposed maps as determined by evaluation 3, the yield and type of double digests observed and expected from the maps as calculated by evaluation 2, were not consistent.

All the "perfect" maps assembled from the "perfect" data(apart from PAACT2) were classed as "ideal" maps. PAACT2, was classed only as an "adequate" map, due to the evaluation from evaluation 3. In PAACT2, there were a large number of "BM" double digests to be fitted and the position of the fragments that would have resulted in an "ideal" map was overlooked. Nevertheless, PAACT2 was still acceptable and the alternative permutation of the cut sites, PAACT1 was "ideal". These results indicated that the evaluation mechanism was successful in identifying "perfect" maps as mainly "ideal" maps, but all as acceptable.

All the optimal maps, as determined by the geneticists, were classed as "ideal" or "good" maps by the evaluation mechanism.

From the results of the exhaustive search described in section 3 it was clear that evaluation 1, which evaluated maps on how well the observed single digest data fitted, was very broad. A large number of maps scored as acceptable based on evaluation 1 alone. It was more difficult for maps to score as acceptable using evaluations 2 and 3. Evaluation 2 scored a map on the yield and type of double digests expected from a map compared to the yield and type of double digests observed. Evaluation 3 scored a map on how well all the observed fragment lengths fitted into the map. Each of the evaluations applied individually were not sufficient to identify an optimal map. There were various problems with the "ideal" maps found by evaluations 1, 2 and 3 when applied individually. Generally all three evaluations were applied to identify optimal maps although as evaluation 1 was a subset of evaluation 3, applying evaluations 2 and 3 would have been sufficient. The number of acceptable maps using all three evaluations was low compared to the total number of maps in the search space, there were still too many for the geneticist to consider manually. The number of acceptable maps was narrowed down to those that were "ideal" maps, then further still to "ideal" templates. Considering only "ideal" templates resulted in a more manageable number of possibilities for PIL, PI and AACT. The "ideal" templates for each gene contained different cut site options which indicated that the "ideal" templates were not close to one another. As expected, templates for each of the revised published maps for the genes were among the "ideal" templates found by exhaustive search. Shorter "ideal" templates than the revised published maps were found for each of the genes which in theory represented better maps than the revised published maps. (Problems with these shorter templates were not apparent until the time came to merge the maps together to create a multi-gene map as described in chapter 8).

5 Summary

The results of applying the evaluation mechanism to the test list of maps showed that the evaluation mechanism could identify "perfect" maps assembled from "perfect" data as mainly "ideal" maps. The maps assembled by the geneticist and considered optimal were identified as "ideal" or "good" by the evaluation mechanism.

Given two competing maps generated from the same data, the evaluation mechanism was successful in discriminating between maps and highlighting map problems. This was illustrated using the author's Proposed maps and the geneticist's optimal maps. The evaluation mechanism discounted the Proposed maps as "unacceptable" due to inconsistencies between the number and type of double digests expected and observed. The reason why the Proposed maps were not better than the geneticist's maps had not been clear to date.

All possible maps from the Sefton et al(1990) PIL/PI/AACT data were systematically enumerated, evaluated and classified. The number of acceptable maps was reduced by defining the notion of a template. All the "ideal" templates present for each gene were shown. Better "ideal" templates than the geneticists maps were found for each of the genes. (These were discounted at a later stage when gene maps came to be merged together.)

CHAPTER 6 - DEVELOPING A HYBRID GENETIC ALGORITHM

1 Introduction

Automating map assembly required some means of generating potential gene maps and an approach using a hybrid genetic algorithm (HGA) was proposed. A Genetic Algorithm (GA), as introduced in chapter 3, is a type of search strategy based on the mechanics of natural selection that has had good success finding near-optimal solutions to a range of difficult problems with large search spaces and inexact data. A hybrid approach was chosen as the traditional GA could not be applied to sequencing problems and because the traditional GA was a general method. The development of the HGA is described in this chapter and the results of applying the HGA to generate potential maps from different data sets are contained in the next chapter. There were several areas that had to be addressed to develop an HGA, such as problem representation, choice of reproduction operators and choice of evaluation function, that were introduced in chapter 3. These issues are discussed in the following sections. The implementation of the HGA is outlined and methods for determining the success of the HGA are considered.

2 Representing the problem

In the "traditional" GA (as defined by Davis(1991a), "chromosomes" were represented as binary coded fixed length strings. When developing a hybrid GA, Davis(1991a) recommends that a representation which reflects the problem be adopted. Having considered several options, it was decided that "chromosomes" would be represented as single gene maps. A "chromosome" (gene map) consisted of a number of Restriction Enzyme(RE) cut sites and the gene, in a particular order as shown below. Here, four REs, "B","M","S" and "F" were used. The complete cut sites (those that cut every time) are indicated in upper case and the partial cut sites (those that sometimes cut) are indicated in lower case.

eg   B b M m S F * S M F B

The number of cut sites in the map depended on the number of single and double digest fragments observed experimentally.

3 Developing a set of genetic operators

The role of the genetic operators in a GA was to take either one or two potential solutions (referred to as "parents" in GA notation) to a problem, and to recombine them in such a way as to create potential solutions that were better still (referred to as "children"). The traditional operators for GAs consisted of one-point crossover and mutation, and relied on a binary representation. As binary representation was not being used and because the map assembly problem was a sequencing problem, different recombination operators to the traditional operators were required. To develop appropriate operators for map assembly, the way in which the geneticist approached the problem was examined and a way of breaking up the maps that was natural for the problem was sought, taking into account previous work on operators for sequencing problems (Goldberg and Lingle(1985), Davis(1985), Oliver et al(1987), Davis(1991a), Whitley et al(1989), Goldberg (1989), Whitley et al(1991) and Fox and McMahon(1991)).

Three operators were developed - side swap, order swap and case swap. Given a map, the operators allowed different maps to be reached within the "option" space of the map. An "option" was the name (as introduced in chapter 2, table 2.1) used to describe the several alternative ways that cut sites could be combined around a gene and yet generate the same data. The number of maps possible for a given data set was calculated by taking the number of single digests and calculating the number of ways these could be generated from different numbers of cut sites on either side of the gene (different options). Then, the number of ways the cut sites could be permutated for each option was calculated. Adding up the number of permutations for each map option arrived at the total number of maps possible for a single gene. It was proposed that the operators would permit searching within a map option space but would not permit different map options to be mixed. Using the operators to mix map options would be beneficial, in that it would allow all the maps possible to be reached; however it would create problems as to how such maps would be evaluated. The mechanism developed to evaluate maps was based on the likelihood that the number of single digests expected would be the same as the number observed. (It was more common for the number of double digests to vary and this was taken account of in the evaluation mechanism.) Mixing map orders would result in problems with assigning the observed fragment lengths to the fragments expected from the map. It was proposed that a reasonably large initial population of maps would be created to ensure that the various map options were present.

The operators are described in more detail in the following sections.

3.1 Side swap

The side swap operator was a modified form of the traditional crossover operator - crossover occurring at the position of the gene. Side swap swapped the left side of one parent map with the right side of the other parent map, as shown in figure 6.1. Side swap was permitted between two parent maps providing the resulting two children yielded the correct number of observed fragments. For example, consider the case below where two maps are shown, each consisting of a series of RE cut sites and a gene, indicated by an asterisk.

map 1 - B b M S * b B M S

map 2 - B M S * b b b B M S

Four single digest fragments would be observed for the gene using RE "B". There would be three ways the "B" cut sites could be arranged around the gene to give four fragments - "B b * b B" (as shown in map 1), "B * b b b B" (as shown in map 2) or "B b b b * B" (map 2 reversed). If side swap were to occur between these two maps, two child maps would be obtained as shown. From child map 1, eight "B" fragments would be expected and from child map 2, two "B" fragments would be expected.

child map 1 - B b M S * b b b B M S (8 "B" fragments)

child map 2 - B M S * b B M S (2 "B" fragments)

It was determined from discussions with geneticists, that it was likely that more fragments would be expected from a map than would be actually observed, as fragments could go missing or could be misinterpreted. This situation was considered typical for double digests results, due to the experimental technique required to perform double digests. It was not considered to be as common for single digests results as the single digest results tended to be much more accurate. This was one of the reasons why side swaps that would result in an inconsistent number of fragments were disallowed. The other reason was the difficulties that such a situation would create when it came to evaluating the maps. The evaluation mechanism determined how well a set of observed experimental data fitted into a map. In evaluations 1 and 3, the observed single digest lengths had to be assigned to positions in the map. Even when there was the same number of observed and expected fragments, there was uncertainty with assigning the lengths (as described in detail in chapter 4 section 5.1.) So, it was proposed that the side swap operator only be applied to two maps that would result in the correct number of single digests expected as observed. This constraint did not apply to double digests as it was more likely that there would be differences in the number of double digests expected and observed and a method of handling the inconsistencies had been incorporated into evaluation 3 (as described in chapter 4, section 5.3.)

3.2 Order swap

Normally, there are two types of map that the geneticist can refer to for the region of DNA they are interested in, a "framework map" and a "comprehensive map". A framework map shows amongst other things, the position of genes, RE cut sites and probes for a particular region of DNA. The information within a framework map is very reliable as it has been verified by several sources. A map that is still under trial is called a comprehensive map. A comprehensive map shows the same type of information as the framework map however the information within it is not supported to the same degree as the information in the framework map. The geneticist has most confidence in the information described in a framework map. The geneticist will attempt to fit the sequence of genes and cut sites in a potential map, into the comprehensive map. The geneticist examines the consequences of moving cut sites one position to the left or right of their current position - a concept known as "local support".

An operator called "order swap" was developed to allow for cut sites to be moved on either side of a map, implementing the local support concept. Order swap was considered a mutation operator as it carried out local modification of a single map. It introduced variation within each side of a map by swapping the order of two different REs providing the swap was legal. A swap was considered legal providing a complete cut site was not moved closer to the gene than any of its partial cut sites. As a complete cut site always cut, it should not be possible to obtain fragments as a result of a more distant cut site cutting. A partial cut site can not be moved further away from the gene than its complete cut site. Order swap is illustrated in figure 6.2.

3.3 Case Swap

Some mechanism for enabling the nature of cut sites (ie whether or not the cut site was partial or complete) to be altered was required. The position of a complete cut site in a map prevented any of the same type of cut sites further away from the gene from producing fragments. The position of a partial cut site enabled longer fragments to be generated as illustrated in chapter 2, figure 2.4. Generally, more double digests tended to be expected from a map than were observed. To allow for variation in the number of double digest fragments expected, it was proposed that an operator for changing the nature of cut sites was required. A "case swap" operator was devised. Case swap swapped the outermost cut site for a particular RE to partial if complete, and to complete if partial. The operation of case swap is illustrated in figure 6.3. Case swap worked by randomly selecting a RE and a side of map (left or right) and changed the case of the outermost cut site for the RE.

4 Developing the evaluation function

A GA requires some method of evaluating the goodness, or "fitness" (using GA terminology) of potential solutions. This is referred to as the evaluation function. The way in which potential solutions are assessed is critical to the success of any GA as this has a direct influence on the parents of the next generation. For the map assembly problem, it was essential that an evaluation function captured the essence of a good or bad map. The evaluation function for assessing maps was based on the map evaluation mechanism developed and tested in chapters 4 and 5. The mechanism awarded a score to a map that indicated the goodness of fit of the observed experimental data in a potential map.

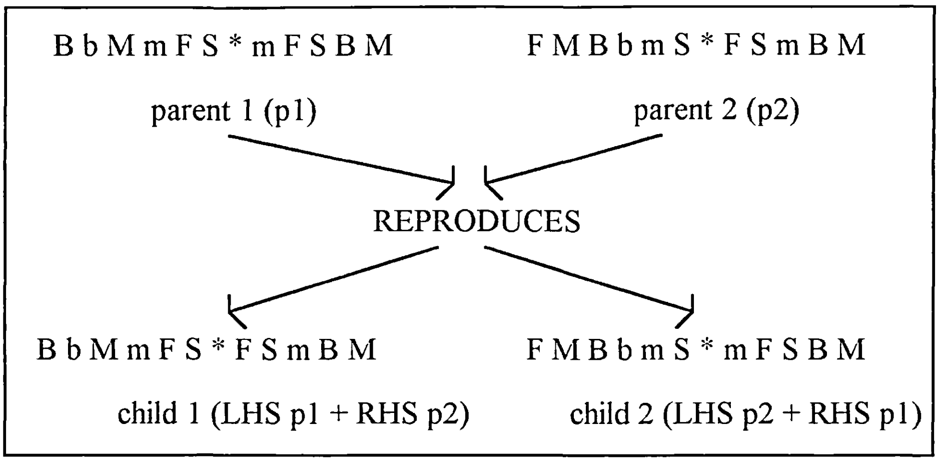


Figure 6.1 - Side swap

B b**M** S **F*** b M F B S

position 0 1 2 3 4 5 6 7 8 9 10

order swap positions 2/4 > B b**F** S **M** * b M F B S

Figure 6.2 - Order swap

B b**M** S F * b M F B S

position 0 1 2 3 4 5 6 7 8 9 10

case swap position 2 > B b**m** S F * b M F B S

Figure 6.3 - Case swap

The mechanism comprised three evaluations : evaluation 1 which assessesed the fit of the single digest data; evaluation 2 which assessesed the nature and yield of the double digest data; and evaluation 3 which assesses the fit of the total data. The mechanism was applied to a test list of maps assembled from three different sources of data. The results, shown in chapter 5, indicated that the evaluation mechanism was successful in identifying perfect maps and the maps that the geneticist considered optimal.

5 Setting parameter values

There are several parameters in a GA that require to be set to appropriate values - population size, number of trials and operator probabilities. There are established parameter settings described in the literature for GAs using binary representation, binary crossover and mutation (Schaffer et al(1989)) however, finding good setting for non-binary representations is not a trivial task (Davis(1989)). Poor settings for parameter values can have a profound impact on GA performance. In order to investigate the effect of changing parameter settings, the HGA was equipped with a front-end menuing system to allow parameters to be changed easily to tune performance.

5.1 Population size

The size of the population is an important parameter in a GA. If the population size chosen is too small, the GA tends to end up with a population consisting primarily of similiar individuals (population has "converged") with insufficient processing of too few types of potential solution. If the population size is too large, the GA tends to take a long time before significant improvement is achieved as there is not enough mixing of building blocks per unit of computation time. Goldberg(1989) carried out a theoretical analysis of optimal population sizes which suggested that optimal population size for traditional GAs increased exponentially with the length of the string. However, empirical studies have found that population sizes of 50-100(DeJong(1975)), 30 (Grefenstette (1986)), 20-30(Schaffer et al(1989)) and a value between n and 2n (where n=string length) (Alander(1992)) were optimal. Reeves(1993b) investigated the performance of GAs with very small population sizes and reported that small populations were adequate for binary strings but for non-binary strings, larger population sizes were necessary.

DeJong(1975) suggested good performance be obtained from high crossover probability/low mutation rates and moderate population size for traditional GAs.

For the experiments conducted here, the size of the population and number of trials were varied to optimise performance. Reasonably large populations were chosen to ensure that all the map options for a data set were present. For each experiment, 20 runs were performed.

5.2 Number of trials

The number of trials or generations to run is closely linked to the size of the population. In general, a GA either runs until the population converges, or for a fixed time slot.

5.3 Operator probabilities

The rate at which operators are applied while the GA is running has to be set appropriately. The optimal rate depends on the reproduction technique employed. If a policy of replacing the whole population at each generation is adopted (refered to as "Generational Replacement without Elitism"), the rate at which the operators are applied needs to be kept low to ensure that the good material is not all lost. However, if the best solution is preserved and automatically put forward to the next generation, an "elitist" strategy, the operator rates can be higher as the best material is preserved. Traditional GAs tended to use a Generational Replacement with Elitism policy and studies found crossover rates of 0.6 (DeJong(1975)), 0.95(Grefenstette(1986)) and 0.75-0.95 (Schaffer et al(1989)) to be optimal. The optimal mutation rate tended to be much lower at 0.001 (DeJong(1975)), 0.01 (Grefenstette(1986)) and 0.005-0.01 (Schaffer et al (1989)).

Three operators were used: side swap, order swap and case swap and the rates of applying them were varied.

6 Generating the initial population

Traditional GAs generated the initial population at random. However, it was recognised that if domain specific knowledge was available, it could be usefully exploited in the GA. It was decided that the HGA would use the domain specific information in the form of the experimental data to generate maps. A "gene map builder" was developed which took as input the single digest data for a gene (as shown in chapter 1, table 1.1) and generated as many maps as specified by the population size. Given the number of strong fragments per digest, the way in which the cut sites could be ordered were calculated. For each map, an order for each digest was selected at random and legal gene maps for the initial population were created.

7 Selecting a reproduction technique

Three main types of reproduction technique are described in the literature and reviewed by Davis(1991a) - generational replacement(GR), generational replacement with elitism(GRE) and steady-state replacement without duplicates(SSWD). With GR, all the members of the parent generation were replaced by the new generation of children. The traditional GA used GR normally with an "elitist" strategy, copying the best member of the current population into the new population. Although in some cases elitism increased the speed of dominance of a super individual, it appeared to improve GA performance.

Whitley(1988) and Syswerda(1989) investigated modifying the reproduction technique so that only one or two individuals were replaced at a time - a technique Syswerda called "steady-state"(SS) reproduction. Only a small number of new potential solutions were created in each generation. The same number of members were deleted from the population and selection for deletion was done through inverse ranking, starting with the worst member in the population. A form of automatic elitism existed in that good members tended not to be deleted. Syswerda(1989) believed that SS worked better than the standard GR technique for two main reasons. Firstly, that SS was not as susceptable to higher error rates and secondly, that SS took better advantage of good schemata in the population. Initially researchers found that this technique did not do as well as GR, however, it was found that if SS replacement was applied with the strategy of discarding any duplicate children, resulting in every population member being different, it out performed GR. This technique was known as "steady-state replacement without duplicates".

All three reproduction techniques were included as options in the HGA.

8 Selecting a parent selection technique

The purpose of parent selection is to give a greater reproductive chance to the most fit members of the population. There are several ways this can be done which are described in the literature (DeJong(1985),Brindle(1981) and Goldberg(1989)). One of the most well known and commonly used parent selection techniques is called Roulette Wheel Parent Selection and for these reasons, it was chosen for the HGA. Roulette wheel parent selection works by summing the fitnesses of all the population members to obtain the total fitness. A random number between zero and the total fitness is generated and the first population member whose fitness when added to the fitness of the preceeding population members is greater than or equal to the random number is returned.

9 Selecting a fitness technique

The method of converting the evaluation figure into a fitness value is called the fitness technique. The simplest technique is to assign the value arrived at by the evaluation function as the fitness value, however, there are problems with this approach. For example, if there is a potential solution in a population with a very high fitness value and the rest of the individuals have low fitness values, it is likely that the highly scoring individual is selected for reproduction more often than the low scoring individuals and after a number of generations, the population prematurely converges. The low scoring individuals would not have had much chance to be selected and the amount of recombination between the lower scoring individuals and the high scoring individuals would be low. Another problem arises if there are a number of individuals with fitness values clustered together. It is possible that there is not enough of a difference to allow the better ones to reproduce more often.

There are scaling techniques described in the literature, such as "windowing" and "linear normalisation", that can be used to convert the evalution figure into a fitness value. The aim of these techniques is to ensure that appropriate levels of competition are maintained through a run by re-scaling the evaluation values to produce a distribution which smoothes out exceptionally high evaluations or widens a band of similiar evaluations.

For the HGA, linear normalisation was chosen as the fitness technique as it handles the problems of very highly fit individuals, and tightly packed individuals well(Davis 1991a). The maps were ordered by increasing evaluation. The map with the lowest score was assigned the top fitness value of 100. Subsequent maps were assigned fitness values which decremented at a constant rate. As a preliminary measure, the decrement value chosen was 5. The next lowest score was assigned the fitness value 95 and so on. The minimum fitness value was set at 1.

10 Implementation issues

As it was essential that the HGA parameters were varied until satisfactory results were achieved, it was decided that a "front-end" menuing system would be developed to allow for various features to be selected. This would facilitate implementation and allow the effect of changing parameters to be observed. The HGA program was written in Microsoft C version 6 and developed on an IBM compatible PC. The software package Liant C-scape was used to assist with the user interface. The main steps of the program are illustrated in figure 6.4.

11 Determining the success/limitations of the HGA

Silver(1980) discussed four properties of a good heuristic. A good heuristic should : perform well on average; minimise the chance of a very poor solution; be simple to understand; and should use a realistic amount of computational effort. Reeves(1993a) considered how the performance of a heuristic could be measured by using analytical methods, empirical testing and statistical inference. The success of a GA in any application area can only be determined by experimentation (Davis(1991a), Bagchi(1991)).

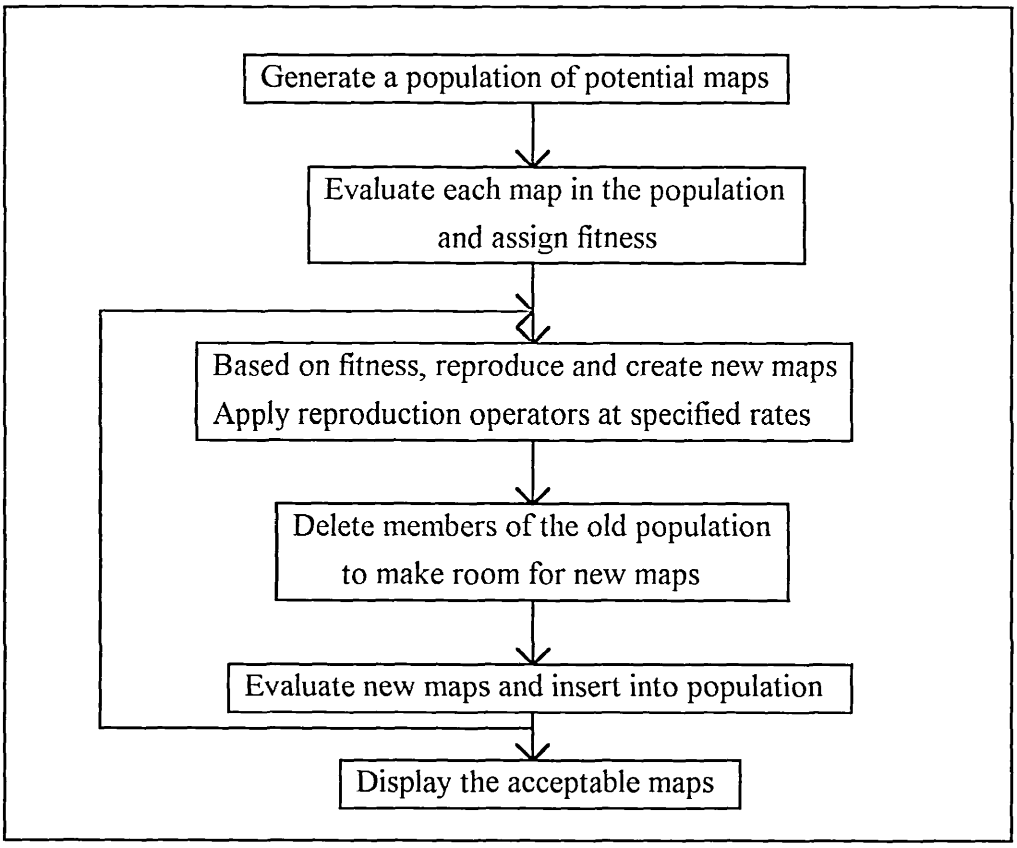

Generate a population of potential maps
Evaluate each map in the population and assign fitness
Based on fitness, reproduce and create new maps
Apply reproduction operators at specified rates
Delete members of the old population to make room for new maps
Evaluate new maps and insert into population
Display the acceptable maps

Figure 6.4 - Hybrid genetic algorithm description

(Note, the process is repeated either for a fixed period of time or until the population has converged.)

The performance of the HGA was assessed in five different ways -

* the number of acceptable maps found by the end of a run;

* the average map score at the end of a run;

* comparison of performance of the HGA with the performance of a similiar program that generated new maps at random, rather than using the operators and fitness proportionate selection of the HGA;

* the amount of time the HGA took to generate acceptable maps was compared with the time required for an exhaustive search; and

* the percentage of ideal templates found by the HGA with respect to the fraction of search space examined.

An acceptable map was an "ideal", "good" or "adequate" map. The number of acceptable maps generated was a result of the score of the maps. In general, a population of maps would initially have a high average map score which would hopefully improve over the run. When a map in the population achieved a score of nine or less, the map was considered acceptable. At the end of each HGA run the acceptable maps generated were analysed and any enhancements that could be made to the evaluation function to improve the search were incorporated, where possible. Whether or not the optimal map, as assembled manually by the geneticist, was present at the end of a run was noted. The effectiveness of the different components of the evaluation function (evaluations 1,2 and 3) in identifying acceptable maps was determined by comparing the actual number of acceptable maps, as established by exhaustive search, with the number of acceptable maps generated using the HGA. Two types of performance graphs were produced showing the average number of acceptable maps generated over a number of trials and showing the average map score over a number of trials.

The difference in performance between two or more runs was measured by comparing the number of acceptable maps or average map score over the run. The number of acceptable maps or average map score was plotted on the y-axis of a graph where the x-axis represented the number of trials in the run. The points on the y-axis were transformed into straight lines, where necessary, generally using a logarithmic or reciprocal transformation. This produced a straight line for the data for each of the runs. An analysis of covariance was applied to the two or more lines to determine whether or not all the points could be fitted by one line. The null hypothesis was that a single line was sufficient to fit all the points from the two or more runs. If the null hypothesis was rejected at the five percent level, the runs were considered to be significantly different.

12 Summary

A hybrid genetic algorithm(HGA) was developed to generate single gene maps given a set of experimental data. Gene maps were represented in GA notation; three problem specific operators were devised; the evaluation function was chosen; and the options for other features were outlined. Ways of determining the success of the HGA were considered. The results of applying the HGA to different data set are contained in the next chapter.

CHAPTER 7 - HYBRID GENETIC ALGORITHM RESULTS

1 Introduction

The results of using the HGA to generate acceptable single gene maps from three different sets of data are described. The principal data set used in the experiments to determine the parameter settings for the HGA was the Sefton et al(1990) PIL/PI/AACT data set. Once reasonable parameter settings had been achieved, the HGA was run using two further data sets, the perfect PIL/PI/AACT data and the ACE/AT data. (All three data sets are contained in Appendix A.) Various modifications were made to the HGA as a consequence of the results. (The four versions of the HGA used are summarised in Appendix H. They are referred to as HGAv0, HGAv1, HGAv2 and HGAv3, respectively.) The performance of the HGA in generating acceptable gene maps is discussed.

2 HGA results using data set 1 - PIL/PI/AACT

Several experiments were performed to tailor the HGA to the map assembly problem and the experiments and their results are described in the following sections. As the HGA was a probabilistic algorithm, twenty runs were conducted to calculate the average performance (unless otherwise stated). (That twenty runs were sufficient was established by running the same experiment three times (sixty runs in total) and calculating that there was no significant difference between the results of the three experiments.)

Three different reproduction techniques were available in the HGA: Generational Replacement (GR); Generational

Replacement with Elitism (GRE); and Steady-State(SS). Early experiments using GR showed that although acceptable gene maps were produced during the course of a run they tended to get lost. The GRE technique was used for several experiments and the effect of varying the population size was investigated for AACT. Populations of 80, 40 and 20 maps were used. The large populations converged quicker than the smaller population, but took more time. Although there appeared to be a high proportion of acceptable maps in the population (between 25% - 50% by trial 30), many of the maps were identical which made interpreting the results complicated. For this reason, the decision was made to use a Steady State reproduction technique for future experiments.

2.1 Varying the evaluation mechanism

The effect of the different evaluations on the number of acceptable maps was investigated for AACT. Evaluations were applied separately (Evaluation 1(E1), Evaluation 2(E2), Evaluation 3(E3)) and in combinations (E12, E13, E23, E123) and the results are shown in Figure 7.1. The number of acceptable maps using evaluation 1 alone was very high - 80% of the population consisted of acceptable maps by the end of the run. The number of acceptable maps obtained at the end of the run using evaluations 2 and 3 alone was lower than evaluation 1 as it was harder for maps to score well on these evaluations.

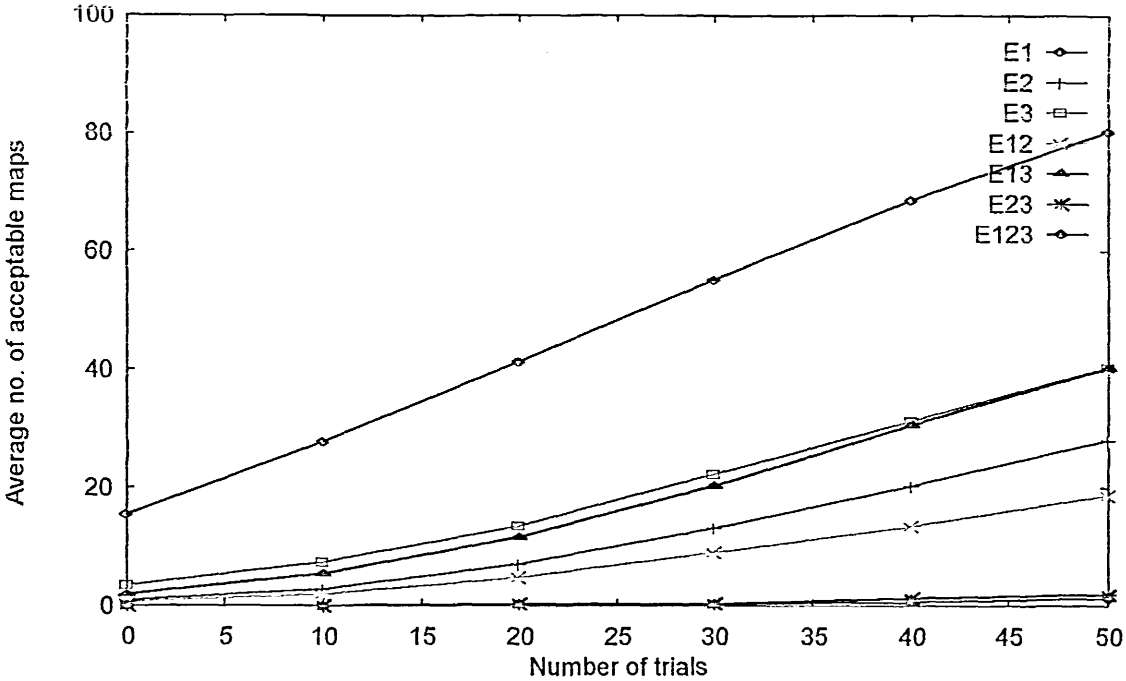


Figure 7.1 The effect of the different evaluations on the number of acceptable maps for AACT using HGAv1. (Steady-State reproduction technique used replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

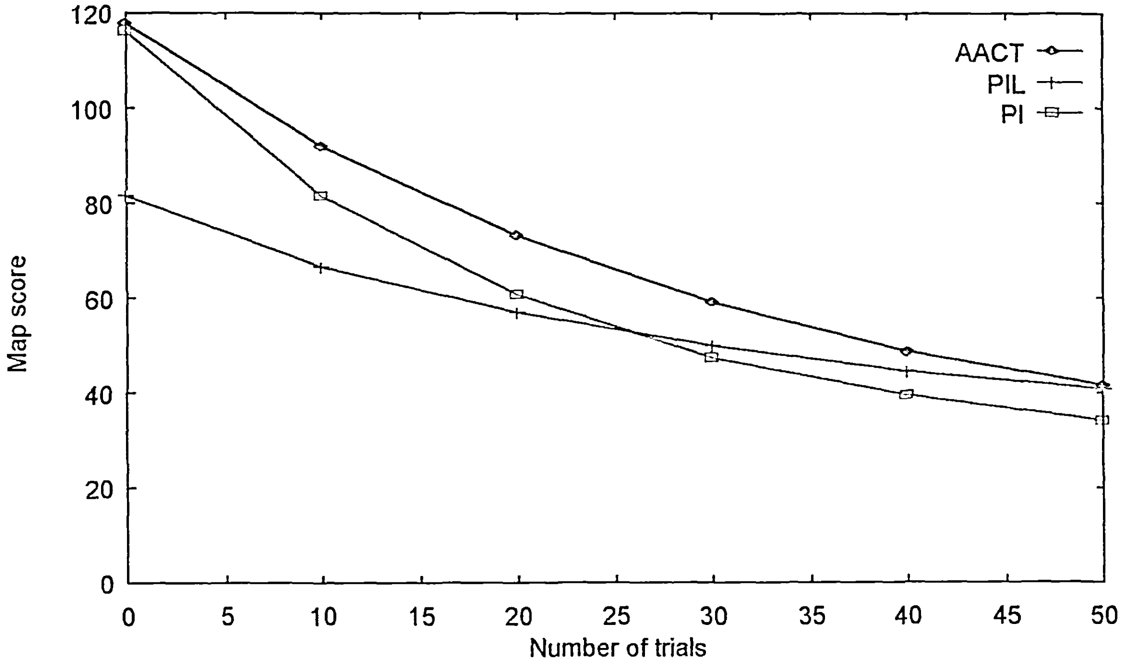


Figure 7.2 The change in the average map score for PIL, PI and AACT using evaluations 1,2 and 3, HGAv1. (Steady-State reproduction technique used replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

Using all three evaluations together produced a low number of acceptable maps at the end of 50 trials for AACT(1.25),PIL(1.65) and PI(2.8). The corresponding change in average map score over the run is shown in figure 7.2 for PIL, PI and AACT (using all three evaluations, HGA v1). Although there was a significant improvement in map score over the run (64.5% for AACT; 49.9% for PIL; and 70.6% for PI) it did not have a great impact on the number of acceptable maps. This was probably because the average map score had to improve to nine or less before a map was classed as acceptable.

2.2 Tailoring the reproduction technique

A "Steady-State-Without-Duplicates" (SSWD) reproduction technique was introduced in an attempt to improve upon the results shown in section 2.1. Figures 7.3 and 7.4 show the effect on the number of acceptable maps of applying the evaluations separately and in different combinations for PIL and PI using the SSWD reproduction technique. Figure 7.5 shows the effect on the number of acceptable maps of applying all three evaluations together for AACT. The change in the average map score for PIL, PI and AACT applying all three evaluations together is shown in figure 7.6. When all three evaluations were applied together the number of unique acceptable maps increased significantly for AACT and PI. The average map score improved over the run as shown in Figure 7.6.

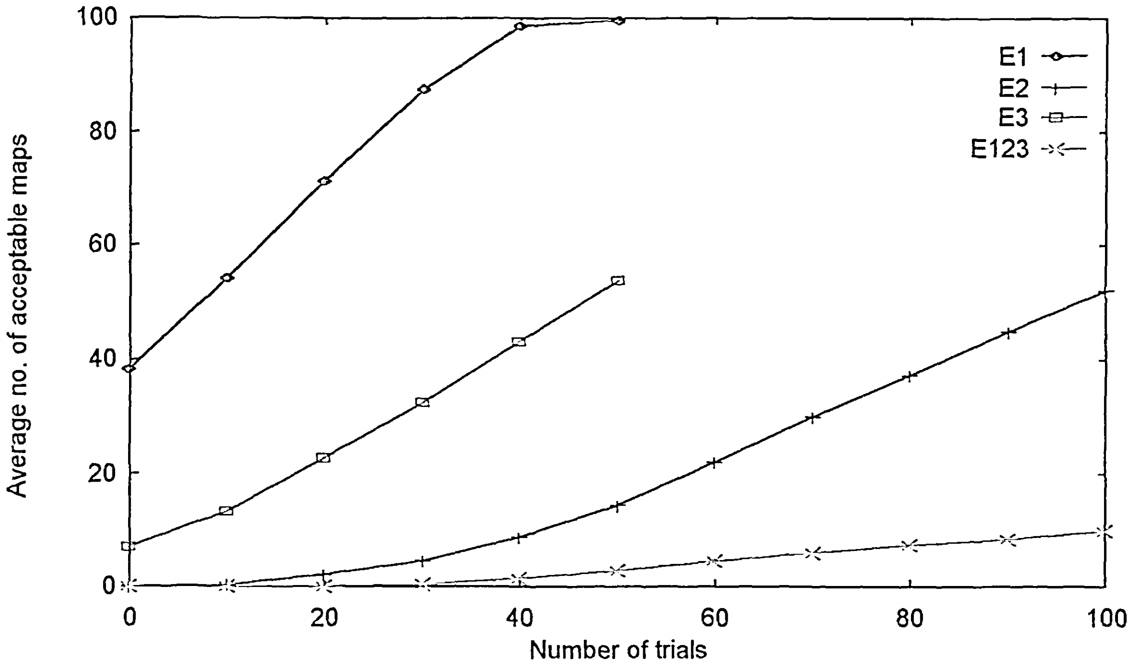


Figure 7.3 The effect of the different evaluations (E1,E2,E3,E123) on the number of acceptable maps for PIL using HGAv2. (Steady-State-Without-Duplicates reproduction technique was used, replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

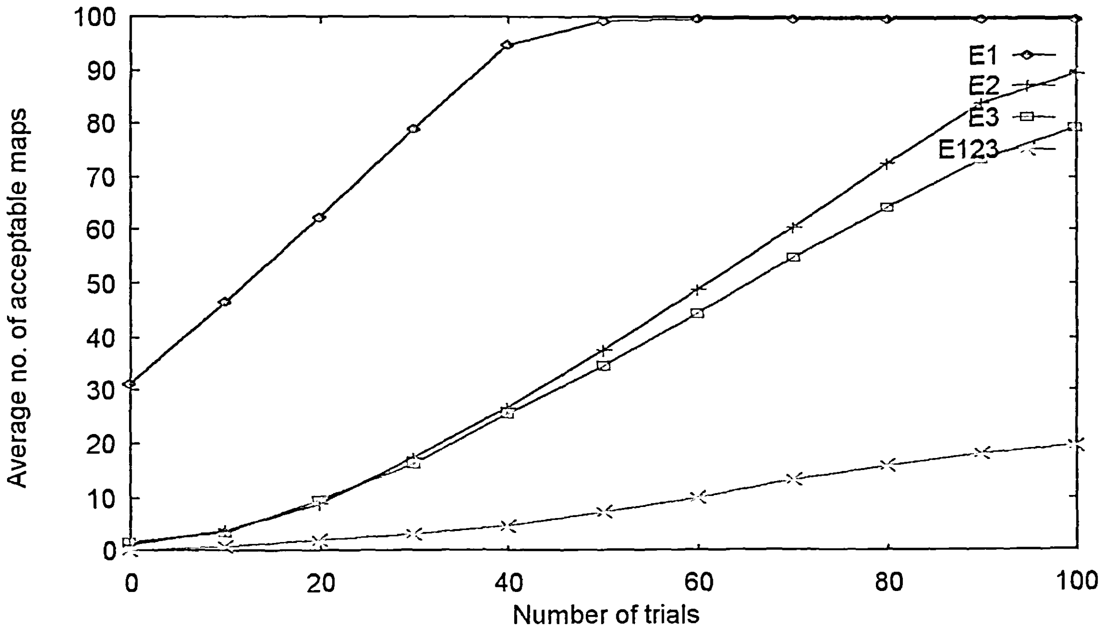


Figure 7.4 The effect of the different evaluations (E1,E2,E3,E123) on the number of acceptable maps for PI using HGAv2. (Steady-State-Without-Duplicates reproduction technique used, replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

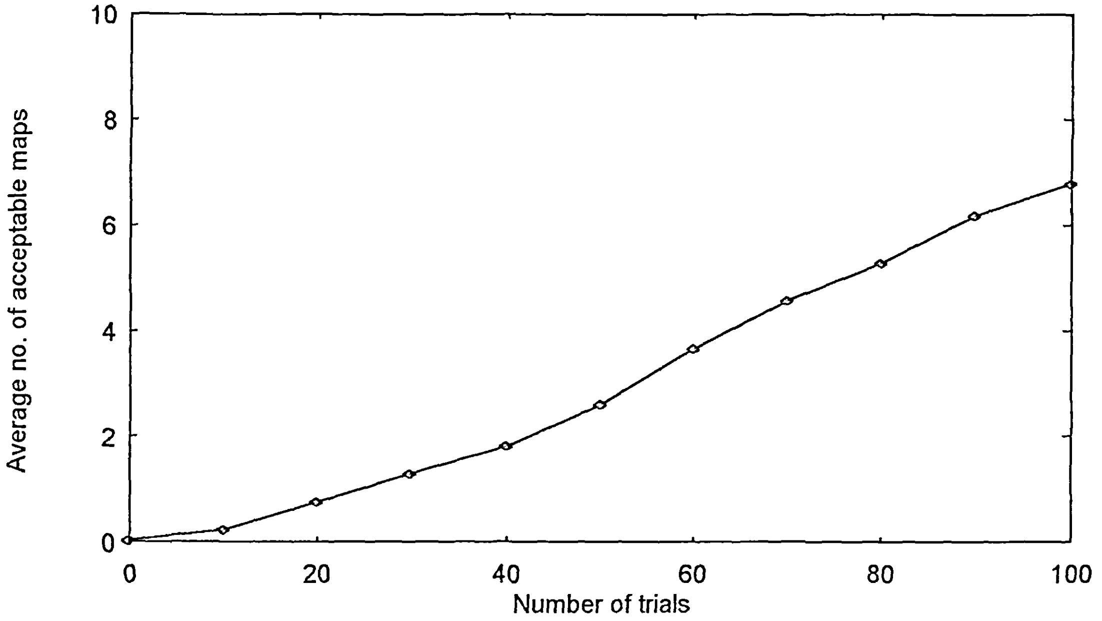


Figure 7.5 The number of acceptable maps for AACT using evaluations 1,2 and 3 and HGAv2. (Steady-State-Without-Duplicates strategy used, replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

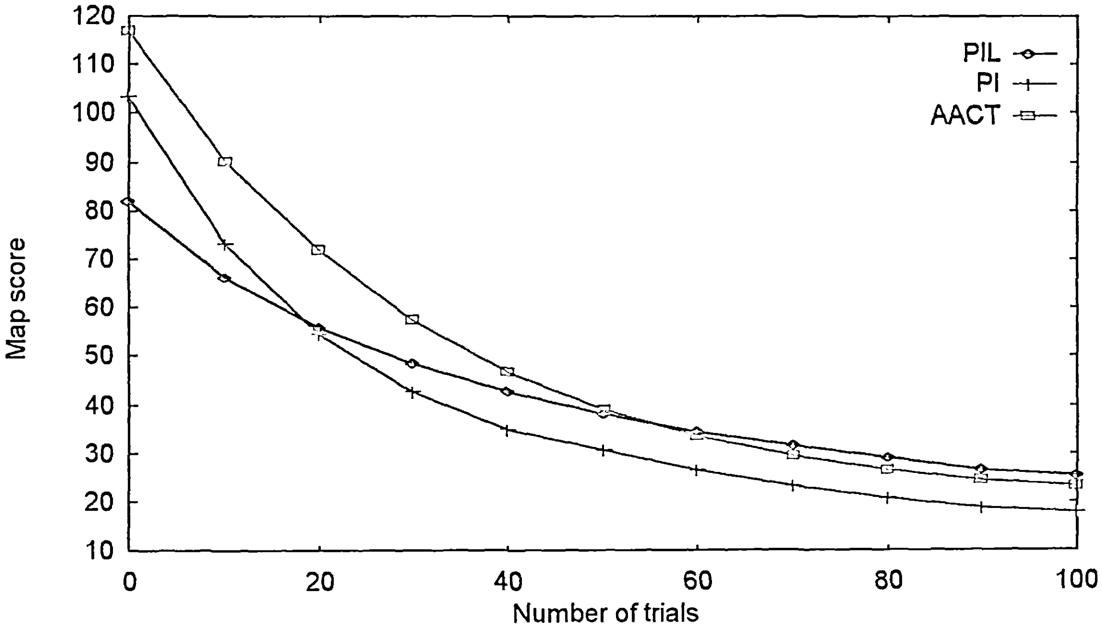


Figure 7.6 The change in average map score for PIL,PI and AACT using evaluations 1,2 and 3, HGAv2. (Steady-State-Without-Duplicates strategy used, replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

2.3 Cutting down on duplicate templates and illegal maps

Although there were no identical maps present, there were many maps which had the same number of cut sites in the same position which differed only in the nature of the cut sites (whether or not the cut site was partial or complete). It was conjectured that much of the search effort may have been spent modifying the same basic map with cut site variations. In an attempt to broaden the search and improve performance, the SSWD reproduction technique was adjusted to become a "steady-state without duplicate templates"(SSWDT) technique. A map template was defined as a map where the nature of the cut sites was not relevant. The SSWDT technique did not allow duplicate map templates to be added to the population unless the new template scored better than an existing template.

Further examination of the acceptable maps revealed a high proportion of maps that scored well using evaluation 3. With evaluation 3, it was possible that intervals between cut sites could be zero which allowed for the representation of coincident cut sites. A problem arose when the interval over the gene was set to zero as this had the potential of creating an illegal map. Many of the maps that had scored well using evaluation 3 had a zero interval across the gene. Evaluation 3 was revised to check for and penalise such a situation. (Modifying the reproduction technique and revising evaluation 3 resulted in HGA v3.) The effect of these changes on the number of acceptable maps and the map score is shown in figures 7.7 and 7.8 for PI.

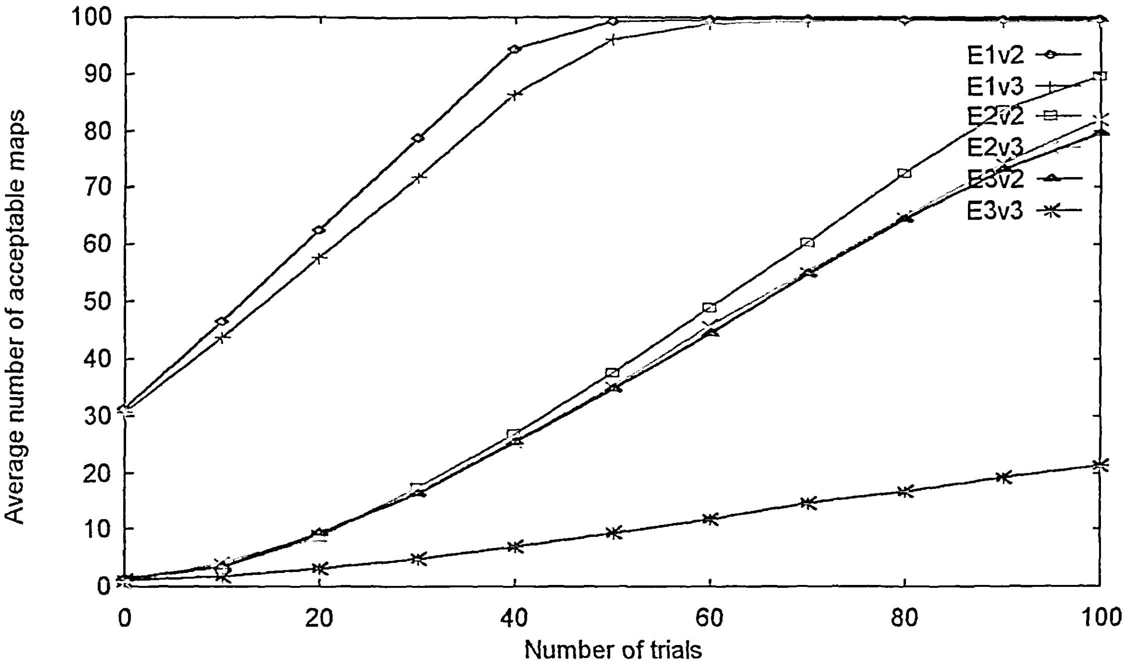


Figure 7.7 The effect of disallowing duplicate templates between HGA v2 and v3 on the no.of acceptable maps for PI using evaluation 1(E1), evaluation 2(E2) and evaluation 3(E3). (SSWDT strategy used; operators applied 30% each; and population size = 100.)

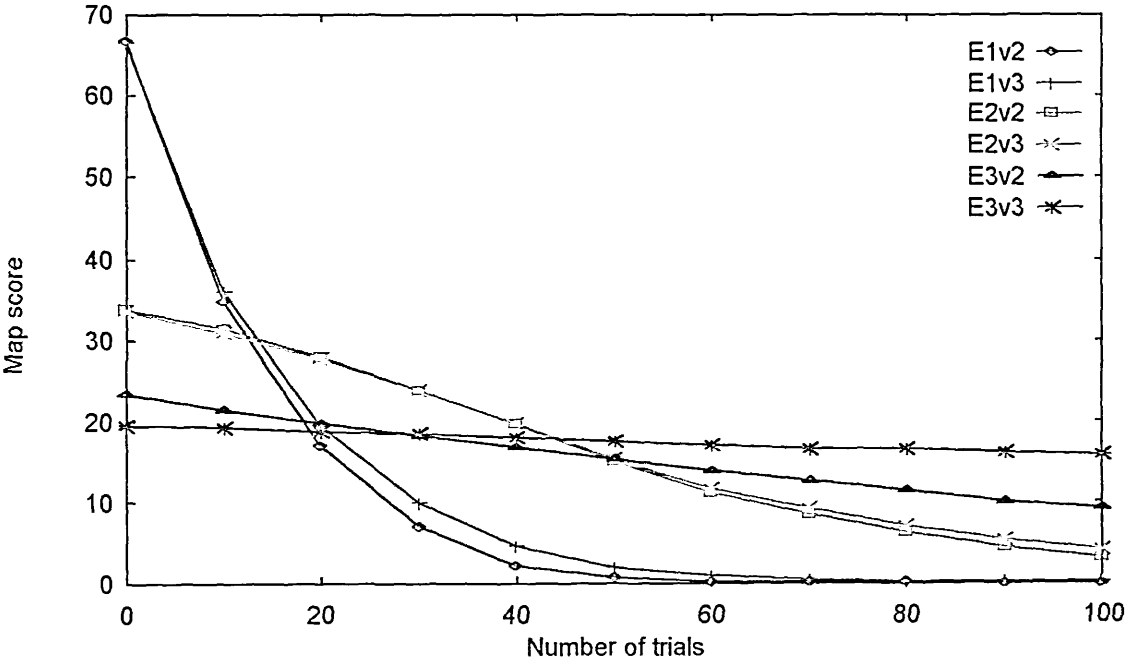


Figure 7.8 The effect of disallowing duplicate templates between HGA v2 and v3 on the average map score for PI usingevaluation 1(E1), evaluation 2(E2) and evaluation 3(E3). (SSWDT reproduction technique used; operators applied 30% each; and population size = 100.)

Using evaluation 1, all the maps in the population became acceptable maps by trial 70. There was a significant difference in the number of acceptable maps generated up to trial 70 between the two versions of the HGA. Disallowing identical templates had a significant effect on the number of acceptable maps generated using evaluation 2. Using evaluation 3 alone, there was a significant drop in the number of acceptable maps at the end of the run. It was conjectured that this was the result of the combined effect of revising evaluation 3 to penalise maps with a zero interval over the probe, and disallowing identical templates.

2.4 Varying parameter settings

The rate of applying the three operators had been set at 30%. The effect of varying the operator settings on the number of acceptable maps and the average map score for AACT and PI was examined. Case, order and side swap were applied at the following rates for AACT using all three evaluations and HGAv1: experiment A - 20% each; experiment B - 30% 30% 60%; and experiment C - 100% each. For PI, HGAv3 was used with all three evaluations and case, order and side swap were applied at the following rates: experiment A - 30% each; experiment B - 50% 50% 0%; and experiment C - 50% 50% 25%.

For both genes, there was a significant difference in performance between the three settings (settings from experiment C were best for AACT and from experiment B for PI) however, the difference was not enough to have a practical effect on the number of acceptable maps. (All other parameters were set as for section 2.3.)

The effect of extending the number of trials over which the HGA ran on the average map score and on the average number of acceptable maps was examined and the results are shown in figures 7.9 and 7.10 for PIL using HGAv3. The operator rates were set at 30% each. Reducing the population size to fifty significantly improved the map score although this did not have much of a practical effect on the number of acceptable maps.

2.5 Random replacement

The effect of using fitness proportionate selection coupled with the problem specific operators in the HGA was compared with a random replacement scheme and the results are shown for PI in figure 7.11. (AACT and PIL displayed a similiar trend.) HGAv3 was modified so that new maps were produced by the gene map builder and not by selecting parents and applying the operators. As before, the worst maps in the population were replaced at each generation. The fitness of the population, as indicated by the map score, for each of the three genes was significantly poorer using the random replacement scheme.

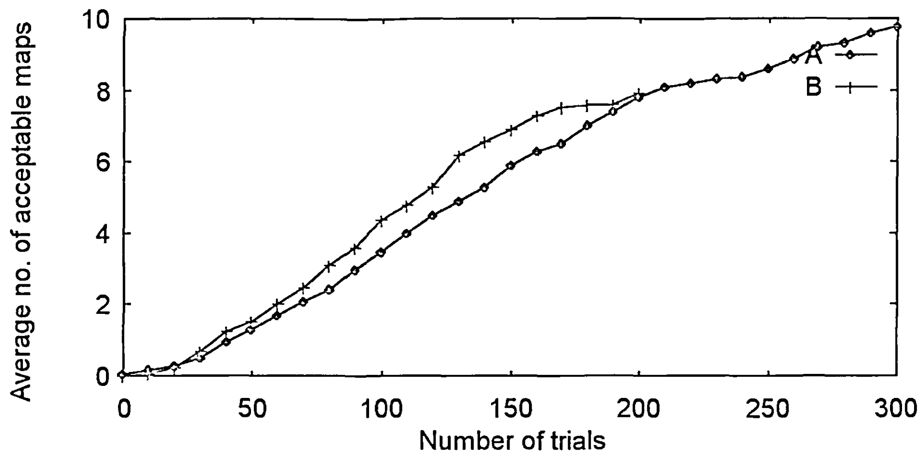


Figure 7.9 The number of acceptable maps for PIL using evaluations 1,2 and 3, HGAv3 and a population size of 100 (A) and 50 (B). (Steady-State-Without-Duplicate-Templates strategy used; and operators applied 30% each.)

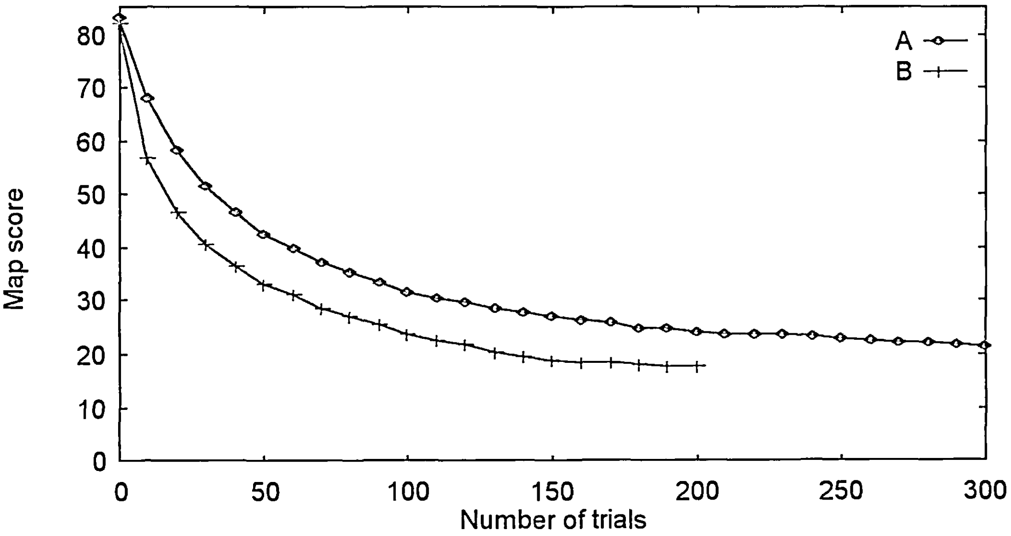


Figure 7.10 The change in average map score for PIL using evaluations 1,2 and 3, HGAv3 and a population size of 100 (A) and 50 (B). (Steady-State-Without-Duplicate-Templates stratgey used; and operators applied 30% each.)

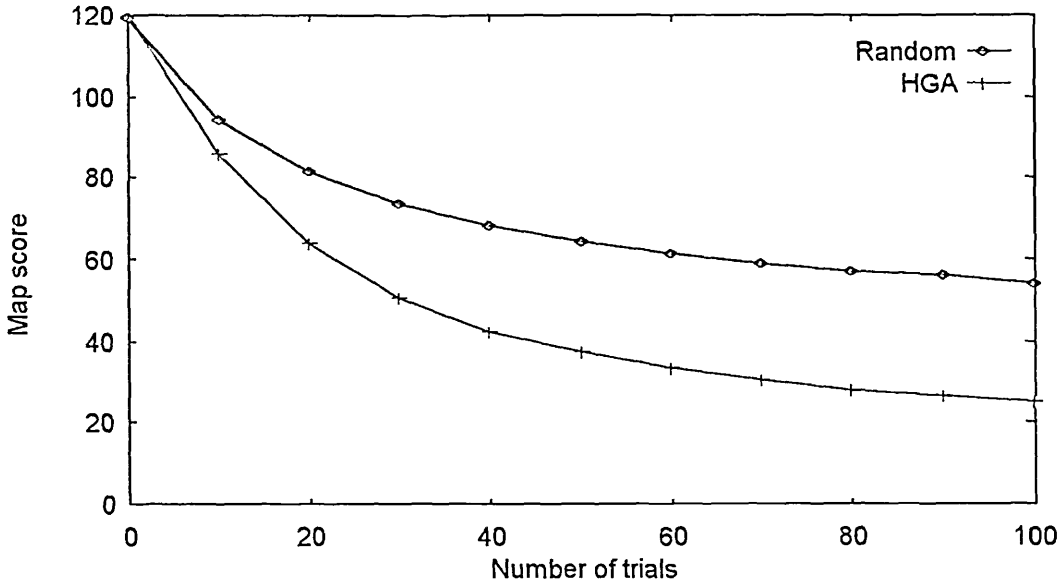
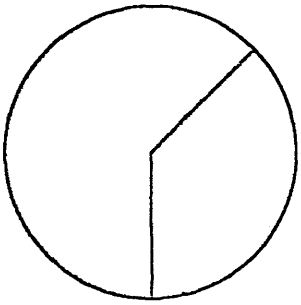


Figure 7.11 The effect of replacing maps at random compared to using fitness proportionate parent selection and the special operators (as in HGAv3) on the map score for PI using evaluations 1,2 and 3.

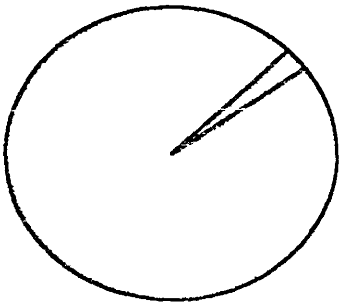
2.6 HGA success rate

The average number of acceptable maps generated at the end of 100 trials using HGAv3 and applying all three evaluations together was found to be low for PIL(4.2 maps), PI(3 maps) and AACT(4.85 maps). The results represented the average of 20 runs. When the cumulative number of ideal maps alone generated for the 20 runs was examined, the HGA appeared to have been reasonably successful in finding a high proportion of ideal templates for each gene from examining a small fraction of the total search space. The number of map evaluations carried out at each trial depended on the reproduction technique employed. For example, when Generational Replacement with Elitism (GRE) was used initially 100 maps were produced and evaluated at each trial.

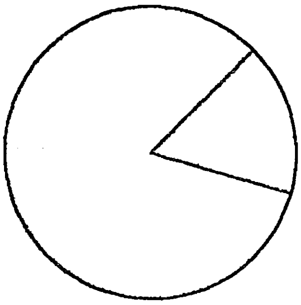
A disadvantage of GRE was the amount of time spent evaluating a completely new population at each generation coupled with the probability that time was being spent evaluating identical maps as there were likely to be a high number of identical maps produced at each generation. For a population size of 100 and 100 trials, 100,000 new maps were generated and evaluated. This was in contrast to a steady state reproduction technique when a small number of maps were produced and evaluated at each generation. Using a population size of 100 and 100 trials, replacing 2 maps at each generation, 298 new maps were produced and evaluated. Each experiment of 20 runs conducted would have produced and evaluated 5960 maps (([100 maps in the initial population + (2 new maps each generation x 99 generations)]x20).



A - PIL 37.6 % of the PIL search space was examined in total by the HGA over 20 runs. 66% of the ideal templates were found.



B - PI 2.6% of the PI search space was examined in total by the HGA over 20 runs. 60% of the ideal templates were found.



C- AACT 17.2% of the AACT search space was examined in total by the HGA over 20 runs. 41% of the ideal templates were found.

Figure 7.12 The fraction of the search space sampled by the HGA for PIL (A), PI (B) and AACT (C) and the percentage of ideal templates discovered.

Using a steady-state reproduction technique, 37.6% of all possible maps for PIL were examined and 66% of all the ideal templates were found. 2.6% of all the possible maps for PI were examined and 60% of all the ideal templates were found. 17.2% of all the possible maps for AACT were examined and 41% of all the ideal templates were found. These figures are shown pictorially in figure 7.12.

2.7 HGA dynamics

The behaviour of the HGA was examined in terms of the number of acceptable maps generated over the 20 runs and when during the run the acceptable maps were produced. (Operators applied 30% each; population size = 100; and a steady-state-without-duplicate-templates strategy, replacing 2 maps at each trial.)

The acceptable maps produced at the end of selected runs were examined and it was found that the maps were very close in terms of the operations required to go from one acceptable map to another. When the whole search space for AACT was examined (as described in chapter 5, section 3), the number of acceptable maps was split almost evenly between the four map options. It was conjectured that if all map options could be preserved during a run, there would be a greater number of acceptable maps generated than from those runs which lost map options. The results obtained from this exercise were inconclusive. It was conjectured that the results from a larger number of runs would need to be examined.

The point in the run at which the first acceptable map was produced was investigated. It was conjectured that the presence of acceptable maps either in the initial population or early on in the run would give rise to an increased number of acceptable maps for that population. It was found that the majority of acceptable maps were produced prior to generation 50.

3 HGA results using data set 2 - perfect PIL/PI/AACT data

The exact data that would be expected from the Sefton revised map was calculated for PIL/PI/AACT. The perfect data set contained no errors. All fragment lengths were accurate and all the expected fragments were present. The perfect data set is shown in Appendix A. The number of acceptable maps for PPIL, PPI and PAACT using all evaluations together are shown in figure 7.13 and the change in the average map score is shown in figure 7.14. Although the number of acceptable maps at the end of 100 trials was low for each of the genes, the fitness of the population, as indicated by the map score, improved most noticeably for PAACT followed by PPI then PPIL.

4 HGA results using data set 3 - AT/ACE

The AT/ACE data set, shown in appendix A, was used with HGAv3. The number of acceptable maps for AT and ACE using all evaluations together are shown in figure 7.15 and the change in the average map score is shown in figure 7.16. The number of acceptable maps at the end of the 100 trials was low but the map score had improved substantially over the course of the run.

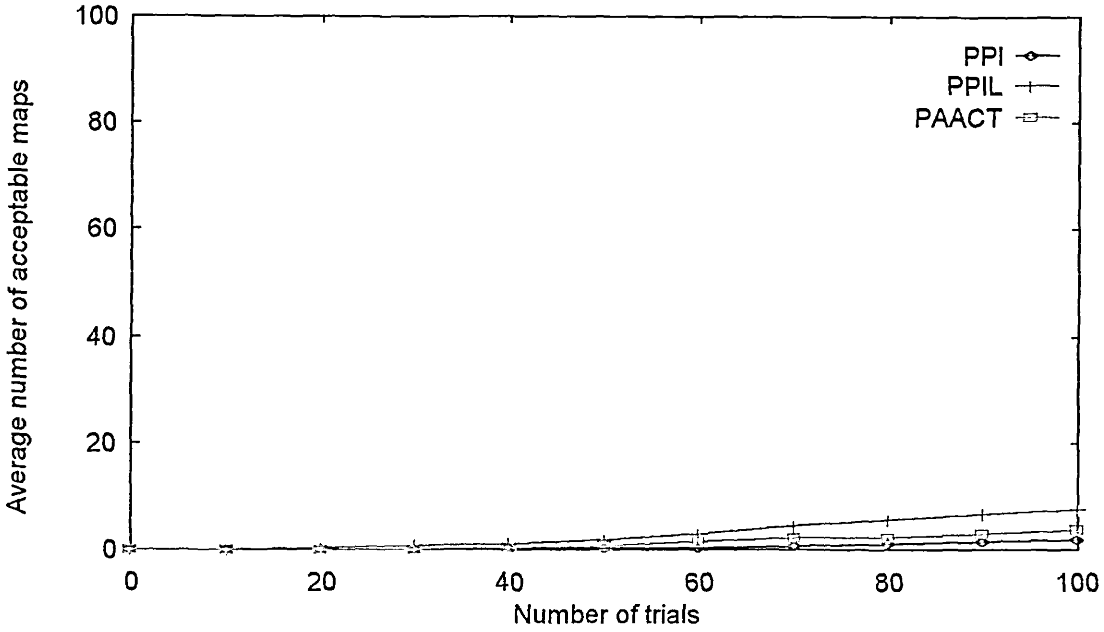


Figure 7.13 The number of acceptable maps for PPI,PPIL,PAACT using evaluations 1,2 and 3, HGAv3. (Steady-State-Without-Duplicate-Templates technique used replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

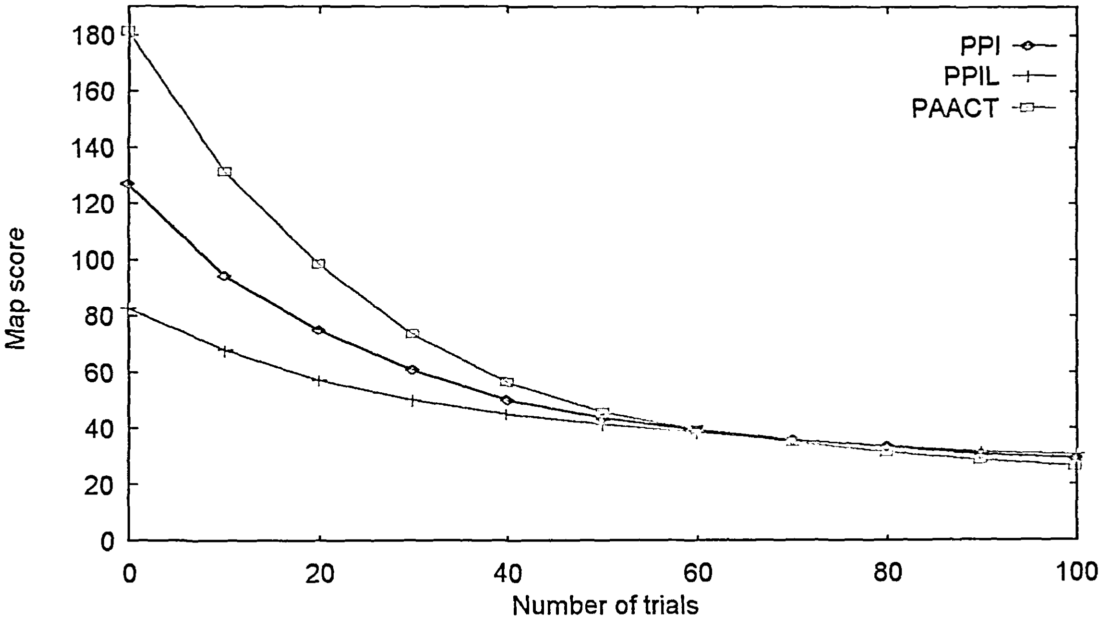


Figure 7.14 The change in average map score for PPIL, PPI and PAACT using evaluations 1,2 and 3, HGAv3. (Steady-State-Without-Duplicate-Templates technique used replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

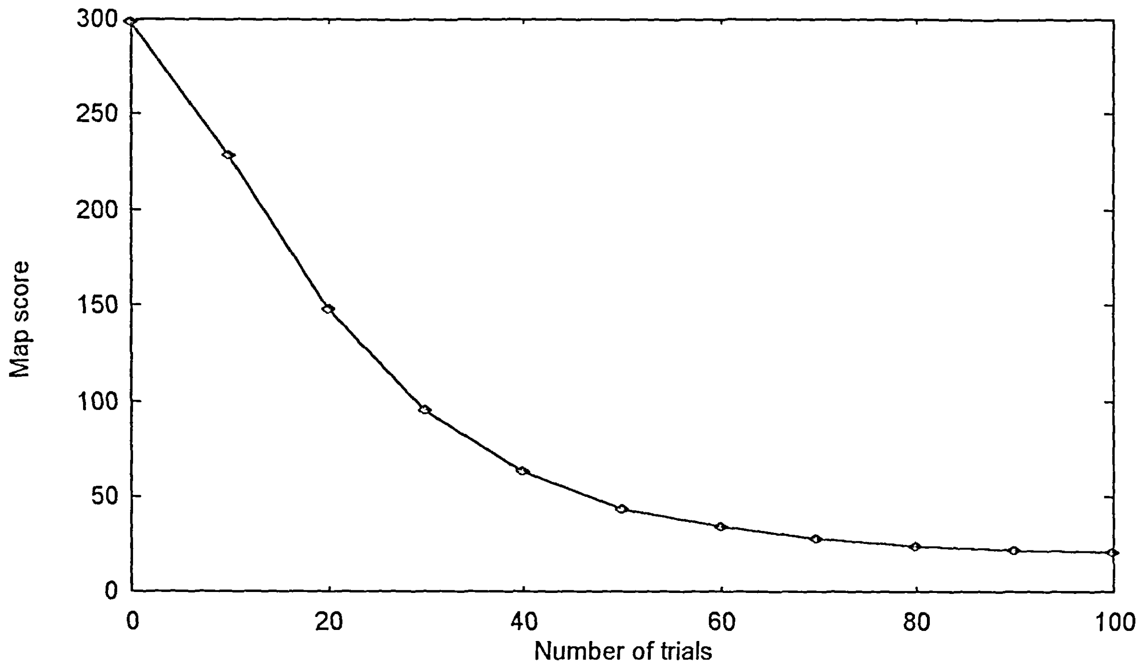


Figure 7.15 The change in map score for AT using evaluations 1,2 and 3, HGAv3. (Steady-State-Without-Duplicate-Templates strategy used, replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

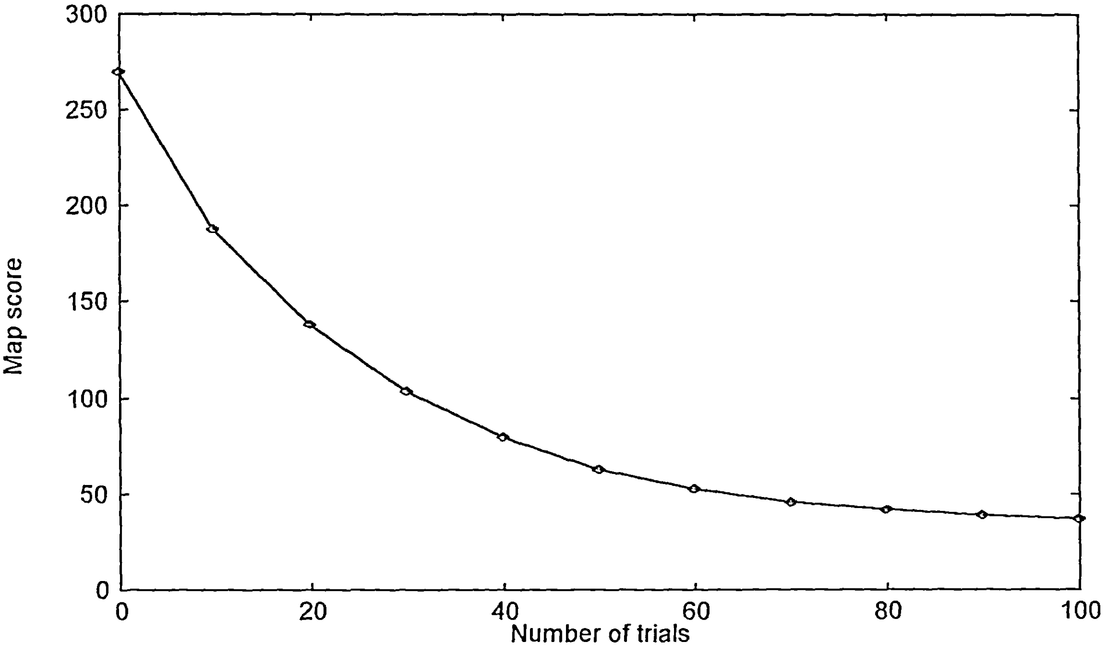


Figure 7.16 The change in map score for ACE using evaluations 1,2 and 3, HGAv3. (Steady-State-Without-Duplicate-Templates strategy used, replacing 2 maps at each trial; operators applied 30% each; and population size = 100.)

5 Discussion of results

Various modifications were made to the parameter settings in the HGA to optimise performance. The effect of changing the population size, the number of trials, the operator settings and the reproduction technique on performance was investigated. The effect of using different combinations of evaluations in the evaluation mechanism was examined. The HGA was applied to three different data sets: the PIL/PI/AACT data set as contained in Sefton et al(1990); the perfect PIL/PI/AACT data (the data calculated by the author that should have been observed given the revised published map); and the AT/ACE data as contained in Sefton et al(1990). Four main versions of the HGA were used (HGAv1, HGAv2, HGAv3, HGAv4) for the experiments reported here (each version is summarised in appendix H).

The effect of the different evaluations on the number of acceptable maps for AACT was examined, using HGAv1. A steady state reproduction technique replacing two maps at each trial was used. As can be seen from figure 7.1, using evaluation 1 alone, after fifty trials, 80% of all the maps in the population would be considered acceptable based on this evaluation alone. This was as expected because evaluation 1 was a broad evaluation as illustrated in chapter 5, table 5.2. Using the figures for AACT in chapter 5 table 5.2, it was calculated that approximately 19% of all the maps possible for AACT would be considered acceptable based on evaluation 1 alone. Using evaluations 1 and 2 together resulted in fewer maps being considered as acceptable maps at the end of the run. Evaluation 3 resulted in 40% of the population being considered as acceptable maps and evaluation 2 resulted in 28% of the population being considered as acceptable maps. Applying evaluations 1 and 3 together gave a similiar result to applying evaluation 3 alone. This is what would be expected because a map which was acceptable according to evaluation 3 must be acceptable according to evaluation 1. (Evaluation 1 measured how well the single digest data fitted and evaluation 3 took account of this also.) Applying all evaluations together gave a similiar result to applying evaluations 2 and 3 together. The change in average map score showed that the fitness of the population improved by 64.5% for AACT, 49.9% for PIL and 70.6% for PI over the 50 trials. The average number of acceptable maps using all three evaluations over the 50 trials was low (AACT(1.25), PIL(1.65%) and PI(2.7%)). The number of maps in the population was set to 100 to ensure that there was a good mix of map options in the initial population. (Both AACT and PIL had four map options each and PI eight.)

The effect of modifying the reproduction technique on the number of acceptable maps and the map score was examined. Generational Replacement with Elitism (GRE) was used in several experiments and in the runs reported in Walker et al(1994) and the results of varying the population size on the number of acceptable maps and map score for AACT was investigated. Although the number of acceptable maps in the population appeared high compared to the results found using a Steady-State(SS) strategy for AACT(between approximately 35% to 56% of acceptable maps compared to approximately 1% acceptable maps) the effect of duplicate maps was not taken into account. It was postulated that if duplicate maps were disallowed, the diversity in the population would be increased and that this would improve the performance of the HGA. Disallowing duplicates did result in a significant increase in the number of unique acceptable maps for AACT and PI as reported in section 2.2. (It was not possible to determine whether or not the increase for PIL was significant using the method described in chapter 6 section 11 as a simple transformation of the results into a straight line could not be found.)

The number of trials conducted was originally set to 50 as this had seemed adequate to allow the population to converge when using evaluation 1. To allow comparisons between the results of different experiments it was maintained initially at 50. Generally, GA practice suggests that a GA should be allowed to run ideally until the population converges or alternatively for a set period of time. For these reasons, it seemed appropriate to increase the number of trials. When the number of trials was extended to 100 (using HGAv2), the number of acceptable maps generated by the HGA using each evaluation separately and all together increased as can be seen in figure 7.3 for PIL and 7.4 for PI and the map score improved significantly as shown in figure 7.6. The rate of increase of the map score was slowest when all three evaluations were applied together.

The number of acceptable maps for PIL and PI (figures 7.3 and 7.4) using evaluation 3 alone was higher than expected as it was not considered possible that the observed data could be fitted well into so many different maps. The acceptable maps were analysed in more detail and it was found that many of them contained a zero interval over the gene position. Although correct as far as determining the intervals was concerned, such a situation gave rise to maps which were illegal. A zero interval across a gene altered the number of cut sites on either side of the map which in turn, altered the fragments that would be expected. In addition, many maps were found which had the same number of cut sites in the same order but differed in whether or not the cut sites were partial or complete. It appeared that a considerable amount of search effort had focused on exploiting similiar maps. It was conjectured that an improvement in performance might be gained by developing the reproduction technique one step further. In addition to disallowing duplicate maps, duplicate templates would be disallowed too. The SSWD reproduction technique became a steady-state-without-duplicate-templates (SSWDT) reproduction technique in HGAv3 and evaluation 3 was upgraded to ensure that a zero interval across the gene would result in a poor score. The effect of these modifications on the number of acceptable maps for PI was shown in figure 7.7 and on the map score in figure 7.8. The effect of disallowing duplicate maps alone is seen for evaluation 1 and evaluation 2 and the combined effect of disallowing duplicate maps and a zero interval across the gene was seen for evaluation 3. Disallowing duplicate templates using evaluation 1 slowed down the rate at which the average number of acceptable maps rose but by generation 70, all the maps scored as acceptable maps using HGAv2 and HGAv3. A similiar trend was visible in the number of acceptable maps using evaluation 2. There was little difference in the map score. There was a marked difference in the performance of HGAv2 and HGAv3 using evaluation 3. From examining the acceptable maps produced by HGAv2 and HGAv3, the difference in the number of acceptable maps was found to be largely due to penalising maps with a zero interval across the gene. The average map score using HGAv3 was worse at the end of the 100 trials than using HGAv2. The effect on the number of acceptable maps was quite pronounced - at the end of 100 trials there were only twenty-one acceptable maps compared to eighty previously.

Using HGAv3, the SSWDT reproduction technique and applying all three evaluations together, the effect of adjusting the operator settings was examined. Although the alternative operator settings produced significantly different average map scores, there was not a practical difference in the number of acceptable maps.

The effect of increasing the number of trials and population size on the HGA performance was examined and the results are shown for PIL in figures 7.9 and 7.10. The best performance (in terms of the average map score) was obtained when the population size was decreased to 50. Here the average map score at trial 200 was 20% better with the smaller population size. Nevertheless, the map score had not improved enough to have an impact on the number of acceptable maps which remained the same between both runs.

To examine the effect that selecting parents according to fitness and applying the problem specific operators was having on performance, the HGAv3 was modified so that new maps were generated at random at each generation. All other parameters were kept the same. The results for PI were shown in figure 7.11. (AACT and PIL displayed a similiar trend.) The average map score using the parent selection technique and operators was more than 50% better than when a random map replacement technique was used.

An exhaustive search of the search space was conducted for the PIL/PI/AACT data set to find out how many "ideal", "good" and "adequate" templates were actually present using each of the evaluations separately and together. The results of this exercise were shown in chapter 5, section 3. As discussed in section 2.6, the number of acceptable templates found by the HGAv3 using all three evaluations was low. If the number of ideal templates alone which were accumulated over the 20 runs was examined, the HGA appeared to be reasonably successful in finding a high proportion of ideal templates from examining only a fraction of the total search space as shown in figure 7.12. From examining the acceptable maps found at the end of individual runs, it was found that the acceptable maps were very close to one another in terms of the steps required to transform one acceptable map to another. For example, one application of the order swap operator to an acceptable map could be sufficient to transform the map into another acceptable map. In some runs, no acceptable maps at all were generated; however, if an acceptable map was generated, it was quite likely that due to the "closeness" of other acceptable maps, that more would be generated, providing the operators were applied to the better maps.

The results of running the HGAv3 using perfect data for PIL, PI and AACT were shown in figures 7.13 and 7.14. The difference in the average map score for the initial populations was due to differences in the number of fragments expected for each gene; however the average map score for all three genes settled at around 25 to 30 at the end of 100 trials. The results of running the HGAv3 with the AT, ACE data were shown in figures 7.15 and 7.16. In both cases, the average map score was shown to improve substantially over the run, but not enough to have an effect on the number of acceptable maps. The average number of acceptable maps found at the end of the runs using the perfect data was less than five and for the AT/ACE data set was less than one. The perfect data set and the AT/ACE data set represent a much larger search space than the PIL/PI/AACT data set, as illustrated in chapter 2, table 2.3. In addition, both AT and ACE have many more map options than the PIL/PI/AACT data set to be maintained in the population (AT had sixteen options and ACE had eight options).

6 Summary

The results of applying the HGA to three different data sets were presented. Various modifications to the HGA were made to improve performance and the success of the HGA in generating acceptable maps was discussed. The best performance was obtained using a Steady-State-Without-Duplicate-Templates reproduction technique, replacing 2 maps at each trial. Evaluations 2 and 3 had to be applied in order to obtain correct maps. Varying the operator settings had a significant effect on the fitness of the population however, this did not have a practical effect on the number of acceptable maps. The same parameter settings used for data set 1 were used for data set 2 and data set

3. Significant improvements in the population fitness were achieved, however, the number of acceptable maps generated at the end of the runs were low. In terms of computational efficiency, it was shown that the HGA was reasonably successful in finding a high proportion of the ideal templates for data set 1 from examining a small fraction of the search space.

The next chapter considers how the acceptable maps found for individual genes can be merged together to create a complete map.

CHAPTER 8 - DEVELOPING A MECHANISM FOR MERGING MAPS

1 Introduction

Acceptable single gene maps showing the sequence of restriction enzyme(RE) cut sites around a single gene were generated by a hybrid genetic algorithm(HGA) from experimental data. The technique for aligning and merging the single gene maps together to create a new sequence indicating the sequence of all the genes and RE cut sites is described. The results of applying the merge facility are presented and discussed. Some suggestions for enhancing the basic mechanism are made.

2 The Merge Operation

Essential to merging the single gene maps correctly was the occurrence of at least one RE cut site whose position was common to the two maps. This cut site was necessary to enable maps to be aligned. Determining which cut sites were common to both maps relied on the presence of long fragments in the data set that contained two or more genes. Such fragments could be identified by their length. If fragments for different genes, produced by the same RE, had the same length, it was likely they were the same fragment containing both genes. This is illustrated by example in figure 8.1.

Having aligned two maps on a common cut site, the distances between the cut sites in the individual maps could be used to produce a merged map. The merged map would show the sequence of the genes and all RE cut sites.

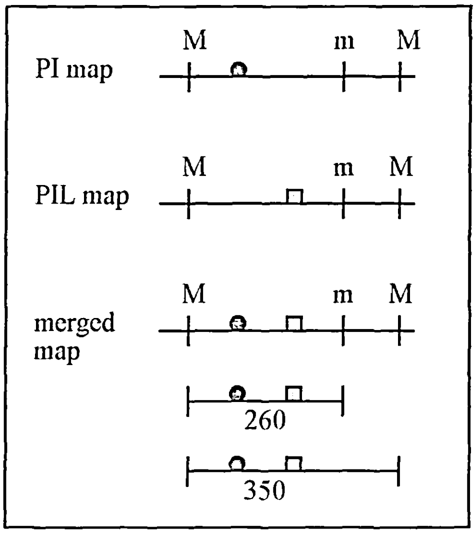
B			M			S			F		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	355	350	350	350	255	255	70	165	(190)	(195)
225	225	275	260	260	80				(10)	(135)	(175)
	(65)	230		(180)						80	135
										65	
										10	

B/M			B/S			B/F			M/S		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	230	125	125	70	165	(190)	135	255	255	70
225	225	105		(65)		(10)	50	105		(180)	30
130	130						10				
	(65)										

M/F			S/F		
PIL	PI	AACT	PIL	PI	AACT
85	80	135	70	(130)	30
(10)	65	85	(10)	80	
	10			65	
				10	

Figure 8.1 - Common cut sites.

The table of experimental data from Sefton et al(1990) is shown. Where two fragments for different genes had the same lengths when cut by the same restriction enzyme, it was likely that they were the same fragment, as shown below for **M**. The cut sites at either end of the fragment would be common to both gene maps.



2.1 Description

Crucial to aligning the single gene maps was the identification of the common cut site(s). Possible common cut sites could be identified automatically by considering fragment lengths. However, it was proposed that it would be of benefit to involve the geneticist at this stage. It was quick and relatively easy for the geneticist to specify which of the fragments in a data set were common to two or more genes, from examining fragment lengths and by visual examination of the DNA bands on the gel columns.

A merge facility was developed (written in Microsoft C, running on an IBM compatible PC) which took as input single gene maps and the experimental data from which they were generated. The geneticist was prompted to name a common cut site between the gene maps. The merge facility aligned the gene maps on the common cut site and produced a merged map that showed the sequence of both genes and all the RE cut sites; the distances between them; and the length of the map. Extra single gene maps could be merged into the multi-gene map as necessary. The multi-gene maps were assessed using two of the criteria used by the geneticist - map length and number of cut sites. The geneticist considered that a good multi-gene map was one which had a short length and small number of cut sites: both features implied that the gene maps had overlapped. If there was a high degree of overlap between two maps, the merged map would be short and would contain a minimal number of cut sites. These criteria were used to evaluate the merged maps.

3 Results of Merging

The merge facility was tested using: the perfect gene maps - to establish whether or not the perfect multi-gene map could be generated; and the revised published gene maps - to establish whether or not the revised published multi-gene map could be generated. The first case tested the operation of merge using ideal maps. The second case tested merge on less than ideal maps. Merge was then applied to merge the acceptable gene maps, generated by the HGA, together to produce complete maps.

3.1 Merging the perfect PIL/PI/AACT gene maps

The perfect PI and PIL maps (shown in figure 8.2) were merged together using M as the common cut site and the exact PIL/PI merged map was obtained. In the merged PIL/PI map, the number of cut sites and their distances from one another were accurate. The AACT map was added. The complete map (shown in figure 8.2) placed the genes in the correct sequence. The merged map was slightly shorter than the correct map (677.02 compared to 710 kb); contained one extra cut site; and some of the distances between the cut sites were inaccurate. However, in general, the sequence of cut sites between the merged map and the correct map was very similar.

The reason why the merged map differed from the correct map was that the merge facility had calculated a shorter map for AACT than the correct map. As the map was shorter it was considered better than the correct map by the merge facility. Both maps are shown in figure 8.2.

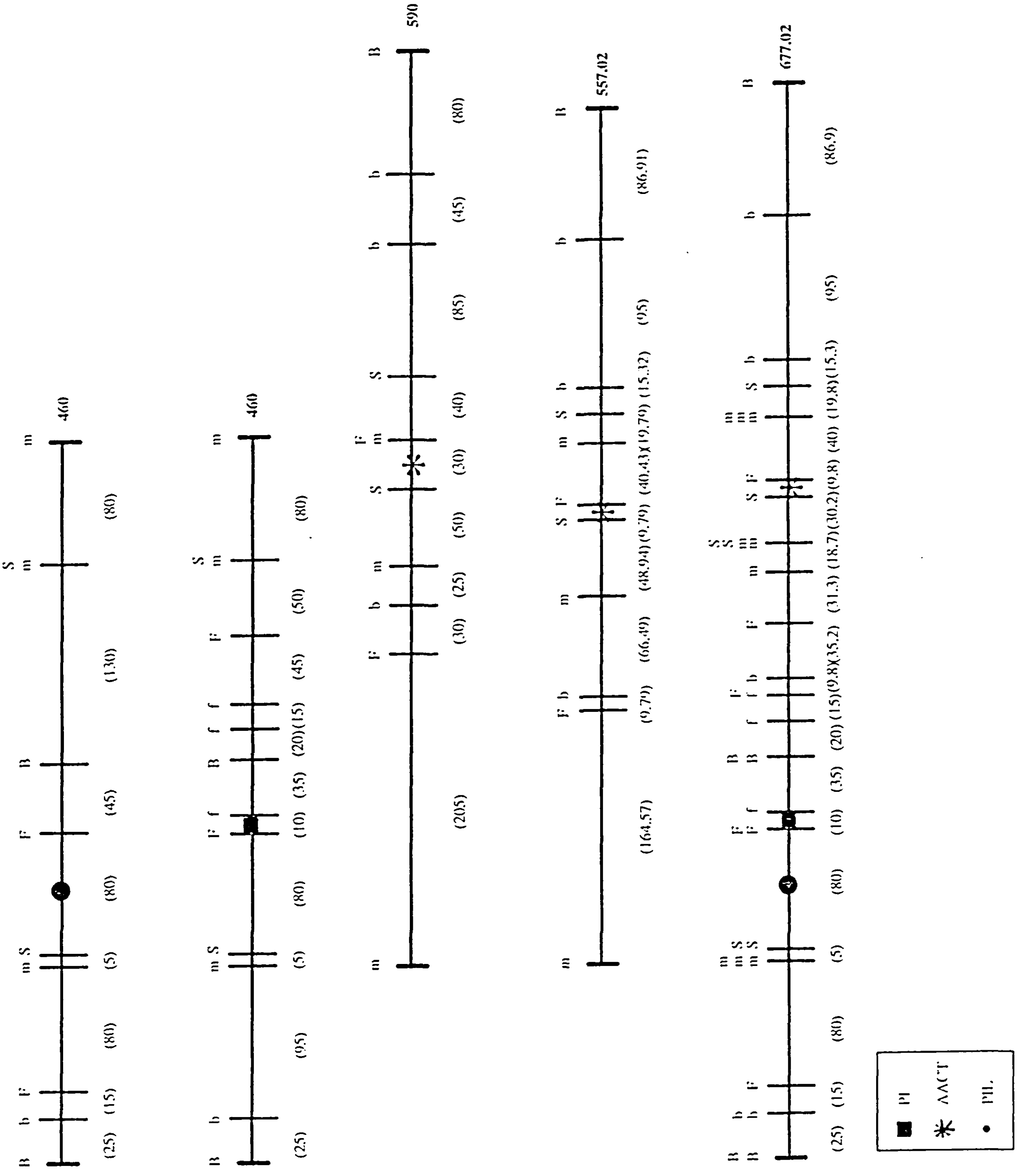


Figure 8.2 - Merging the perfect PIL,PI,AACT maps.

The PIL and PI single gene maps were merged using M as the common cut site. Two maps for AACT are shown. The AACT map used by the merge facility was the shorter map at length 557.02 kb. Again, M was used as the common cut site and the AACT map was merged with the PIL/PI map to create the complete merged map at length 677.02 kb. Although the merge facility chose the shorter AACT map in preference to the correct longer map at 590 kb, the sequence of genes and cut sites is still very similiar to the correct multi-gene map (shown in chapter 4, figure 4.2.)

The "better" map was considered to be the shorter one at 557.02 kb long which was chosen over the correct, longer one at 590 kb long. In the longer map, the data fitted perfectly in contrast to the shorter map which had more error.

3.2 Merging the revised published gene maps

The revised published gene maps for PIL/PI/AACT were merged using M as the common cut site. Four merged maps were obtained and are shown in figure 8.3. (There were three common cut sites between PIL and PI - it did not appear to make much difference which of these was used to align the maps as shown in figure 8.4.)

The merged maps in figure 8.3 were similar to the revised published map (shown in chapter 4, figure 4.2), apart from the occurrence of some additional cut sites. In most cases, when a cut site was separated from an identical type of cut site by a distance <10kb long, the cut sites should have overlapped. When identical cut sites were taken into account, the correct number of cut sites and their ordering were obtained. (The intervals between the cut sites varied slightly between the maps and was a result of the way the merge algorithm processed distances.)

The revised published gene maps for AT and ACE were merged using M as the common cut site. Apart from being slightly longer (791.15 compared to 735 kbp), and containing some duplicate cut sites, the merged map and revised published maps were very similar.

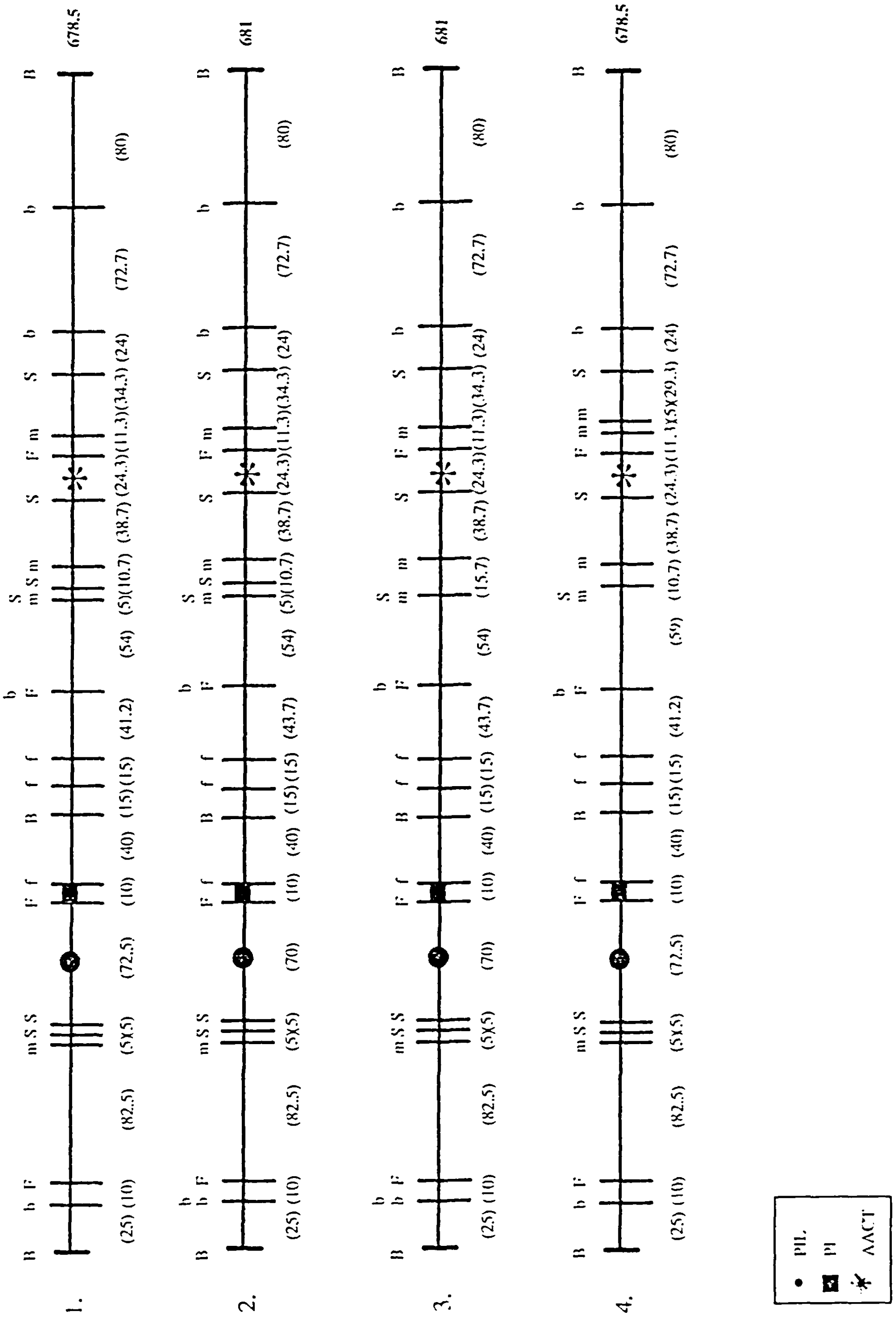


Figure 8.3 - Merging the revised published PIL,PI,AACT maps.

The PIL and PI single gene maps were merged using M as the common cut site. The AACT map was merged with the PIL/PI map using M as the common cut site to create the complete merged map. The four maps produced by the merge facility were all very similiar and were very close to the revised published map (shown in chapter 4, figure 4.2).

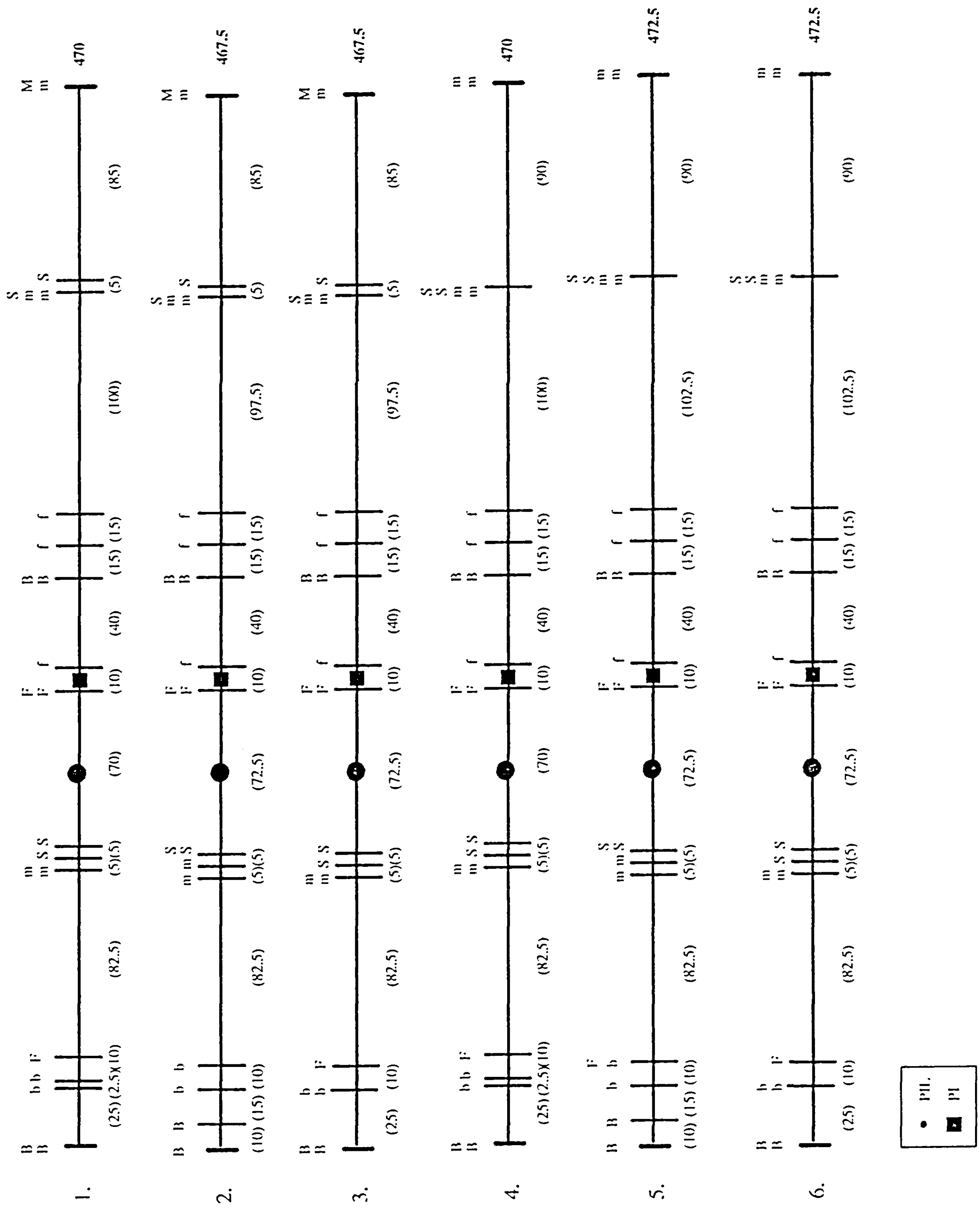


Figure 8.4 - The effect of merging maps using different common cut sites.

There were three common cut sites between PIL and PI - M, B and S. It did not appear to make much difference which of the three were used. Maps 1 and 2 were merged on M; maps 3 and 4 were merged on B; and maps 5 and 6 were merged on S.

3.3 Merging the HGA generated maps for PIL/PI/AACT (Sefton data)

Ideal templates generated by the HGA using the data from Sefton et al(1990) for PIL, PI and AACT (from one run) were merged together using M as the common cut site. 82 multi-gene maps were produced, ranging in length from 401.67 to 823.1 kb.

The criterion of shortest length and least cut sites were used to identify the best merged maps. There were 8 maps that had the least number of cut sites(25) and they are shown in figure 8.5 (maps 2-9). Although there were duplicate cut sites present, the majority of maps had the genes in the correct sequence. All the maps found were shorter than the revised published map. Allowing for duplicate cut sites, map 4 reduced to 18 cut sites. This map appeared to be better than the revised published map - it was 30% shorter and contained 3 less cut sites. However, if the fit of all the data in the complete map were considered, problems with the AACT fragments were highlighted.

The shortest length map at 401.67 kb had 27 cut sites and is shown in figure 8.5 (map 1). This map was interesting as the order of the genes was PI/PIL/AACT which disagreed with the ordering arrived at by Sefton et al(1990) and agreed with the ordering proposed originally by Cox et al(1987). Again, problems arose when the fit of all the data in the complete map was checked.

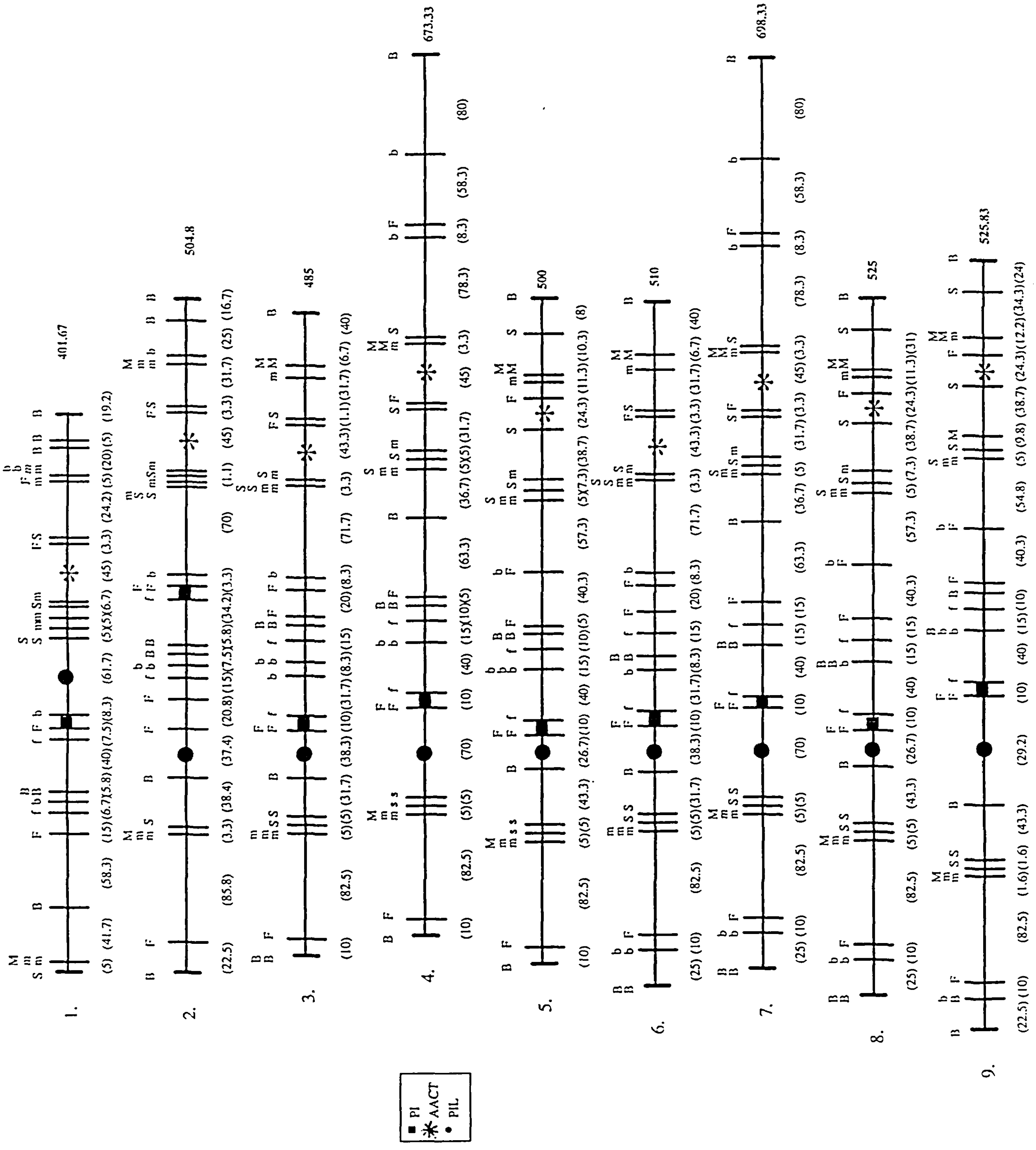


Figure 8.5 - Merging the HGA generated PIL,PI,AACT maps (Sefton data).

The ideal templates found for PIL,PI and AACT were merged using M as the common cut site. The shortest maps and those with the least cut sites are shown. The shortest map had the genes in the wrong sequence, but the eight maps with the least cut sites had the genes in the correct sequence and the sequence of cut sites was similiar to the revised published map (shown in chapter 4, figure 4.2).

3.4 Merging the HGA generated maps for PIL/PI/AACT (perfect data).

The ideal templates generated for PIL and PI by the HGA using the perfect data were merged together using M as the common cut site. Seventy merged maps were produced, the shortest of which was 380 kb in length which also had the fewest cut sites at 15. All the distances between the cut sites were identical to the distances between the cut sites in the perfect merged map except for one M cut site. This M cut site had been folded inside the map rather than being out on a limb on the right-hand side. The correct map was 80 kb longer at a total length of 460 kb, with 15 cut sites. When the merged maps were examined, five the same length as the correct map and four (maps 2 to 5) had the genes in the correct sequence and the number and sequence of cut sites were very close to the correct map.

An attempt was made to merge the PIL/PI maps with the ideal templates generated by the HGA for AACT using M as the common cut site. The merge facility disallowed the merge on the basis that the M cut sites could not be considered common to the two maps. This was due to differences in the length of the M fragments between the AACT ideal maps and the PIL/PI merged maps. In the AACT ideal maps, although all the data had been fitted, the size of the average residual was high which resulted in the distance between the M digests being outwith the limits allowed by the merge facility. This was a problem that could be solved by taking into account the size of the average residual when generating a score for a map from evaluation 3.

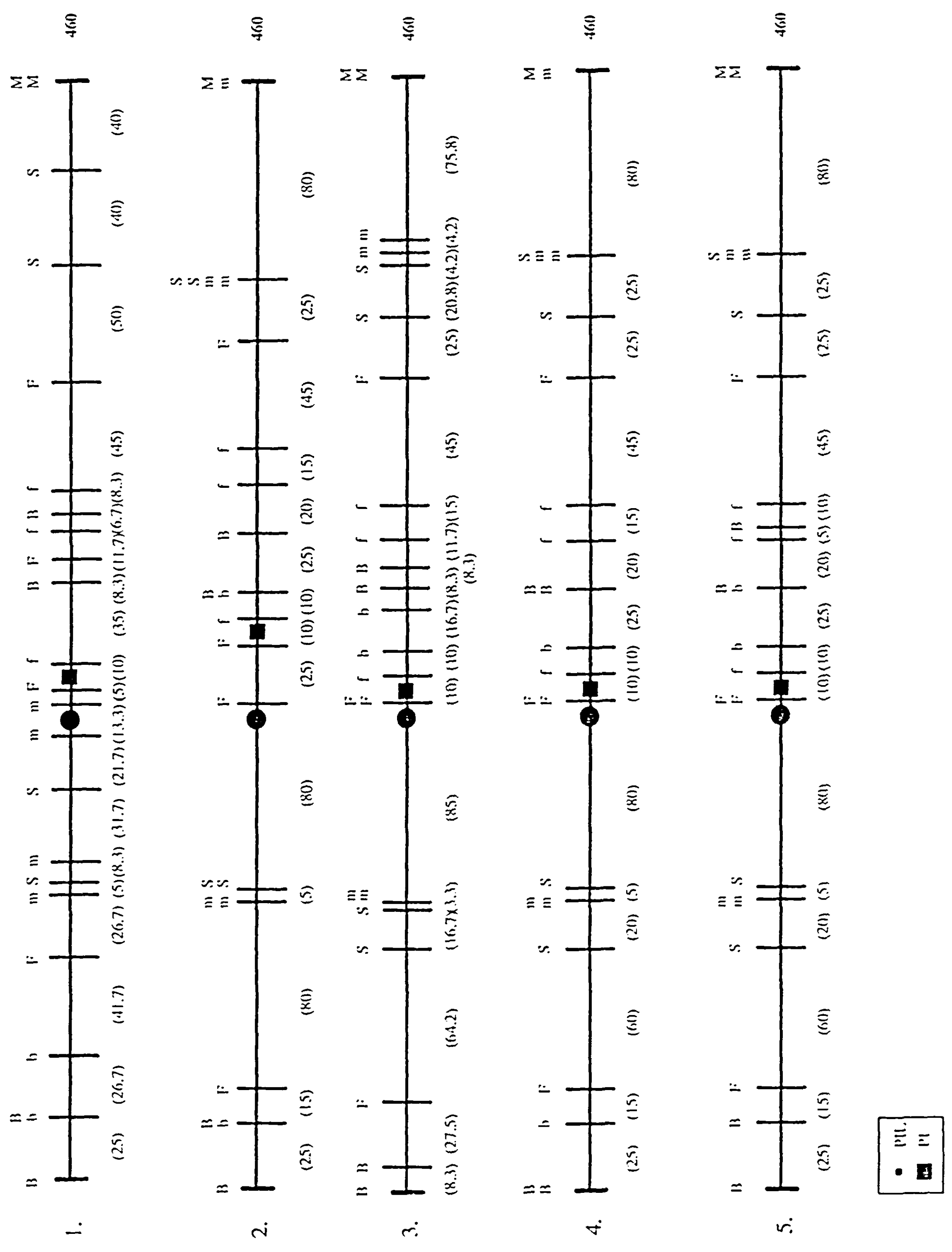


Figure 8.6 - Merging the HGA generated PIL/PI maps (perfect data).

The ideal templates for PIL and PI were merged together using M as the common cut site. Four of the five maps that had the same length as the correct map (460 kb), had a similiar sequence as shown.

3.5 Merging the HGA generated AT/ACE gene maps

Acceptable templates generated by the HGA for ACE and AT were merged together using M as the common cut site. Six maps were produced and are shown in figure 8.7. They varied in length from 740 to 766 kb and had around the same number of cut sites. There were more cut sites contained in the AT/ACE maps than the PIL,PI,AACT maps and the merge operation resulted in many duplicate cut sites. These factors complicated comparisons with the correct map.

4 Discussion

The merge facility was considered to be a basic means of merging gene maps together. The operation of the merge facility was tested using as input the perfect maps for PIL,PI,AACT and the revised published maps for PIL,PI,AACT. Although there were some duplicate cut sites, in both cases, the sequence of genes and REs compared favourably with the multi-gene maps. The occurence of duplicate cut sites could be overcome by enhancing the basic facility. When identical cut sites <10kb from each other were allowed to overlap, the complete PIL,PI,AACT revised published map was obtained with the correct order and number of cut sites apart from one extra M cut site which was 16kb from another M site. It did not appear to make any difference which of the common cut sites the maps were merged on.

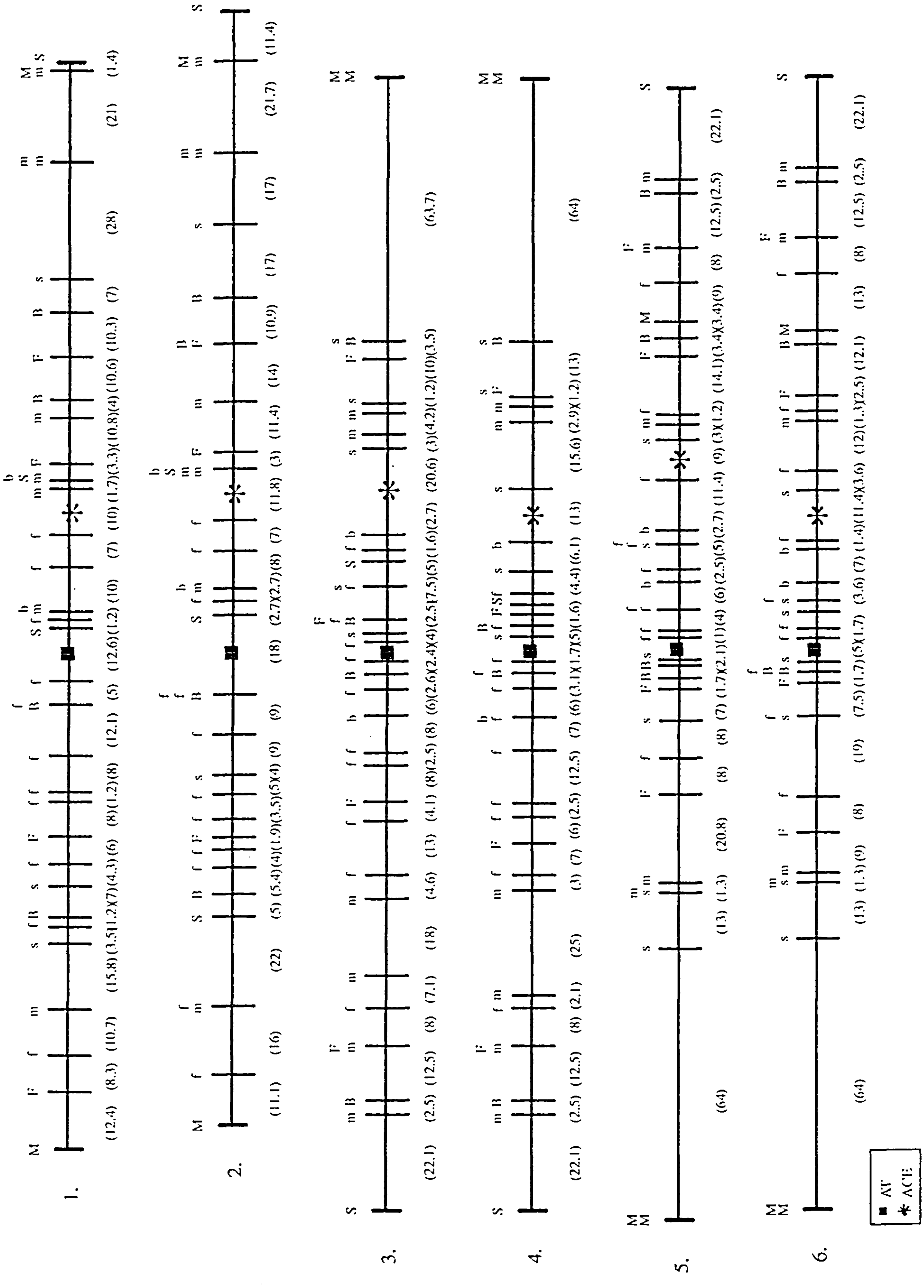


Figure 8.7 - Merging the HGA generated AT/ACE maps.

The acceptable AT and ACE maps were merged together using M as the common cut site. The merged maps with the shortest length and least cut sites are shown. These maps varied in length from 740 - 766 kb which was similiar to the revised published map at length 735 kb.

When the ideal templates produced by the HGA (generated using the Sefton et al(1990) data) were merged, the best map in terms of the shortest, was one which had the genes ordered as PI,PIL,AACT which was the ordering found by Cox et al(1987) and disagreed with Sefton et al(1990). In terms of the least number of cut sites, there were several maps which had 25 cut sites. When the overlapping of identical cut sites <10kb was allowed, a merged map was obtained which was approximately 30% smaller with 3 less cut sites than the revised published map; however analysing the map in more detail highlighted problems with the AACT fragments expected.

Even when merging small numbers of gene maps together, the quantity of options generated by the merge facility was excessive. The number of merged maps produced could be limited by incorporating a look-ahead mechanism. For example, if two cut sites were the same type and were within 9kb of each other, they could be overlapped. In general, the geneticist would not expect to be able to measure fragments < 10 kb and in all the merged maps generated, all cut sites of length < 10kb should have overlapped. This would have resulted in fewer duplicate cut sites.

The success of the merge operation relied on the quality of the single gene maps produced by the HGA. In the case of PIL,PI and AACT (using both the Sefton data and the perfect data), the HGA generated "ideal" templates. For AT and ACE, only "adequate" templates were generated and it was considered that the uncertainty in the single gene maps was compounded when the maps were merged. Various ways of increasing the quality of the maps generated by the HGA were proposed in chapter 7.

The method used to evaluate the multi-gene maps was based on the geneticist's heuristic of looking for the shortest map with the fewest cut sites. This was perhaps a useful rough guide, but was found to be insufficient as the sole means of assessing merged maps. Contrary to the heuristic described above, in several cases longer maps were found to be better than shorter maps. The heuristic suggested that a high degree of overlap between two maps was expected. In the PIL/PI/AACT map, there was a large number of common cut sites to PIL and PI because the genes were so close to one another. 75.00% of PI cut sites were shared by PIL and 83.33% of PIL cut sites were shared by PI. When the number of common cut sites with AACT was examined, it was found to be very small. Only 27.27% of AACT cut sites were shared by PIL and PI. In the AT/ACE data set, AT shared 45.00% of cut sites with ACE, and ACE shared 75.00% of cut sites with AT. The merge facility would benefit from an enhanced mechanism for assessing multi-gene maps. The mechanism developed for single gene maps could be extended to evaluate whole maps. There would be several other ways the merge facility could be enhanced - instead of aligning gene maps on a common cut site, maps could be aligned on whole fragments or on a set of common fragments as specified by the user.

5 Summary

A mechanism for merging the sequence of cut sites for individual genes together to create a new sequence containing more than one gene was developed. Single gene maps were aligned on a common cut site, as specified by the geneticist. In spite of being a basic facility, merge did have some success in generating multi-gene maps whose sequences were similar to sequences of the correct maps. Possible enhancements to the basic mechanism were suggested.

CHAPTER 9 - DISCUSSION

1 Introduction

Automating map assembly involved three main activities - developing an objective system for evaluating gene maps; developing a hybrid genetic algorithm(HGA) to generate potential gene maps from experimental data; and developing a facility for merging single gene maps to create multi-gene maps. Each activity has been presented and discussed separately in the individual chapters. Here, the general approach to automation is discussed. The success of the techniques and the implications for similiar types of problems are considered.

2 General strategy

The two main requirements for automating map assembly were: some means of objectively assessing maps; and some means of generating potential maps from experimental data. It was shown in chapter 2 that the number of possible multi-gene maps that could be generated for a set of data was very large. The strategy for tackling map assembly consisted of generating maps for individual genes, in an attempt to reduce the complexity of the problem. However, even the number of single gene maps that could be generated for a set of data was very large. A hybrid genetic algorithm(HGA) was used to generate acceptable single gene maps. A basic facility was developed to merge the single gene maps together.

3 Map evaluation mechanism

One of the key components to the success of automating map assembly was devising a way of assessing the quality of potential maps objectively. Assembling maps manually was time-consuming and complicated. It relied on the geneticist's judgement of the data and several solutions could appear feasible. Using knowledge elicitation techniques and a questionnaire, three means of assessing maps were developed. Common to any exercise involving the scoring of subjective judgement was the potential for introducing errors. The steps taken to minimise the effect of error on the design of the questionnaire and in the analysis of results were discussed in chapter 4. Developing the evaluation system was successful in highlighting critical problems with published maps that lead to the geneticist's revision of those maps. Maps were awarded a score by each evaluation and the total score of a map was used to classify the map as "ideal", "good", "adequate" or "unacceptable". Maps in any of the first three categories were all considered acceptable. The first means of evaluating a map was based on how well the single digest data fitted into a map. This was a very broad evaluation and although many maps could score as acceptable based on this evaluation alone, the maps were not correct as there were problems with the fit of the double digest data. The second evaluation was developed which awarded a map a score based on the type and yield of double digests. When applied along with the first evaluation, the number of acceptable maps was reduced, however, there were still problems with how the lengths of the double digests fitted into these maps. The third evaluation was developed to take account of how well all the observed single and double digest fragments fitted. Generally, all three evaluations were applied to identify optimal maps.

Given a set of data, a formula for calculating the number of maps possible was derived in chapter 2. The number of possibilities depended on the number of genes, restriction enzymes and fragments used in the experiment. It was shown that even in the single gene case, the number of maps possible was worse than an exponential function of the input.

To test the specificity of the evaluation mechanism, all maps possible for the PIL,PI and AACT genes from the Sefton et al(1990) data were enumerated. There were many acceptable maps for each gene. The use of templates reduced the number of ideal maps to several for each gene (a decrease from 112 to 8 for AACT; 208 to 11 for PI; and 96 to 7 for PIL). The ideal templates for a gene were quite distinct from one another in terms of the map options comprising the templates. Templates with different options were far apart from one another. Shorter templates than the revised published templates were found for each of the genes. Although these appeared to be better maps than the revised published maps, problems arose when the single gene maps were merged.

4 Techniques to generate potential maps

Rather than generating multi-gene maps from the complete set of data, single gene maps were created from a subset of the data. This strategy was chosen as it represented an approach the genticist applied to the problem.

Various methods could have been chosen to generate single gene maps. The decision was made to use a form of genetic algorithm (GA) because GAs have been reported to perform well on combinatorial optimisation problems. The result of incorporating problem specific knowledge into GAs has lead to the development of very effective search algorithms (Davis(1991a)) which have been applied to various combinatorial sequencing problems (eg Oliver et al(1987), Whitley et al(1989) (1991), Syswerda(1991)).

Having generated single gene maps, a mechanism was developed to merge the maps together to produce multi-gene maps.

4.1 The hybrid genetic algorithm

One of the challenges with modifying the traditional genetic algorithm(GA) was that the theory used to describe the operation of the algorithm no longer held. Although it is unclear what part of the schema theorem could be applied to HGAs, Whitley(1993) argued that this should not prevent various forms of GAs being developed. Existing theory could provide guiding principles but experimentation has been leading the way in this area of research.

In addition to varying the standard parameters, such as population size, number of trials, operator probabilities etc, other features of the HGA were modified in an attempt to obtain optimum performance. Deriving optimum settings for HGAs is a very time-consuming and laborious task. For these reasons, settings were varied until satisfactory results were obtained. The results were discussed in detail in chapter 7 and are summarised here.

With each of the three data sets used, the average map score at the end of a run had improved substantially from the initial population (using HGA version 3, all three evaluations applied). The improvement was most dramatic for the gene maps that had the larger search spaces. The average map score in the initial population for AT was 300 which reduced to 20 by the end of 100 generations. The improvement in average map score could be attributed to the use of the parent selection technique and the problem specific operators. This was determined by comparison with replacing maps at random. The maps in the final population were much fitter but were not fit enough to have much of a practical impact on the number of acceptable maps.

The impact of varying the number of trials and the population size depended on the evaluations used. Populations evaluated using evaluation 1 alone were quicker to converge than populations using all three evaluations. Generally, the size of the population was kept high (at 100) to ensure that there was a good mix of map options.

Although varying the operator rates made a significant difference to the average map score, it did not have much practical effect on the number of acceptable maps.

In general, the best performance was achieved using HGAv3; a Steady-State-Without-Duplicate-Templates reproduction technique, replacing two maps at each trial; and applying at least evaluations 2 and 3 to identify correct maps.

4.2 The merge facility

The merge facility took as input the best maps generated by the HGA for single genes. The single gene maps were aligned on a common cut site as specified by the geneticist, and using the distance information from both maps, a new sequence of cut sites and genes was produced that represented the complete map. The merge results were discussed in chapter 8 and are summarised here.

The merge facility was successful in generating the sequence of cut sites and genes for multi-gene maps. Although in general duplicate cut sites were produced, the sequence of genes and cut sites for PIL,PI and AACT were very similiar to the sequence in the revised published map. The success of the merge facility in producing correct multi-gene sequences relied on the quality of the single gene maps. If the merge facility was enhanced, it is more likely that acceptable maps of poorer quality (such as those found for AT and ACE) could be merged to produce reasonable multi-gene maps. A more detailed mechanism for assessing multi-gene maps would improve the effectiveness of the merge facility.

5 Success of the overall approach

The success of the overall approach was measured in terms of: the time required to generate a complete map automatically compared to the time taken by the geneticist; and the correctness of the sequence of genes and cut sites in the complete map.

Using the first criterion, automation was successful as it was considerably quicker producing maps automatically rather than manually. The time taken to generate a complete map depended on the number of genes, restriction enzymes and fragments produced. Using the data for the

PIL,PI and AACT genes from Sefton et al(1990), it took approximately an hour to perform one experiment (20 runs) on a 486 computer running at 66MHz. Merging ideal maps for the three genes took approximately two hours on a 286 computer. Depending on the data, assembling a map manually could take the geneticist several days.

Using the second criterion, it was possible to generate multi-gene maps that were very close to the correct maps for two of the three data sets. It was more difficult to generate correct maps for AT/ACE partly because the maps being merged were only "adequate" maps and because of the occurence of duplicate cut sites.

6 Summary

The overall approach for automating map assembly and the success of the techniques developed were discussed.

CHAPTER 10 - SUMMARY

Determing the sequence of genes and restriction enzyme cut sites from experimental data is a process known as map assembly. It is a difficult problem due to the amount of error in the data. There are many sequences possible and several may be feasible. No means of objectively assessing competing maps has been available; the process has relied solely on the judgement and expertise of the geneticist which has lead to incorrect maps being published. Various computer applications have been reported for map assembly, however, they either tackle related problems or a restricted instance of the problem.

This thesis has investigated and developed suitable artificial intelligence techniques to automate the process.

A system for objectively assessing single gene maps was devised. Critical problems in maps published by Sefton et al(1990) were highlighted that lead to the geneticist's revision of the maps.

Optimal single gene maps for different data sets were generated using a hybrid genetic algorithm(HGA) tailored specifically for the map assembly problem. The HGA incorporated the objective system to evaluate maps and was successful in generating feasible maps much quicker than if assembled manually. A facility was developed to merge the best maps generated for the individual genes together to create a complete map containing several genes. Depending on the quality of maps, the merge facility could produce very similiar maps to the maps assembled manually by the geneticist.

CHAPTER 11 - CONCLUSIONS

1 Introduction

In spite of the widespread and successful application of computers to problem solving, there still exist several classes of problems that cannot be solved by automatic means. The aim of this thesis was to develop suitable artificial intelligence (AI) techniques to tackle difficult problems. General purpose AI techniques were devised, implemented and evaluated in the context of a real-world problem from the biological domain. The particular problem was chosen as it exhibited many of the characteristics associated with difficult problems ; it was a highly constrained combinatorial optimisation problem that relied on expert judgement and had to deal with incomplete and conflicting data.

The conclusions outline the general techniques that were developed by the thesis; the wide range of problems to which they could be applied; the outcome of their application in a particular problem domain; and suggestions for future work.

2 General Techniques

Three general AI techniques were developed. The techniques and the types of problems to which they could be applied are summarised in the following sections.

* A modified genetic algorithm search technique

A new search technique was developed based on a genetic algorithm (GA) approach to handle a highly constrained, combinatorial optimisation problem. The GA developed in this thesis used a natural representation and operators were devised which were based on the expert's approach to solving the problem. Other applications of genetic algorithms to sequencing problems have used various penalty or repair mechanisms where either the fitness function penalised illegal solutions or repair mechanisms were used to correct illegal solutions. Here, specialised operators coupled with the chosen representation disallowed the generation of illegal solutions and provided a mechanism for handling a highly constrained problem. The method of constructing a GA based search technique and the decisions involved in that process are applicable to other combinatorial problems; especially highly constrained problems.

Crucial to the success of any GA is the choice of fitness function. The thesis has shown how a GA approach could be applied to a problem which relied on subjective judgement. By using the second method developed in this thesis (described below), an objective measure of a subjective process was devised. The objective measure formed the basis of a composite fitness function which was incorporated into the GA.

*** A method for encapsulating expert judgement**

A method was developed to generate an objective measure of a subjective, judgemental process. The method consisted of the four stages: Knowledge Elicitation; Analysis; Quantification; and Application, as described in chapter 4.

There are many problem solving activities from different disciplines that rely on expert appraisal to evaluate between competing solutions. Typically, expert judgement is subjective as it draws upon past experience coupled with the use of expert knowledge to evaluate any particular problem instance. In order to tackle problems by computer that depend on such subjective evaluation, some means of identifying, capturing and quantifying characteristics of good solutions and of dealing with uncertainty present in the data is essential. The method devised in this thesis can be applied to similar types of problem to generate an objective measure of a subjective process. Once generated, the objective measure can be used (where applicable) by any search technique to evaluate potential solutions. Here, the objective measure was incorporated into a GA based search technique.

*** A problem solving strategy**

A strategy for tackling highly-constrained, real-world, combinatorial optimisation problems was developed based on an assessment of the problem characteristics and of the problem solving process. An approach was employed which involved decomposing the problem into smaller units to reduce the overall problem complexity, solving the sub-units and merging them together to create a complete solution. The search space of the problem was shown to be very large and even when decomposed into sub-units, the size of the search space for a restricted problem instance was shown to be an exponential function of the input. The problem solving strategy devised was based on using a modified GA as a heuristic search technique to generate sub-units and a merge facility to generate complete maps.

The same process for devising a suitable problem-solving strategy could be applied to other real-world problems. Typically, automating real-world problems is often complicated as the problem characteristics are such that no single technique is appropriate. In practice, a range of techniques would be considered and various trade-offs tend to be made to arrive at a suitable problem solving strategy.

3 Application

The application of the general methods developed in this thesis to a real-world problem has illustrated how a difficult activity that previously relied on expert judgement and inaccurate data could be automated.

By applying the modified genetic algorithm as a heuristic search technique to this highly constrained optimisation problem, optimal solutions were generated directly from error-prone data by sampling only a small number of alternatives. The system provided a quick (couple of hours rather than several days) automatic alternative to solving the problem manually. The use of GA features (genetic operators and creating new maps from better than average old maps) resulted in fitter populations of maps than in the absence of those features (no operators and creating new maps at random.)

By applying the method for encapsulating expert judgement an objective measure for evaluating potential solutions was produced. The objective measure was applied to published problem solutions generated by the expert and it was successful in highlighting critical errors in the expert's solution. The expert subsequently revised the published solution to remove the errors.

The overall task of generating a multi-gene map was broken down into the simpler, yet still difficult and combinatorial problem of generating single gene maps first then merging them together to produce a complete map. In so doing, two distinct but potentially related ordering problems have been attacked. The approach overcame many of the limitations of previous attempts at applying computing techniques to the problem. Bearing in mind the difficulties of the expert in solving the problem manually, that the automated system developed could produce reasonable maps at all, was a major result.

4 Future Work

The general methods developed in the thesis could be applied and evaluated in different application domains and the methods themselves could be revised and extended in a variety of ways, taking into account recent work in the field.

The method developed by the thesis for representing expert judgement was based on the expert rating a given set of problem situations through the use of a questionnaire. The problem situations were devised to assess how the expert handled uncertainty in the data. There are various techniques that can be used to augment knowledge representation with statistical measures that describe levels of evidence and belief. Alternative techniques for quantifying expert judgement could be evaluated and, if promising, could be used to enhance the method described here.

Ways of extending the modified GA to incorporate local search techniques could be investigated. The results of applying the modified GA suggested that if some form of local search had been employed at the end of a run, a greater number of acceptable maps could have been generated from the final population of maps. Moscato and Norman(1992) refer to evolutionary algorithms in which local search plays a significant role as "memetic algorithms". Recent empirical results from Radcliffe and Surry(1994a)(1994b) using the Travelling Salesman Problem support the view that incorporating local search in GAs leads to superior performance.

Appropriate methods for expanding the modified GA technique to handle multi-modal optimisation could be assessed. In the particular application, several optimal solutions were possible for a data set and the best results were obtained from multiple runs. Goldberg and Richardson(1987) found that simple GAs converged to a single peak when dealing with multi-modal functions, even though there may have been multiple peaks of equal quality. Deb and Goldberg(1989) investigated modifications to the simple GA to implement a form of sharing analagous to that observed in nature, where stable subpopulations of organisms surrounding separate niches are formed by forcing similiar individuals to share the available resources. Beasley, Bull and Martin(1993) describe a new technique called "sequential niche" for multi-modal function optimisation where once an optimum is found, the evaluation function is modified to eliminate the solution to prevent the same optimum from being re-discovered. In doing so, subsequent runs incorporate the knowledge discovered in previous runs. Spears(1994) has also proposed an algorithm that implements the ideas of sharing and restrictive mating.

The nature of the representation and the interaction of operators with the representation are key elements in determining the performance of modified GAs. The choice of representation and operators for the modified GA were heavily influenced by the highly constrained nature of the problem. The approach advocated by Davis(1991) of hybridising a GA with domain-specific knowledge to generate operators for real-world problems was followed. Alternative representations and operators could be investigated based on Radcliffe's "Design Principles" (Radcliffe (1991a, 1991b, 1993) in the context of the forma analysis framework. Analysing the role of the operators developed using the key criteria of "Respect" and "Assortment" could highlight ways in which these operators could be improved with the chosen representation. Radcliffe(1994a) has identified several characteristics of representations and has presented empirical evidence of a useful performance predictor (fitness variance of formae) for evolutionary algorithms.

There are various refinements that could be made to consolidate the application of the methods described in the thesis and these are outlined.

There are a number of ways the objective mechanism could be enhanced to improve performance in the particular applications area. Implementation aspects of evaluation 1 and evaluation 3 could be reviewed as it was possible, in some circumstances, that good individuals were overlooked. The scoring system on which the objective mechanism was calibrated was based on the response of one group of experts to the questionnaire. The questionnaire could be extended and a wider sample of experts could be used to complete the responses to make the scoring system more robust.

The objective evaluation mechanism developed for the problem represented the encapsulation of the expert's criteria for assessing maps. Although incorporated into the modified GA, the mechanism could also operate in a stand-alone capacity and could easily be incorporated into other tools and be used to independently validate existing and proposed maps.

The merge facility was basic and there are several areas that offer scope for improvement. More effective ways of combining single gene maps could be investigated, possibly by incorporating some "look-ahead" mechanism to reduce the amount of duplication between maps which should increase the accuracy of the maps. Various enhancements could be made to the way complete maps were evaluated by the merge facility. Utilising the faster processing capability of a workstation, it is possible that potential gene maps could be generated and viewed by the expert in real-time. Optimal maps could be displayed graphically and the results of alternative ways of merging maps could be assessed and compared.

Appendix A - The three sets of data from which maps were generated from.

B			M			S			F		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	355	350	350	350	255	255	70	165	(190)	(195)
225	225	275	260	260	80				(10)	(135)	(175)
	(65)	230		(180)						80	135
										65	
										10	

B/M			B/S			B/F			M/S		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	230	125	125	70	165	(190)	135	255	255	70
225	225	105		(65)		(10)	50	105		(180)	30
130	130						10				
	(65)										

M/F			S/F		
PIL	PI	AACT	PIL	PI	AACT
85	80	135	70	(130)	30
(10)	65	85	(10)	80	
	10			65	
				10	

The experimental data for the PIL,PI AACT genes taken from Sefton et al (1990).

Four restriction enzymes were used B,M,S,F and were applied on their own and in combinations. Each number in the table represents a fragment of DNA that contains the gene shown. Fragments in brackets indicate suspect fragments.

B		M		S		F	
AT	ACE	AT	ACE	AT	ACE	AT	ACE
260	225	600	600	465	(125)	285	285
65	115	535	535	425	75	260	260
		355	355	310		170	200
		310	310	275		130	170
						85	145
						60	125
						10	

B/M		B/S		B/F		M/S	
AT	ACE	AT	ACE	AT	ACE	AT	ACE
260	225	260	(125)	(170)	125	465	(125)
100	150	140	75	65	100	425	75
65	100	65	(40)	50		(310)	30
				20		275	
				10		235	

M/F		S/F	
AT	ACE	AT	ACE
(285)	(285)	(170)	70
260	260	130	35
(170)	200	85	
130	170	60	
85	145	10	
60	120		
10			

The experimental data for AT,ACE.

Four restriction enzymes were used B,M,S,F and were applied on their own and in combinations. Each number in the table represents a fragment of DNA that contains the gene shown. Fragments in brackets indicate suspect fragments.

B			M			S			F		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	460	340	340	340	255	255	70	165	125	135
225	225	380	260	260	80					80	
		355								65	
		335								10	
		275									
		230									

B/M			B/S			B/F			M/S		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	460	125	125	70	165	45	135	255	255	70
225	225	380					10	105			30
130	130	355									
		335									
		330									
		275									
		250									
		230									
		210									
		205									
		105									
		80									

M/F			S/F		
PIL	PI	AACT	PIL	PI	AACT
165	125	135	80	125	30
85	80	80		80	
	65			65	
	10			10	

The perfect data calculated by the author that would be expected from the map for the PIL, PI, AACT genes contained in Sefton et al (1990). All fragments are present and their lengths are accurate.

Four restriction enzymes were used B,M,S,F and were applied on their own and in combinations. Each number in the table represents a fragment of DNA that contains the gene shown.

Appendix B - Analysis of the discrepencies in three maps for the PIL/PI/AACT genes.

The three maps for the PIL/PI/AACT genes were the Sefton map, the author's map and the Billingsley map. The data expected from each map was calculated and compared with the data observed. Discrepencies between the expected(EXP) and observed(OBS) data are highlighted with an asterisk and a letter and their meaning is described in the foot-notes.

SEFTON MAP

GENE	PIL		PI		AACT	
RE	EXP.	OBS.	EXP.	OBS.	EXP.	OBS.
B	-	225*a	-	225*g	-	230*l
	250	250	250	250	260	275*m
					355	355
M	260	260	260	260	80	80
	340	350*b	340	350*h	340	350*n
S	255	255	255	255	70	70
F	165	165	10	10	135	135
			65	65		
			80	80		
B/M	130	130	130	130	80	-*o
	-	225*c	-	225*i	115	105*p
	-	250*d	-	250*j	210	230*q
B/S	125	125	125	125	70	70
B/F	165	165	10	10	115	105*r
			45	50*k	135	135
S/M	255	255	255	255	30	30
					-	70*s
M/F	85	85	10	10	80	85*t
	165	165*e	65	65	135	135
			80	80		
S/F	80	70*f	10	10	30	30
			65	65		
			80	80		

SEFTON MAP - footnotes.

PIL GENE

a) B - No 225 fragment is expected if the B site to the RHS of PIL is cutting completely.

The fragment has been fitted on the map with the LHS being cut by M, crossing the complete B cut site and the other end being cut by a partial b site.

The longer B fragment expected, given the partial nature of the cut site, is not observed.

b) M - The expected fragment (340) is 2.8% smaller than the observed fragment (350).

c) B/M - No 225 fragment is expected if the B site to the RHS of PIL is cutting completely.

Two longer B/M fragments would be expected, given the partial nature of the b and m cut sites, they are not observed.

d) B/M - No 250 fragment is expected if the M site to the LHS of PIL is cutting completely.

e) M/F - The fragment observed (165) is likely to be the F single digest fragment as it is the same length, however, for it to be fitted where it is, the M cut site must be cutting partially.

f) S/F - The expected fragment (80) is 14.2% larger than the observed fragment (70).

PI GENE

g) B - No 225 fragment is expected if the B site to the RHS of PI is cutting completely.

The fragment has been fitted on the map with the LHS being cut by M, crossing the complete B cut site and the other end being cut by a partial b site.

The longer B fragment expected, given the partial nature of the cut site, is not observed.

h) M - The expected fragment (340) is 2.8% smaller than the observed fragment (350).

i) B/M - No 225 fragment is expected if the B site to the RHS of PI is cutting completely.

Two longer B/M fragments would be expected, given the partial nature of the b and m cut sites, they are not observed.

j) B/M - No 250 fragment is expected if the M site to the LHS of PIL is cutting completely.

k) B/F - The expected fragment (45) is 10% smaller than the observed fragment (50).

AACT GENE

l) B - The one expected fragment is considered to represent both the 230 and 275 fragments.

The expected fragment (260) is 13% larger than the observed fragment (230).

m) B - The one expected fragment is considered to represent both the 230 and 275 fragments.

The expected fragment (260) is 5.5% smaller than the observed fragment (275).

n) M - The expected fragment (340) is 2.8% smaller than the observed fragment (350).

o) B/M - A fragment is expected (80) but not observed.

p) B/M - The expected fragment (115) is 9.5% larger than the observed fragment (105).

q) B/M - The expected fragment (210) is 8.6% smaller than the observed fragment (230).

r) B/F - The expected fragment (115) is 9.5% larger than the observed fragment (105).

s) S/M - The observed fragment (70) is not expected given that the M cut site on the RHS of AACT is cutting completely. It has been fitted over this complete cut site.

t) M/F - The expected fragment (80) is 5.8% smaller than the observed fragment (85).

PROPOSED MAP

GENE	PIL		PI		AACT	
RE	EXP.	OBS.	EXP.	OBS.	EXP.	OBS.
B	225 250	225 250	225 250	225 250	230 275 355	230 275 355
M	260 350	260 350	260 350	260 350	90 350	80*j 350
S	255	255	215	255*f	70	70
F	165	165	10 65 80	10 65 80	135	135
B/M	130 - -	130 225*a 250*b	130 - -	130 225*g 250*h	90 350	105*k 230*l
B/S	125	125	125	125	70	70
B/F	165	165	10 50	10 50	100 135	105*m 135
S/M	255	255	215	255*i	30 70	30 70
M/F	80 165	85*c 165*d	10 65 80	10 65 80	90 -	85*n 135*o
S/F	75	70*e	10 65 80	10 65 80	30	30

PROPOSED MAP - Footnotes.

PIL GENE

No. Problem Description

a) B/M - If the M cut site to the left of PIL is cutting completely, you would not expect to get a 225 fragment.

b) B/M - If the M cut site to the left of PIL is cutting completely, you would not expect to get a 250 fragment.

c) M/F - The expected fragment (80) is 5.8% smaller than the observed fragment (85).

d) M/F - The fragment observed (165) is likely to be the F single digest fragment as it is the same length, however, for it to be fitted where it is, the M cut site must be cutting partially.

e) S/F - The expected fragment (75) is 7.1% larger than the observed fragment (70).

PI GENE

f) S - The expected fragment (215) is 15.7% smaller than the observed fragment (255).

g) B/M - If the M cut site to the left of PIL is cutting completely, you would not expect a 225 fragment.

h) B/M - If the M cut site to the left of PIL is cutting completely, you would not expect a 250 fragment.

i) S/M - The expected fragment (215) is 15.7% smaller than the observed fragment (255).

AACT GENE

j) M - The expected fragment (90) is 12.5% larger than the observed fragment (80).

k) B/M - The expected fragment (90) is 14.2% smaller than the observed fragment (105).

l) B/M - The expected fragment (220) is 4.3% smaller than the observed fragment (230).

m) B/F - The expected fragment (100) is 4.7% smaller than the observed fragment (105).

n) M/F - The expected fragment (90) is 5.8% larger than the observed fragment (85).

o) M/F - The M cut site to the RHS of AACT appears to be cutting completely in the single digest results yet the 135 fragment observed is only expected if it is cutting partially.

BILLINGSLEY MAP

GENE	PIL		PI		AACT	
RE	EXP.	OBS.	EXP.	OBS.	EXP.	OBS.
B	255	280*a	255	280*i	115	115
	365	N.D*b	365	370*j	240	250*q
	395	*c	395	385*k	370	370
	465	*d	465	495*l	510	610*r
	510	*e	510	610*m	735	710*s
	735	*f	735	710*n		
F	175	165*g	9.5	9.5	100	105*t
			175	165*o	175	180*u
M				340		530
				900		1030
				1300		1300
B/F	175	165*h	9.5	9.5	70	70
			175	165*p	100	*v
					175	*w

NOTE : Only the results for B, F and M are shown to enable comparison with the Sefton map and the proposed map.

The Billingsley map was drawn to scale from the map shown in Billingsley et al(1993). The cut sites which were cutting partially were located on the map, then the expected fragments were calculated and drawn underneath the map. Differences between the lengths of the expected fragments and the observed fragments were noted. The position of the M cut sites were not indicated on the Billingsley map therefore the expected lengths could not be calculated.

BILLINGSLEY MAP - Footnotes.

PIL GENE

No. Problem Description

a) B - The expected fragment (255) is 8.9% smaller than the observed fragment (280).

b) B - A fragment of 365 is expected but not observed (N.D.).

c) B - A fragment of 395 is expected but not observed (N.D.).

d) B - A fragment of 465 is expected but not observed (N.D.).

e) B - A fragment of 510 is expected but not observed (N.D.).

f) B - A fragment of 735 is expected but not observed

(N.D.).

g) F - The expected fragment (175) is 6% larger than the observed fragment (165).

h) B/F - The expected fragment (175) is 6% larger than the observed fragment (165).

PI GENE

i) B - The expected fragment (255) is 8.9% smaller than the observed fragment (280).

j) B - The expected fragment (365) is 1.3% smaller than the observed fragment (370).

k) B - The expected fragment (395) is 2.5% larger than the observed fragment (385).

l) B - The expected fragment (465) is 6% smaller than the observed fragment (495).

m) B - The expected fragment (510) is 16.3% smaller than the observed fragment (610).

n) B - The expected fragment (735) is 2.8% larger than the observed fragment (710).

o) F - The expected fragment (175) is 6% larger than the observed fragment (165).

p) B/F - The expected fragment (175) is 6% larger than the observed fragment (165).

AACT GENE

q) B - The expected fragment (240) is 4% smaller than the observed fragment.

r) B - The expected fragment (510) is 16.3% smaller than the observed fragment (610).

s) B - The expected fragment (735) is 2.8% larger than the observed fragment (710).

t) F - The expected fragment (100) is 5% smaller than the observed fragment (105).

u) F - The expected fragment (175) is 2.7% smaller than the observed fragment (180).

v)w) B/F - Given the nature of the b and f cut sites to the LHS of AACT, 2 longer fragments of length 100 and 175 would be expected but are not observed.

Appendix C - Completed questionnaire

COMPLETED QUESTIONNAIRE

PROBLEM RATINGS - there were 5 ratings which ranged in a scale of increasing severity from rating "none" to "critical".

1. None
2. Minor problem
3. Significant problem
4. Serious problem
5. Critical problem

Each of the problems described below have been rated by the external advisors, Povey(P) and Bickmore(B) as shown.

SECTION 1 - FRAGMENT LENGTHS P B

Errors are present in the observed lengths of the fragments due to experimental limitations. The allowable error (determined previously which represents the "no problem" situation) was up to 10% on fragments < 1000kbps and up to 20% on fragments > 1000kbps.

1. In order to fit a fragment > 500 kbps,
the observed length must be out by 15% ? 2 1

2. In order to fit a fragment 200 - 500 kbps,
the observed length must be out by 15% ? 3 1

3. In order to fit a fragment < 200 kbps,
the observed length must be out by 15% ? 4 2

4. In order to fit a fragment < 1000kbps,
the observed length must be out by 20% ? 4 2

5. In order to fit a fragment > 1000kbps,
the observed length must be out by 30% ? 2 2

6. In order to fit a fragment > 1000kbps,
the observed length must be out by 40% ? 4 3

SECTION 2 - NATURE OF CUT SITES

The objective of this section is to determine the importance of the nature and number of cut sites relative to the fragments observed.

7. A particular cut site on a map is shown
as partial, yet the longer fragment which
would be expected, is not observed? 2 2

8. A single digest fragment is fitted into
a map with the ends being cut by different REs ? ..5 1

P B

9. From the single digest results, a cut site is considered to be cutting completely, however, the proposed map has a double digest fragment fitted across the cut site ? 2 3

10. You are trying to fit a fragment into a map which you have almost completed. The fragment length (taking into account the allowable error) does not fit in with the cut sites shown on the map.

10a) Add a new cut site and fit the fragment even though you do not observe the other fragments which would be expected as a result of an additional cut site ? 4 4

10b) Consider the fragment to be the same fragment as an existing longer or shorter fragment which has already been fitted ? 4 2

11. Two fragments are obtained in the single digest results, and the same two fragments are obtained in the double digest results. You are trying to fit both fragments into a map which you have almost completed. Only one fragment fits in with the cut sites shown on the map. If you add a new cut site, you do not observe the other double digest fragments which would be expected.

11a) Add a new cut site ? 3-4 2

11b) Consider the two fragments to be the same fragment ? 4 3

12. From the proposed map, you would expect fragments that you did not observe from the experiment ? 2-4 2-3

13. You observe fragments from the experiment that would not be expected from the proposed map ? 4-5 3

14. What kind of problem would you consider it to be if the smallest fragment detected from your experiments for a particular gene (single or double digest), was not the smallest fragment fitting across the gene given a proposed map ?....................3 3

SECTION 3 - WEAK FRAGMENTS

15. Given the number and position of the cut sites on the proposed map, none of the weak fragments fit without having to add extra cut sites ? 3 3

16. In order to fit a weak fragment, an additional cut site is added. You do not observe the extra fragments you would expect having added an extra cut site ? 3 3

17. The "no problem" situation for weak fragments is when what percentage fit into the proposed map (weak fragments being those fragments expected from a proposed map but no observed) ?........................100%   100%   P   B

18. What percentage of weak fragments fitting into the proposed map would you consider to be a minor problem ?..............80%   75%

19. What percentage of weak fragments fitting into the proposed map would you consider to be a significant problem ?.........0%   0%

20. In order to fit a weak fragment, an additional cut site is added. You don't observe the extra fragments you would expect having added the extra cut site ?.......3   3

21. If you saw the same length of weak fragments in both the single and double digests, would you be more confident that the bands on the gel represented actual fragments than if the weak fragments only appeared in either the single or the double digest ?......................................Y   N

SECTION 4 - LENGTH OF SINGLE DIGEST FRAGMENTS WITH RESPECT TO EACH OTHER

22. Given a proposed map, you are looking at the single digest data. One single digest fragment is nested within another single digest fragment. (Povey's answers do not include the 10% allowable variation.)

```
eg  M    S   gene    S      M
  --|----|------------|------|
         |------------|
            S 100
    |------------------------|
           M 50
```

a) On the same gels, the length of the nested fragment is greater than the outer fragment by 50% ?.....................................5   5

b) On the same gels, the length of the nested fragment is greater than the outer fragment by 25% ?.....................................4   4

c) On the same gels, the length of the nested fragment is greater than the outer fragment by 10% ?.....................................2   3

d) On different gels, the length of the nested fragment is greater than the outer fragment by 50% ?.....................................5   3

e) On different gels, the length of the nested fragment is greater than the outer fragment by 25% ?.......................................4 2 — P B

f) On different gels, the length of the nested fragment is greater than the outer fragment by 10% ?.......................................2 1

SECTION 5 - FIT OF DOUBLE DIGEST FRAGMENTS

Looking through the observed double digest data, fragments can be classed as either single digest fragments (not altered in the double digest) or new fragments. A similiar exercise can be carried out looking at the proposed map, and the expected double digest data.

23. There are differences in the number of double digest fragments you have observed and the number you would expect. There are also differences in the types of fragments (single digest/new) between those observed and those expected.

23a) You observe 10 double digests, from the proposed map, you would expect 15 double digests ? ....2-3 -

23b) You observe 15 double digests, from the proposed map, you would expect 10 double digests ? ....4-5 -

24. Out of the type of double digest fragments EXPECTED from the map, what % would have to be observed for there to be -

a) no problem ?...............................70% -
b) significant problem ?......................60% -
c) critical problem ?.........................50% -

What percentage of the type of double digest fragments OBSERVED would have to be expected for there to be -

d) no problem ?...............................95% -
e) significant problem ?......................90% -
f) critical problem ?........................<90% -

SECTION 6 - MAP ASSESSMENT

The objective of this section is to determine how the problems identified could be related to one another to allow an overall assessment of a map to be made. I would like to categorise a map in one of the following ways depending on the number of minor, significant, serious and major problems present.

MAP CATEGORIES

Very Good Map
Good Map
Acceptable Map
Unacceptable Map

P B

25. How many minor problems would you consider make a significant problem ? ....3 >3

26. How many significant problems would you consider make a serious problem ? ....3 >2

27. How many serious problems would you consider make a critical problem ? ....2 >1

The size of the map must be taken into account.

28. What do you consider are the most important variables which represent map size -

a) number of probes used ?.....................N Y
b) total number of single and double digest fragments ? ....Y N
c) length of DNA being mapped ?................N Y

In terms of the variables identified above,

29. Define a small map?...............< 10frags 10-20
- 2-3probes
- 100-200kbps

30. Define a medium map ?............10-20frags 25-50
- 3-6probes
- 500-1000kbps

31. Define a large map ?................>20 >50
- 10probes
- >1000kbps

How many problems of the various types would be allowed for a -

32. Very good SMALL MAP ?................None None

33. Good SMALL MAP ?..............1 Minor 1 Minor

	P	B
34. Acceptable SMALL MAP ?............	2 Minor or 1 Signif	1 Minor 1 Signif
35. Unacceptable SMALL MAP ?.........	Any Serious or Critical	1 Minor 1 Signif 1 Serious
36. Very good MEDIUM MAP ?.................	None	1-2 Minor
37. Good MEDIUM MAP ?...............	1-2 Minor	1 Minor 1 Signif
38. Acceptable MEDIUM MAP ?...........	3-4 Minor or 1 Signif	1-2 Minor 1-2 Signif
39. Unacceptable MEDIUM MAP ?........	Any Serious or Critical	2 Minor 1 Signif 1 Serious
40. Very good LARGE MAP ?................	None	1-2 Minor
41. Good LARGE MAP ?..............	1-3 Minor	1-2 Minor 1 Signif
42. Acceptable LARGE MAP ?............	4+ Minor or 1 Signif	2 Minor 1-2 Signif
43. Unacceptable LARGE MAP ?.........	Any Serious or Critical	2 Minor 2 Signif 1 Serious

Appendix D - Scoring the Sefton, Proposed and Billingsley Maps

Three maps for the PIL/PI/AACT genes, the Sefton map, Proposed map and Billingsley map were scored using the questionnaire results. The number of the problem is represented by a letter which correspond to the description of the problem shown in Appendix B. The rating applied to the problem is shown (eg MINOR, CRITICAL etc) and in brackets after the rating is the reference to the questionnaire number.

SEFTON MAP

PIL	PI	AACT

a) MINOR(9)
CRITICAL(8)
MINOR (7)
	g) MINOR(9)
	CRITICAL(8)
	MINOR(2)
		l) SERIOUS(11b)
		MINOR-SIGNIFICANT(2)

b) ALLOWABLE ERROR	h) ALLOWABLE ERROR	m) SERIOUS (11b) ALLOWABLE ERROR
c) MINOR (9) MINOR (7) MINOR (7)	i) MINOR (9) MINOR (2) MINOR (2)	n) ALLOWABLE ERROR
d) MINOR (9)	j) MINOR (2)	o) MINOR (7)
e) MINOR(9)	k) ALLOWABLE ERROR	p) ALLOWABLE ERROR
f) SIG.- SERIOUS (2)		q) ALLOWABLE ERROR
		r) ALLOWABLE ERROR
		s) MINOR (9)
		t) ALLOWABLE ERROR

TOTAL

7 minor 1 sig.- serious 1 critical	6 minor 1 critical	2 minor 1 minor - sig. 2 serious

PROPOSED MAP

PIL	PI	AACT
a) MINOR (9)	f) SIG. (2)	j) SIG.-SERIOUS (3)
b) MINOR (9)	g) MINOR (9)	k) SIG.-SERIOUS (3)
c) ALLOWABLE ERROR	h) MINOR (9)	l) ALLOWABLE ERROR
d) MINOR (9)	i) SIG. (2)	m) ALLOWABLE ERROR
e) ALLOWABLE ERROR		n) ALLOWABLE ERROR
		o) MINOR (9)
TOTAL		
3 minor	2 minor 2 sig.	1 minor 2 sig.-serious

BILLINGSLEY MAP

PIL	PI	AACT
a) ALLOWABLE ERROR	i) ALLOWABLE ERROR	q) ALLOWABLE ERROR
b) MINOR-SER.(12)	j) ALLOWABLE ERROR	r) MINOR-SER.(1,4)
c) MINOR-SER.(12)	k) ALLOWABLE ERROR	s) ALLOWABLE ERROR
d) MINOR-SER.(12)	l) ALLOWABLE ERROR	t) ALLOWABLE ERROR
e) MINOR-SER.(12)	m) MINOR-SER.(1,4)	u) ALLOWABLE ERROR
f) MINOR-SER.(12)	n) ALLOWABLE ERROR	v) MINOR (7)
g) ALLOWABLE ERROR	o) ALLOWABLE ERROR	w) MINOR (7)
h) ALLOWABLE ERROR	p) ALLOWABLE ERROR	
TOTAL		
5 minor - serious	1 minor - serious	2 minor 1 minor-serious

Appendix E - The revised Sefton map

The individual published and revised maps for each of the PIL/PI/AACT genes are shown below. The published maps for PIL and PI were considered to contain critical problems because a partial B cut site was shown outwith a complete B cut site.

PIL

published	B		F	M	S	*	F	B	b	S m	M
revised	B	**b**	F	**m**	S	*	F	B	–	S m	**m**

Compared to the published map, the revised PIL map has an extra B cut site on the left hand side rather than on the right hand side. 2 M sites have been changed from complete to partial.

PI

published	B		M	S	F	*	f	B	f	F	b	S m	M
revised	B	**b**	**m**	S	F	*	f	B	f	**f**	–	S m	**m**

Compared to the published map, the revised map has an extra B cut site and both M complete cut sites and one of the F complete cut site have been changed to partial cut sites.)

AACT

published	M	B	F	b	m	S	*	F M	S			B
revised	**m**	–	F	b	m	S	*	F **m**	S	**b**	**b**	B

Compared to the published map, there are 2 additional partial B cut sites, 2 M complete cut sites have been changed to partial and a complete B cut site has been removed. The changes have been highlighted.

Complete map

The complete revised map, taking account of the changes detailed above, is shown below. Three extra cut sites were added (1 B site to the left of the PIL gene and 2 B sites to the right of the AACT gene) and three cut sites were changed from being complete to partial cut sites (m site to the right of AACT, f site at map position 105, m site to the left of PIL).

```
                     m       F
BbFmS*PIL*F*PI*fBffFbSS*AACT*mSbbB
```

With the revised Sefton map, the expected data and the observed data were calculated. Discrepancies were highlighted and were rated using the questionnaire. The results are summarised here. All gene maps would be considered acceptable.

	PIL	PI	AACT
REVISED SEFTON MAP	1 minor 1 significant	2 minor	5 minor
SCORE =	4-adequate	2-good	5-adequate

Analysis of the discrepencies in the Revised Sefton map for PIL/PI/AACT.

GENE	PIL		PI		AACT	
RE	EXP.	OBS.	EXP.	OBS.	EXP.	OBS.
B	225	225	225	225	230	230
	250	250	250	250	275	275
					355	355
					355+	- *d
M	260	260	260	260	80	80
	340	350*b	340	350*h	340	350*n
	340+	- *u	340+	- *v	340+	-*a
S	255	255	255	255	70	70
F	165	165	10	10	135	135
			65	65		
			80	80		
			80+	- *w		
B/M	130	130	130	130	80	-*o
	225	225	225	225	105	105
	250	250	250	250	230	230
					275	-*b
					355	-*c
					355+	-*e
B/S	125	125	125	125	70	70
B/F	165	165	10	10	105	105
			45	50*k	135	135
S/M	255	255	255	255	30	30
					80	70*f
M/F	85	85	10	10	80	85*t
	165	165	65	65	135	135
			80	80		
			80+	- *x		
S/F	80	70*f	10	10	30	30
			65	65		
			80	80		
			80+	- *y		

REVISED SEFTON MAP - Footnotes.

PIL GENE

(An extra b cut site has been added to the LHS of PIL and the M site to the LHS of PIL has been changed to partial.)

b) M - The expected fragment (340) is 2.8% ALLOWABLE
smaller than the observed fragment (350). ERROR

f) S/F - The expected fragment (80) is 14.2% SIG.-SERIOUS
larger than the observed fragment (70). (2)

u) M - If the M site to the LHS of PIL is partial, MINOR
would expect a longer M fragment. (7)

TOTAL = 1 minor
1 sig. - serious

PI GENE

h) M - The expected fragment (340) is 2.8% ALLOWABLE
smaller than the observed fragment (350). ERROR

k) B/F - The expected fragment (45) is 10% ALLOWABLE ERROR
smaller than the observed fragment (50).

v) M - If the M site to the LHS of PIL is partial, MINOR
would expect a longer M fragment. (7)

w) F - A longer F fragment would be expected. MINOR (7)
x) M/F
y) S/F

TOTAL = 2 minor

AACT GENE

n) M - The expected fragment (340) is 2.8% ALLOWABLE
smaller than the observed fragment (350). ERROR

t) M/F - The expected fragment (80) is 5.8% ALLOWABLE
smaller than the observed fragment (85). ERROR

b) B/M - Would expect 6 fragments - only 3 3 MINOR (7)
c) observed.
e)

d) B - Would expect more fragments. 1 MINOR (7)

f) S/M - The expected fragment (80) is 14.3% larger
than the observed fragment (70). MINOR-SIG (2)

TOTAL = 4 minor, 1 minor - sig.

Appendix F - Examples of applying evaluations 1, 2 and 3.

Examples of how the three evaluations would be applied to maps are given.

1. EVALUATION 1

Examples are given of how evaluation 1 would be applied to three different maps generated from the observed data for gene P. 3 restriction enzymes were used, B, M and S and the data observed for each of the single digests is shown below.

Restriction enyzme	P fragment length
B	200 100
M	50
S	300

Evaluate map S B b M * M B S using evaluation 1.

To evaluate gene map S B b M * M B S, the single digests expected from the map are calculated and the observed fragment lengths are assigned to them as described below.

	map	S	B	b	M	*	M	B	S
	cut site position	1	2	3	4		5	6	7

The observed BB200 fragment length is assigned to the expected BB(2,6) fragment. The observed BB100 fragment length is assigned to the expected bB(3,6) fragment. The observed MM50 fragment length is assigned to the expected MM(4,5). The observed SS300 fragment length is assigned to the expected SS(1,7) fragment.

The single digest fragments are checked to see if they are nested within other single digest fragments as shown.

```
MM(4,5) 50 fragment is nested within
            bB(3,6) 100
            BB(2,6) 200
            SS(1,7) 300 - no problem.

bB(3,6) 100 fragment is nested within
            BB(2,6) 200
            SS(1,7) 300 - no problem.

BB(2,6) 200 fragment is nested within
            SS(1,7) 300 - no problem.
```

The observed single digest data fits into the proposed map so the map score is unaltered.

Evaluate map M B b S * S B M using evaluation 1.

```
             gene map            M B b S * S B M
                         position 1 2 3 4   5 6 7

SS(4,5) 300 fragment is nested within bB(3,6) 100 - penalty
                                      BB(2,6) 200 - penalty
                                      MM(1,7) 50  - penalty

bB(3,6) 100 fragment is nested within BB(2,6) 200
                                      MM(1,7) 50 - penalty

BB(2,6) 200 fragment is nested within MM(1,7) 50 - penalty
```

The score is increased by 5 points, one for each penalty.

Evaluate map SBbM*SBM using evaluation 1.

```
            gene map            S B b M * S B M
                        position 1 2 3 4   5 6 7
```

bB(3,6) 100 fragment is nested within BB(2,6) - no problem.

There is only 1 fragment which is nested within another. Note that all the other single digest fragments are staggered in the map. In such cases, a map in which the fragments are staggered (ie. where no single digests nest within one another) is considered a "no problem" map.

2. EVALUATION 2

Gene PI has had 4 restriction enzymes applied, B, M, S and F and the data observed for each of the single digests is shown below. The double digest data observed for PI is shown in chapter 1, table 1.1.

Restriction Enzyme	PI fragment length
B	250 225
M	350 260
S	255
F	80 65 10

Step 1 - Classify each double digest fragment observed for PI in chapter 1, table 1.1 as either a new double digest fragment or as a single digest fragment, and calculate the ordering of the cut sites.

	obs length	type of fragment	ordering of cut sites
BM	250	B single digest	B * B
	225	B single digest	B * b or b * B
	130	new fragment	B * M or M * B
BS	125	new fragment	B * S or S * B
BF	50	new fragment	B * F or F * B
	10	F single digest	f * f
MS	255	S single digest	S * S
MF	80	F single digest	F * F
	65	F single digest	f * F or F * f
	10	F single digest	f * f
SF	80	F single digest	F * F
	65	F single digest	f * F or F * f
	10	F single digest	f * f

Step 2 - Calculate the ordering of the double digests expected from the map **B M b S m F f * f B S M F** . Compare with the ordering of cut sites of observed data calculated in step 1.

A = double digest
B = number of observed fragments
C = classification of the observed fragments
D = number of fragments expected
E = orderings of fragments expected
F = difference in number between the number of fragments observed and expected
G = difference in the orderings of the observed and expected fragments

A	B	C	D	E	F	G
BM	3	130(new)	3	m*B(new)	0	0
		225(B sd)		b*B(B sd)		0
		250(B sd)		M*B(new)		1
BS	1	125(new)	1	S*B(new)	0	0
BF	2	10(F sd)	4	f*f(F sd)	2	0
		50(new)		f*B(new)		0
				F*B(new)		1
				F*f(F sd)		1
MS	1	255(S sd)	2	S*S(S sd)	1	0
				m*S(new)		1
MF	3	80(F sd)	4	F*M(new)	1	1
		65(F sd)		f*M(new)		1
		10(F sd)		f*f(F sd)		0
				F*f(F sd)		0
SF	3	80(F sd)	4	F*f(F sd)	1	0
		65(F sd)		f*S(new)		1
		10(F sd)		f*f(F sd)		0
				F*S(new)		1
	13		18(some may be lost) (or appear as weak) (fragments)		5	8

Given the suggested map, out of the 18 double digest fragments that would be expected, 5 are not observed from the data and 8 of the expected fragments orderings do not match up with those observed.

Step 3 - Allocate a score for the map based,

3a) on the % of the type of double digest fragments observed which are expected -

		SCORE
95%+	- no problem	= 0
91 - 94%	- minor problem	= 1
90%	- significant problem	= 3
	- serious problem	= 9
<90%	- critical problem	= 18

There were 13 double digests observed and 10 were expected from the map being considered. The % of the type of double digest fragments observed which are expected is 10/13 = 76.92% score = 18.

3b) on the % of the type of double digest fragments expected which are observed -

		SCORE
70%+	- no problem	= 0
61 - 69%	- minor problem	= 1
60%	- significant problem	= 3
51-59%	- serious problem	= 9
<50%	- critical problem	= 18

Out of the 18 fragments expected, 10 of these were observed. The % of the type of double digest fragments expected which are observed is 10/18 = 55.55% score = 9.

Map score based on evaluation 2 = 18 + 9 = 27.

3. EVALUATION 3

The aim of evaluation 3 was to calculate the size of the intervals between the cut sites in a single gene map.

Each of the fragment lengths was represented as an equation with the unknown intervals on the left hand side of the equation and the corresponding fragment length on the right hand side of the equation. The equations were entered into an (n,m) matrix aa[n,m], where n=number of equations and m=number of unknown intervals. The corresponding fragment lengths l, were represented as a vector of size n. The problem was regarded as that of solving n simultaneous equation to produce the lengths of the m unknowns.

Three situations were possible depending on the number of simultaneous equations and the number of unknown intervals in the matrix.

If there were the same number of equations as there were unknown intervals, the matrix was a **square matrix** and the equations in the matrix could be solved. Gaussian elimination with partial pivoting was used to solve the equations. The solutions represented the least squares best fit for the intervals in the map. As the solutions were computed solutions rather than exact solutions, a series of corresponding residual vectors were generated that indicated the amount of error associated with each solution.

If there were more equations than there were unknown intervals (n > m), the matrix was **over-prescribed.** In such situations, an attempt was made to produce a square matrix by pre-multiplying the matrix on the left by the transpose of the matrix aa[n,m]. The vector l[n] was multiplied by the transpose of matrix aa[n,m]. In practical cases, premultiplying the matrix by its transpose normally resulted in a non-singular matrix. The equations were then solved using Gaussian elimination with partial pivoting. The solutions represented the least squares best fit to the equations. Arnold(1990) provides a more detailed treatment of the subject.

If there were fewer equations than unknown intervals, the matrix was **under-prescribed.** In such cases, the data was not considered to fit well into the single gene map and the map was allocated a poor score.

Example

An example of how evaluation 3 was applied to a single gene map is described using the experimental data for PI shown in appendix A.

1. Represent the unknown intervals and corresponding fragment lengths as a series of simultaneous equations.

The positions of the cut sites and gene were numbered as were the intervals between the cut sites in the map, as illustrated overleaf.

```
position 0  1   2  3  4  5  6  7  8  9  10 11  12
map      B  b   m  S  F  *  f  B  f  f  S  m   m
interval   0  1   2  3   4   5  6  7  8  9  10
```

1.1 The equations for the single digests were inserted into matrix aa[n,m], where n=no. of equations (also the no. of fragment lengths) and m=no. of unknown intervals. The fragment lengths (solutions) for each of the equations were inserted into a vector l[n]. For example, in the first equation shown below, the intervals 0 to 5 inclusive had to add up to equal length 250.

	interval (m)												DD		no.of poss.
	0	1	2	3	4	5	6	7	8	9	10			posn	
no. of eqn (n)												length (l)			
aa[0]=	1	1	1	1	1	1	0	0	0	0	0	250	BB	0, 7	1
aa[1]=	0	1	1	1	1	1	0	0	0	0	0	225	bB	1, 7	1
aa[2]=	0	0	1	1	1	1	1	1	1	1	0	260	mm	2,11	1
aa[3]=	0	0	1	1	1	1	1	1	1	1	1	350	mm	2,12	1
aa[4]=	0	0	0	1	1	1	1	1	1	0	0	255	SS	3,10	1
aa[5]=	0	0	0	0	1	0	0	0	0	0	0	10	Ff	4, 6	1
aa[6]=	0	0	0	0	1	1	1	0	0	0	0	65	Ff	4, 8	1
aa[7]=	0	0	0	0	1	1	1	1	0	0	0	80	Ff	4, 9	1

1.2 The equations for the double digests were inserted into matrix aa[n,m]. The solutions for each of the equations were inserted into vector l[n].

This process could be problematic for double digests as there was no one to one relationship between the number of fragments observed and the number of fragments expected. There could be more or less fragments expected than observed. If 2 fragments were observed, but 4 were expected (it could be argued that this was a normal state of affairs as you often do not see all the fragments you would expect), how would you decide which of the 4 fragments were the 2 fragments you observed ? The only fragment whose position you could be reasonably confident about was the smallest fragment, which must lie across the gene.

For each double digest, the expected fragments were classed as single digest fragments or as new double digest fragments. A double digest could be classed as a single digest if both end cut sites were the same and the fragment occupied the same position and had the same length as the single digest fragment. If the fragment was classed as a new double digest, a check was made to see if the length of the observed new double digest could be assigned to the expected fragments. If the length of the observed double digest was greater than the length of any single digest, then an expected double digest which was smaller than the single digest was not feasible. (The position of the expected double digest gave an indication of its size in relation to the known size and position of a single digest.)

```
        interval (m)                      DD        no.of poss.
        0 1 2 3 4 5 6 7 8 9 10                posn
no. of                             length
eqn (n)                             (l)
aa[ 8]= 0 0 1 1 1 1 0 0 0 0 0      130   mB   2, 7   1
aa[ 9]= 0 1 1 1 1 1 0 0 0 0 0      225   bB   1, 7   1
aa[10]= 1 1 1 1 1 1 0 0 0 0 0      250   BB   0, 7   1
aa[11]= 0 0 0 1 1 1 0 0 0 0 0      125   SB   3, 7   1
aa[12]= 0 0 0 0 1 0 0 0 0 0 0       10   Ff   4, 6   1
aa[13]= 0 0 0 0 1 1 0 0 0 0 0       50   FB   4, 7   1
aa[14]= 0 0 0 1 1 1 1 1 1 0 0      255   SS   3,10   1
aa[15]= 0 0 0 0 1 0 0 0 0 0 0       10   Ff   4, 6   1
aa[16]= 0 0 0 0 1 1 1 0 0 0 0       65   Ff   4, 8   1
aa[17]= 0 0 0 0 1 1 1 1 0 0 0       80   Ff   4, 9   1
aa[18]= 0 0 0 0 1 0 0 0 0 0 0       10   Ff   4, 6   1
aa[19]= 0 0 0 0 1 1 1 0 0 0 0       65   Ff   4, 8   1
aa[20]= 0 0 0 0 1 1 1 1 0 0 0       80   Ff   4, 9   1
```

2. In this typical example, there were more equations than there were unknown variables (n > m) and the matrix was over-prescribed. To produce a square matrix, the matrix was pre-multiplied on the left by the transpose of matrix aa[n,m]. The vector l[n] was multiplied by the transpose of matrix aa[n,m].

3. The equations in the square matrix were solved to determine the size of the unknown intervals in the map, using Gaussian elimination with partial pivoting.

The coefficient with the largest absolute value in the first column was chosen as the first pivot. That equation became the pivot equation. Multiples of the pivot equation were added to all other equations to eliminate the unknown. Working through the columns in succession to eliminate the unknowns in natural order, the process was repeated for each unknown. The final system was solved by back substitution.

The solutions represented the least squares best fit for the intervals in the map and were :-

```
m[0] : 25.00
m[1] : 95.00
m[2] : 5.00
m[3] : 75.00
m[4] : 10.00
m[5] : 40.00
m[6] : 15.00
m[7] : 15.00
m[8] : 100.00
m[9] :  0.00
m[10] : 90.00
```

Map length (sum of solutions) = 470.000000

Associated with each computed solution was a residual which indicated the amount of error associated with each computed solution. The residual information is currently unused by evaluation 3. (In the example above, the residuals were all 0.)

Appendix G - Test list of maps

A test list of maps was compiled using three different sets of data. In each case, when coincident cut sites were present in a map, both sequences were given. Eg. if the cut sites "B" and "S" were coincident in a map, two sequences would be used - the map with sequence "BM" and the map with sequence "MB". The sequence number was indicated at the end of the map.

1. Maps generated using the "perfect" data for PIL/PI/AACT. The complete revised published map was taken containing all three genes and the "perfect" gene maps were deduced. The "perfect" data expected from the maps was calculated and is shown in Appendix A. The "perfect" maps were identified with a preceeding "P".

PPIL1 = B b F m S * F B S m m
PPIL2 = B b F m S * F B m S m

PPI1 = B b m S F * f B f f F S m m
PPI2 = B b m S F * f B f f F m S m

PAACT1 = m B F b m S * F m S b b B
PAACT2 = m B F b m S * m F S b b B

2. Maps generated using the PIL/PI/AACT data taken from Sefton et al(1990), shown in chapter 1, table 1.1. The maps comprised -

a) The revised published gene maps. As the published PIL and PI gene maps contained a "critical" problem (in the number and nature of B cut sites), the published maps were revised by the Sefton group. The revised published maps represented the maps that the geneticist considered were the optimal maps for the data. These maps were identified as "PUBR" maps.

b) The proposed gene maps. These were the gene maps assembled manually by the author using the same data as Sefton. These were identified as "PROP" maps.

PIL-PUBR1 = B b F m S * F B S m m
PIL-PUBR2 = B b F m S * F B m S M
PIL-PROP = B b F M S * F B S m M

PI-PUBR1 = B b m S F * f B f f S m m
PI-PUBR2 = B b m S F * f B f f m S m
PI-PROP1 = B b M S F * f B f F m S M
PI-PROP2 = B b M S F * f B f F S m M

AACT-PUBR1 = m F b m S * F m S b b B
AACT-PUBR2 = m F b m S * m F S b b B
AACT-PROP1 = M B S m F * S M b F b B
AACT-PROP2 = M B S F m * S M b F b B

3. Maps generated using the AT/ACE data from Sefton et al(1990). The maps for each gene were extracted from the published maps. The maps were found to contain certain problems in the number of cut sites used to generate the single digests. Two minor amendments were made to the maps. Nevertheless, it was considered that the maps shown represented near-optimal maps.

```
AT1 = S B s m b f * f B f f f f s S f m F m m M
AT2 = S B s m b f * f B f f f f s S m f F m m M

ACE1 = M F f f B S * f m b F S m B m M
ACE2 = M F f f B S * m f b F S m B m M
ACE3 = M F f f B S * f m b F m S B m M
ACE4 = M F f f B S * m f b F m S B m M
```

Appendix H - Description of Hybrid Genetic Algorithm versions 0 - 3

The original version of the Hybrid Genetic Algorithm (HGA) is described below. The changes made in the three updates are highlighted.

HGA VERSION 0 (HGAv0)

Population size	:	20 - 100
No. of trials	:	0 - 500
Operators	:	order swap side swap
Operator rates	:	fixed (100%)
Reproduction technique	:	generational replacement generational replacement with elitism steady state
Initial Population	:	generated at random - possibility of duplicates
Evaluation Function	:	evaluation 1, evaluation 2

HGA VERSION 1 (HGAv1) - the case swap operator was included and the operator rates were changed to become variable. 10% variation in fragment lengths was allowed when evaluating on the fit of the single digest data in evaluation 1. Evaluation 2 was modified to adjust the threshold values for the number of double digests expected and observed. Evaluation 3 was introduced.

HGA VERSION 2 (HGAv2) - the initial population was checked to ensure that all maps were different. The "steady-state" reproduction technique was modified to become "steady-state without duplicates".

HGA VERSION 3 (HGAv3) - the "steady-state without duplicates" technique was modified to become "steady-state without duplicate templates". Evaluation 3 was modified to penalise maps which contained zero intervals over the probe.

Appendix I - Glossary of Biological Terms

Autoradiography - A technique used to highlight the presence of a radioactively labelled substance, eg a gene probe.

Base - A chemical component of DNA. One of four different types namely, Adenine(A), Guanine(G), Cytosine(C) and Thymine(T).

Base Pair - The association of two bases, one from each strand of DNA in a chromosome (as illustrated in figure 2.1). Adenine(A) always pairs with Thymine(T) and Cytosine(C) always pairs with Guanine(G).

Chromosome - Each cell in the human body (apart from the sex cells) contains 46 chromosomes organised into 23 pairs. Chromosomes are made from DNA and contain the 100,000 or so genes estimated to be present for the human.

Cloned DNA - A portion of DNA that has been reproduced artificially (or "cloned") in the laboratory. Normally this would represent a very small fraction of the total DNA of a cell.

Complete cut site - A specific site/position on the DNA that is always cut by the restriction enzyme.

DNA (Deoxyribonucleic acid) - A chemical consisting of two strands loosely joined together as illustrated in Figure 2.1. DNA is made up from four different bases namely Adenine(A), Thymine(T), Cytosine(C) and Guanine(G). Each set of three (triplet) bases represents the genetic code for one amino acid. (Amino acids are the primary building blocks for proteins).

Double digest fragment - A fragment of DNA that has been produced using two restriction enzymes. One or other of the restriction enzymes has cut at each end of the fragment.

Expected data - Given either a single gene map or a multi-gene map with the distances between the cut sites shown, the number and lengths of the fragments containing a particular gene can be calculated. As this data is the data derived from a map, it is referred to as the "expected data". Due to experimental limitations, the data expected from a map contains errors and differs from the data observed experimentally.

Gel electrophoresis - A technique used to separate fragments of DNA according to their weight. The DNA fragments are placed on a column of gel and an electric current is applied. The fragments move down the gel at a speed inversely proportional to their weight. The size of the DNA fragments in terms of number of bases can be calculated using a calibrated column of gel.

Gene - A specific sequence of DNA bases that carries the information for making a particular protein or for a particular function. Genes always occur at the same position on a chromosome and lie in a linear order (or sequence). The length of genes varies. It has been estimated that there are 100,000 genes in the human.

Gene probe - A sequence of bases, radioactively labelled, which will bind to a particular gene. Gene probes are used to highlight the gene of interest.

Kilobase pairs (Kb) - The unit of measurement for DNA fragment length. 1 Kb = 1000 base pairs.

Map assembly - The process of determining the number and sequence (or order) of genes and restriction enzyme cut sites along the DNA from the observed data.

Methylated site - Refers to the chemical state of the DNA bases. If the bases comprising the recognition site of a restriction enzyme are methylated, the restriction enzyme is blocked from cutting the DNA at that recognition site. (The opposite of unmethylated site.)

Multi-gene map - A diagram of a section of DNA showing the sequence (or order) of more than one gene and the number, sequence and position of restriction enzyme cut sites. (Multi-gene maps were produced by this thesis by merging two single gene maps together.)

Observed data - The data obtained by the geneticist from conducting single and double digest experiments. (Also referred to as Experimental Data.)

Optimal map - An optimal map generated by the Hybrid Genetic Algorithm is one which achieves the best score (0) as allocated by the evaluation function.

Overlapping fragments - Two fragments overlap when they share an identical sequence of bases. For example, all fragments that contain a particular gene must overlap each other at the position of the gene.

Partial cut site - A specific site/position on the DNA that is cut in some cases by a restriction enzyme and not cut in others.

Partial digest fragment - A fragment of DNA produced by the presence of a partial cut site.

Recognition site - The sequence of bases that a restriction enzyme recognises on the DNA. The length of a recognition site varies. Different restriction enzymes recognise different sequences.

Restriction enzyme - Substance that cleaves the DNA when it recognises a specific sequence of bases (or recognition site). There are many different types of restriction enzyme that recognise different sequences of bases.

Single digest fragment - A fragment of DNA that has been produced using a single restriction enzyme. Each end of the fragment has been cut at either end by the restriction enzyme.

Single gene map - A diagram of a section of DNA showing the number, sequence (or order) and position of restriction enzyme cut sites around one gene. (Single gene maps were produced by this thesis using a Hybrid Genetic Algorithm.)

Southern blotting - A technique used to transfer DNA fragments from the gels used in Gel Electrophoresis to a nitrocellulose filter to allow the DNA fragments to be manipulated.

Template - A sequence of restriction enzyme cut sites around a gene, ignoring the "nature" of the cut sites ie whether or not the cut sites are complete or partial.

Total DNA - All the DNA (comprising the 23 pairs of chromosomes found in all but the sex cells) contained in a cell. A sample of total DNA used in the single and double digest experiments contains the total DNA from many cells.

Unmethylated site - Refers to the chemical state of the DNA bases. If the bases comprising the recognition site of a restriction enzyme are unmethylated on the DNA, the restriction enzyme can cut the DNA. (The opposite of methylated site.)

Weak fragment - A possible fragment of DNA that appears as a faint band on the gel columns. It may be a proper fragment but it may also be due to either smudging on the gel or to the probe attaching to a related gene. Weak fragments tend not to be used by the geneticist when assembling a map.

REFERENCES

Alander, J.T. 1992. On Optimal Population Size of Genetic Algorithms. Proceedings of CompEuro 92. IEEE Computer Society Press, 65-70.

Antonisse, J. 1989. A New Interpretation of Schema Notation That Overturns The Binary Encoding Constraint. In: Proceedings of the Third International Conference on GAs. San Mateo, CA:Morgan Kaufmann Publishers, 86-91.

Arnold, S.F. 1990. Mathematical Statistics. Prentice-Hall International, Inc.

Back, T. and Schwefel, H. 1993. An Overview of Evolutionary Algorithms for Parameter Optimization. Evolutionary Computation: Volume 1 no.1. Massachusetts Institute of Technology, 1-23.

Bagchi, S., Uckun, S., Miyabe, Y., et al. 1991. Exploring Problem-Specific Recombination Operators for Job Shop Scheduling. In: Proceedings of the Fourth International Conference on GAs. San Mateo,CA: Morgan Kaufmann Publishers, 10-17.

Beasley, D., Bull, D.R., Martin, R.R. 1993. A Sequential Niche Technique for Multimodal Function Optimisation. Evolutionary Computation: Volume 1 no. 2, 101-125.

Bethke, A.D. 1981. Genetic Algorithms as Function Optimisers. (Doctoral dissertation, University of Michegan), Dissertation Abstracts International 41(9), 3503B (University Microfilms No. 8106101).

Billingsley, G.D., Walter, M.A., Hammond, G.L., et al.

1993. Physical Mapping of Four Serpin Genes: Alpha1-Antitrypsin, Alpha1-Antichymotrpsin, Corticosteroid-binding Globulin, and Protein C Inhibitor, within a 280-kb Region on Chromosome 14q32.1. American Journal of Human Genetics, 343-353.

Bosworth, J., Foo, N., Zeigler, B.P. 1972. Comparison of Genetic Algorithms with Conjugate Gradient Methods. (CR-2093) Washington, DC: National Aeronautics and Space Administration.

Brindle A. 1981. Genetic Algorithms for Function Optimisation. Unpublished doctoral dissertation. University of Alberta, Edmonton.

Cinkosky, M.J., Fickett, J.W. 1992. SIGMA User Manual version 0.70, September 1992, Human Genome Information Resource, Theoretical Biology and Biophsics Group, Center for Human Genome Studies, Los Alamos National Laboratory.

Cook, S.A. 1971. The Complexity of Theorem Proving Procedures. In: Proc. 3rd ACM Symp. on the Theory of Computing, ACM, 151-158.

Cox, D.W., Walter, M.A., Coulson, S.E., et al.1987. Regional localisation of alpha-1-antichymotrpsin(AACT) to 14q32.1 and its proximity to alpha-1-antitrypsin(PI) by pulsed field gel electrophoresis. Cytogenet. Cell Genetics, 46:600.

Davis, L. 1985. Job Shop Scheduling with Genetic Algorithms. In: Proceedings of an International Conference on GAs and their Applications, 136-140.

Davis, L. 1989. Adapting Operator Probabilities in Genetic

Algorithms. In: Proceedings of the Third International Conference on GAs. San Mateo, CA:Morgan Kaufmann Publishers, 61-69.

Davis, L. 1991a. The Handbook of Genetic Algorithms. Van Nostrand Reinhold Publishers.

Davis, L. 1991b. Hybridisation and Numerical Representation. In: The Handbook of Genetic Algorithms, Van Nostrand Reinhold Publishers, 61-71.

De Jong, K.A. 1975. An Analysis of the Behaviour of a Class of Genetic Adaptive Systems. (Doctoral Dissertation, Department of Computer and Communication Sciences, University of Michigan). Dissertation Abstracts International 36(10),5140B. (University Microfilms No. 76-9381).

De Jong, K.A., Spears, W.M. 1989. Using Genetic Algorithms to Solve NP-Complete Problems. In: Proceedings of the Third International Conference on Genetic Algorithms, edited by J.D.Schaffer, 124-133.

Deb K., Goldberg, D.E.1989. An Investigation of Niche and Species Formation in Genetic Function Optimisation. In: Proceedings of the Third International Conference on Genetic Algorithms, edited by J.D.Schaffer, 42-50.

Eshelman, L.J., Schaffer, J.D.1993. Real-Coded Genetic Algorithms and Interval-Schemata. In: Foundations of Genetic Algorithms 2, edited L.Darrell Whitley. Morgan Kaufmann Publishers, San Mateo, California, 187-202.

Fickett, J.W., Cinkosky, M.J. 1992. Optimizing Maps to Fit Experimental Data. Abstracts of Papers Presented at the 1992 Meeting on Genome Mapping and Sequencing, Cold Harbor

Laboratory, New York. Published by Cold Spring Harbor Press.

Fickett, J.W., Cinkosky, M.J. 1993. A Genetic Algorithm for Assembling Chromosome Physical Maps. In: Proceedings of the Second International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis, edited by H.A.Lim, J.W.Fickett, C.R.Cantor, R.J.Robbins. World Scientific, 273-286..ls2

Fogel, L.J., Owens, A.J., Walsh, M.J. 1966. Artificial intelligence through simulated evolution. New York: John Wiley.

Fox, B.R., McMahon, M.B. 1991. Genetic Operators for Sequencing Problems. Foundations of GAs, edited by GR.Rawlins, San Mateo, CA: Morgan Kaufmann Publishers.

Fraser, D.A.S. 1976. Probability and Statistics, Theory and Applications, Duxbury Press, Massachusetts.

Garey, M.R., Johnson,D.S., 1979. Computers and Intractability: A Guide to NP-completeness. San Francisco: W.H.Freeman & Company.

Goldberg, D.E. 1983. Computer-aided Gas Pipeline Operation using Genetic Algorithms and Rule Learning. (Doctoral Dissertation, University of Michegan), Dissertation Abstracts International, 44(10), 3174B (University Microfilms No. 8402282).

Goldberg, D.E. 1989. Genetic Algorithms in Search, Optimisation and Machine Learning. Addison Wesley.

Goldberg, D.E. 1994. Genetic Algorithms and Evolutionary Algorithms Come of Age. Communications of the ACM, Vol 37, no. 3, 113-119.

Goldberg, D.E., Lingle, R. 1985. Alleles, Loci, and the Travelling Salesman Problem. In: Proceedings of an International Conference on GAs and their Applications, 154-159.

Goldberg, D.E., Richardson, J. 1987. Genetic Algorithms with Sharing for Multimodal Functions. In: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, 41-49.

Goldberg, D.E., Korb,B., Deb, K. 1989. Messy Genetic Algorithms: Motivation, analysis, and first results. Complex Systems 3, 493-530.

Goldberg, D.E., Korb, B., Deb, K. 1990. Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale. Complex Systems 4, 415-444.

Goldberg, D.E., Deb, K., Thierens, D. 1993. Toward a Better Understanding of Mixing in Genetic Algorithms. **Journal of Soc. Instr. Contr. Eng.** 32, part 1, 10-16.

Grefenstette, J.J., Gopal, R., Rosmaita, B.J., et al.1985. Genetic Algorithms for the Travelling Salesman Problems. In: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, 160-168.

Grefenstette, J.J. 1986. Optimisation of Control Parameters for GAs. IEEE Transactions on Systems, Man and Cybernetics SMC-16, 1, 122-128.

Grefenstette J.J. 1987. Incorporating Problem Specific

Knowledge into Genetic Algorithms. In: Genetic Algorithms and Simulated Annealing, edited by L. Davis. Morgan Kaufmann Publishers, Inc., Los Altos, CA 1987.

Hoffman, K. 1991. Personal correspondence.

Holland, J.H. 1973. Genetic algorithms and the optimal allocation of trials. SIAM Journal of Computing, 2(2), 88-105.

Holland, J.H. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michegan Press.

Hunter, L. 1993. Artificial Intelligence and Molecular Biology. The MIT Press.

Karp, R.M. 1972. Reducibility among Combinatorial Problems. Complexity of Computer Computations, edited by R.E.Miller, J.W.Thatcher. New York: Plenum Press, 85-104.

Kearney, P., Kelsey, G., Abeliovich, D. et al. 1987. Physical linkage between members of the alpha-1-antitrypsin gene family. Cytogenet. Cell Genetics 46:637.

Michalewicz, Z. 1993. A Hierarchy of Evolution Programs: An Experimental Study. Evolutionary Computation, Volume 1 no.1 ps:51-76, Massachusetts Institute of Technology.

Moscato, P., Norman, M.G. 1992. A "Memetic" Approach for the Travelling Salesman Problem - Implementation of a Computational Ecology for Combinatorial Optimisation on Message-Parsing Systems. In: Proceedings of the International Conference on Parallel Computing and Transputer Applications. IOS Press (Amsterdam).

Oliver, I.M., Smith, D.J., Holland, J.R.C. 1987. A Study of Permutation Crossover Operators on the Travelling Salesman Problem. In: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on GAs, 224-230.

Open University 1981. OU Tech/Math TM 361 16 Graphs, Networks and Design. OU Press 1981.

Oppenheim, A.N. 1966. Questionnaire Design and Attitude Measurement. Heinmann Educational Books Ltd.

Pearson, W.R. 1982. Automatic Construction of Restriction Site Maps. Nucleic Acids Research, Vol 10, No. 1, 1982.

Radcliffe, N.J. 1991a. Equivalence Class Analysis of Genetic Algorithms. Complex Systems, 5, 183-205.

Radcliffe, N.J. 1991b. Forma Analysis and Random Respectful Recombination. In: Proceedings of the Fourth International Conference on Genetic Algorithms, ed. R.K.Belew and L.B.Booker, Morgan Kaufmann Publishers, 222-229.

Radcliffe, N.J. 1993. Genetic Set Recombination. In: Foundations of Genetic Algorithms 2, edited L.Darrell Whitley. Morgan Kaufmann Publishers, San Mateo, California, 203-220.

Radcliffe, N.J. 1994. The Algebra of Genetic Algorithms. Annals of Mathematics and Artificial Intelligence 10, 339-384.

Radcliffe, N.J., George, F.A.W. 1993. A Study in Set Recombination. In: Proceedings of the Fifth International Conference on Genetic Algorithms, edited by S.Forrest. Morgan Kaufmann Publishers, 23-30.

Radcliffe, N.J., Surry, P.D. 1994a. Fitness Variance of Formae and Performance Prediction. To appear in: Foundations of Genetic Algorithms 3, edited by L.D.Whitley and M.Vose. Morgan Kaufmann Publishers, San Mateo, California, 628-637.

Radcliffe, N.J., Surry, P.D. 1994b. Formal Memetic Algorithms. In: Evolutionary Computing: AISB Workshop. Springer Verlag, Lecture Notes in Computer Science 865, edited by T.C.Fogarty, 1-16.

Rechenberg, I. 1965. Cybernetic solution path of an experimental problem. Roy. Aircr. Establ. libr. transl. 1122. Hants, UK: Farnborough.

Reeves, C.R. (Editor.) 1993a. Modern Heuristic Techniques for Combinatorial Problems. Blackwell Scientific Publications, Oxford.

Reeves, C.R. 1993b. Using Genetic Algorithms with Small Populations. In: Proceedings of the Fifth International Conference on Genetic Algorithms, edited by S.Forrest. Morgan Kaufmann Publishers, 92-99.

Rich, E., Knight, K. 1991. Artificial Intelligence. 2nd edition, McGraw Hill, 44.

Sefton, L., Kelsey, G., Kearney, P., et al. 1990. A Physical Map of the Human PI and AACT Genes. Genomics 7, 382-388.

Schaffer, J.D., Caruana, L.J., Eshelman, L.J. et al. 1989. A Study of Control Parameters Affecting Online Performance of GAs for Function Optimisation. In: Proceedings of the

Third International Conference on GAs, San Mateo, CA:Morgan Kaufmann Publishers, 51-60.

Shifman, M., Nadkarni, P., Miller, P. 1992. Interactive, Graphical, Computer-Based Tools for Storing and Constructing Pulse Field Gel Maps. Abstracts of Papers Presented at the 1992 Meeting on Genome Mapping and Sequencing, Cold Harbor Laboratory, New York, Published by Cold Spring Harbor Press.

Silver, E.A., Vidal, R.V., de Werra, D. 1980. A tutorial on heuristic methods. European Journal of Operational Research, 153-162.

Southern, E.M. 1975. Detection of specific sequences among DNA fragments separated by gel electrophoresis. J. Mol. Biol. 98:503-517.

Spears, W.M. 1994. Simple Subpopulation Schemes. In: Proceedings of the Third Annual Conference on Evolutionary Programming, San Diego, CA, edited by D.B.Fogel, W.Atmar, IEEE Press.

Starkweather, T., McDaniel, S., Mathias, K. et al. 1991. A Comparison of Genetic Sequencing Operators. In: Proceedings of the Fourth International Conference on Genetic Algorithms, ed. R.K.Belew and L.B.Booker, Morgan Kaufmann Publishers, 69-76.

Stefik, M. 1978. Inferring DNA Structures from Segmentation Data. Artificial Intelligence, Vol 11, 85-114.

Syswerda, G. 1989. Uniform Crossover in GAs.In: Proceedings of the Third International Conference on GAs, edited by J.David Schaffer. San Mateo, California: Morgan Kaufmann Publishers.

Syswerda, G. 1991. Schedule Optimisation Using Genetic Algorithms. In:The Handbook of Genetic Algorithms, edited by L.Davis, Van Nostrand Reinhold Publishers, New York.

Vose, M.D., Liepins, G.E. 1991. Schema Disruption. In: Proceedings of the Fourth International Conference on Genetic Algorithms, ed. R.K.Belew and L.B.Booker, Morgan Kaufmann Publishers, 237-243.

Walker, J.D., File, P.E., Miller, C.J. et al 1994. Building DNA Maps: A Genetic Algorithm Based Approach. In: Advances in Molecular Bioinformatics, edited by S.Schulze-Kremer. IOS Press, 179-199.

Whitley, D. 1988. GENITOR : a different GA. In: Proceedings of the Rocky Mountain Conference on AI. Denver, Colorado.

Whitley, D., Starkweather, T., Fuquay, D. 1989. Scheduling Problems and Travelling Salesman: the Genetic Edge Recombination Operator. In: Proceedings of the Third International Conference on GAs, San Mateo,CA: Morgan Kaufmann Publishers, 133-140.

Whitley, D., Starkweather, T., Shaner, D. 1991. The Travelling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination. In: The Handbook of Genetic Algorithms, edited by L. Davis, Van Nostrand Reinhold Publishers, New York.

Whitley, D. 1993. Introduction to Foundations of Genetic Algorithms 2, edited by L.Darrell Whitley, Morgan Kaufmann Publishers, San Mateo, California.

Wright,L., Lichter,J., Shifman,M. et al. 1992. An Interactive Computer Tool to Help Interpret Restriction Mapping Data. Abstracts of Papers Presented at the 1992 Meeting on Genome Mapping and Sequencing, Cold Harbor Laboratory, New York, Published by Cold Spring Harbor Press.

Zanakis, S., Evans, J., Vazacopoulos, A., 1989. Heuristic Methods and Applications : A Categorized Survey. European Journal of Operational Research 43, Elsevier Science Publishers B.V.(North-Holland Press), 88-110.

PUBLICATIONS

GENIE: A Genetic Algorithm Application to handle Noisy Data in the Biological Domain.

J.D.Walker, P.E.File, C.J.Miller, W.B.Samson
Department of Mathematical and Computer Sciences
Dundee Institute of Technology
Bell Street
DUNDEE DD1 1HG

ABSTRACT

Genetic Algorithms (GAs) have generated much interest in recent years in areas as diverse as machine learning, machine vision, function optimisation and NP-hard problems.

This paper describes the application of a genetic algorithm to a problem in the biological domain. A computer support tool, "GENIE", is being developed to assist geneticists with a problem in their field - the building of restriction "maps" from the results of partial digest experiments. Building restriction maps is a time-consuming and lengthy activity which is based on noisy data and relies on human judgement. The search space of feasible solutions is very large. It is proposed that this problem solving exercise would benefit from the application of artificial intelligence techniques.

The procedure for building restriction enzyme maps is introduced in section 1. There are several aspects of map assembly which make it an interesting problem to study and these are outlined in section 2. The way in which the problem is being tackled, through the development of a computer support tool and by incorporating artificial intelligence techniques into the tool, is discussed in section 3. The paper is summarised in section 4, and section 5 highlights future developments.

1. MAP BUILDING PROCEDURES

Geneticists worldwide are attempting to identify and isolate genes and to generate a "map" for the human showing the position and function of each gene on each chromosome (known as genetic mapping).

A current technique which has had a significant impact on genetic mapping is the use of restriction enzymes (REs). REs are used to cut up chromosomes into fragments. There is a wide range of these enzymes which have different properties and will cut at particular sites on the chromosomes. By observing the fragments produced and using already known genetic markers, geneticists can produce restriction maps for chromosomes showing the sites where the enzymes cut and the position of known markers. Using this technique allows regions of DNA to be characterised.

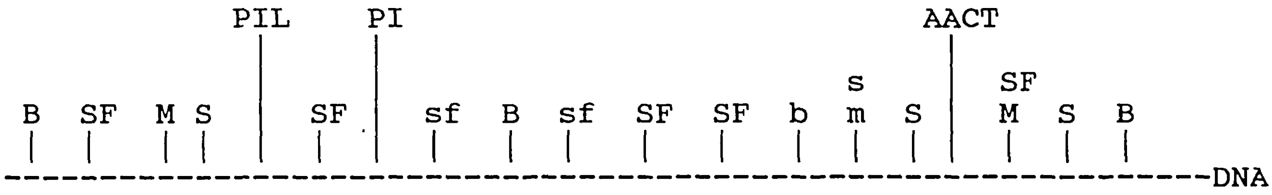
An example of a RE map is shown in figure 1.1 and the data from which it was generated is shown in table 1.1 (Sefton (1990)). The numbers in table 1.1 represent the lengths of the DNA fragments. Each fragment contains a genetic marker and has been cut at either end by the RE shown.

When one RE is used, the fragments obtained are called "single digests" and each end of the fragment has been cut by that RE. When two REs are used, the fragments obtained are called "double digests". Here, each end of the fragment has been cut by one or other of the REs.

RE cut sites are said to be either "complete" or "partial". A complete cut site is a site on the DNA which is always cut by the RE. A partial cut site is a site which is cut on some pieces of DNA, and left intact on others. The occurence of partial cut sites leads to the presence of long fragments. Determining the position of the genetic markers in relation to one another relies on the presence of such fragments. Long fragments are likely to contain two or more markers if the markers are adjacent.

When all fragments (single and double digests), present in table 1.1 are combined, taking into account the error present in the fragment lengths, the choice of RE cutting in the double digests and the nature of the cut sites, a RE map is generated which fits the data best. The map reveals the ordering of the genetic markers, the cut sites of the restriction enzymes, and whether or not the cut sites are partial or complete. The map assembled using the data in table 1.1 is shown in figure 1.1. The distance apart of the markers can be calculated.

Figure 1.1 - RE Map assembled from the data in table 1.1 (not to scale) showing the ordering of the RE cut sites and gentic markers.



NOTES

1. PIL, PI, AACT - genetic markers
2. B, SF, M, S - REs
3. REs shown in capital letters represent complete cut sites, lower case indicates partial cut sites.

Table 1.1 - Table Containing Experimental Data (Sefton (1990)).

NOTES

1. PIL, PI, AACT - genetic markers
2. B, SF, M, S - REs
3. Fragments contained in brackets indicate "weak" fragments.

B			M			S			SF		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	355	350	350	350	255	255	70	165	[190]	[195]
225	225	275	260	260	80				[10]	[135]	[175]
	[65]	230		[180]						80	135
										65	
										10	

B+M			B+S		
PIL	PI	AACT	PIL	PI	AACT
250	250	230	125	125	70
225	225	105		[65]	
130	130				
	[65]				

B+SF			M+S		
PIL	PI	AACT	PIL	PI	AACT
165	[190]	135	255	255	70
[10]	50	105		[180]	30
	10				

M+SF			S+SF		
PIL	PI	AACT	PIL	PI	AACT
85	80	135		[130]	30
	65	85	70	80	
[10]	10		(10)	65	
				10	

2. MAP BUILDING CHARACTERISTICS

There are several aspects of map building which make it an interesting activity to study.

1. Maps do not fit perfectly.

The data used to build a map is noisy, due to experimental limitations, therefore no map ever fits perfectly. "Weak" fragments are sometimes obtained. These tend to be ignored when a map is assembled, but should in theory fit into the solution. A "good" map solution is one in which the data, including the weak fragments, fits well.

At present, there is a lack of criteria by which the correctness of maps can be assessed. Work is in progress to identify and establish appropriate criteria and these will enable the quality of existing maps to be evaluated and the quality of potential maps to be predicted.

2. Fragments overlap.

All fragments obtained for a particular marker must overlap the region of the marker. Many orderings of fragments are possible and it is difficult to determine where the cut sites are.

3. Numerous solutions are possible.

Due to the overlapping nature of the fragments and the error in the fragment lengths, many map solutions can be generated from the same data.

3. SOLVING THE PROBLEMS

The problem solving activity of RE map assembly is being tackled in two phases.

A computer support tool is being developed the aim of which is to make map building easier. Currently, maps are assembled by hand using pencil and paper. Developing the tool is the first step in automating the process.

Due to the characteristics of the map building process described in section 2, it is a relevant activity to study from an artificial intelligence point of view and various search techniques are being examined and assessed for their suitability for implementation in the tool. Subsequently, the tool will be used as a test vehicle to implement promising techniques and automate map assembly.

Map assembly is a sequencing problem which requires search to find a solution from a large problem space. There are many types of search techniques described in the literature (Korf 1988). GAs have been applied to computationally complex sequencing problems such as the travelling salesman's problem (Goldberg (1985), Grefenstette (1985)) and job shop scheduling (Davis (1985), (Syswerda (1991)) to see if they offer advantages over traditional search techniques.

Due to the characteristics of map assembly, it is proposed to investigate the suitability of using a GA as a search technique. Properties of genetic algorithms are being examined and the way in which these could be exploited for this particular application are being considered. The broader implications arising as a result of this work on the performance and limitations of genetic algorithms will be determined.

4. SUMMARY

This paper has outlined the problem solving activity of map building and has described the approach that is being taken to automate the process via the incorporation of artificial intelligence techniques.

5. FUTURE DEVELOPMENTS

It is likely that in order to apply artificial intelligence techniques such as GAs to map assembly, the techniques themselves will need to be developed in some novel way. The maps assembled by the advanced tool will be evaluated both by geneticists and by comparison with existing validated maps which will enable the success of the techniques in generating "best fit" maps to be determined. As a result, new insight will be gained into these techniques and it is believed these will be applicable to a broader spectrum of problems.

REFERENCES

Davis, L. (1985) "Job Shop Scheduling with Genetic Algorithms", Proceedings of an International Conference on Genetic Algorithms and their Applications, 136-140.

Goldberg, D.E., Lingle, R. (1985) "Alleles, Loci and the Traveling Salesman Problem", Proceedings of an International Conference on Genetic Algorithms and their Applications, 154-159.

Grefenstette, J.J., Gopal, R., Rosmaita, B.J., Van Gucht, D. (1985) "Genetic Algorithms for the Traveling Salesman Problems" Proceedings of an International Conference on Genetic Algorithms and their Applications, 160-168.

Korf, R.E. (1988) "Search: A Survey of Recent Results", Exploring AI, H.E Shrobe ed., Morgan Kaufman Publishers Inc, San Mateo, CA 1988.

Sefton, L., Kelsey, G., Kearney, P., Povey, S., Wolfe, J. (1990) "A Physical Map of the Human PI and AACT Genes", Genomics 7, 382-388.

Syswerda, G. (1991) "Schedule Optimisation Using Genetic Algorithms", The Handbook of Genetic Algorithms, L.Davis ed., Van Nostrand Reinhold Publishers.

THE GENIE PROJECT - A Genetic Algorithm Application to a Sequencing Problem in the Biological Domain.

J.D.Walker, P.E.File, C.J.Miller, W.B.Samson
Department of Mathematical and Computer Sciences
Dundee Institute of Technology
Bell Street
DUNDEE DD1 1HG

ABSTRACT

This paper describes the current development and implementation of a form of genetic algorithm (GA) suitable for tackling a complex sequencing problem in the biological domain - the building of restriction "maps" from the results of partial digest experiments. Building restriction maps is a time-consuming and lengthy activity which relies on human judgement of inexact data.

The paper is organised into the following sections. The procedure for building restriction enzyme maps is described in section 2. There are several aspects of map assembly which make it a relevant problem to study from a GA point of view and these are outlined in section 3. The GENIE project is an ongoing project and the way in which the problem is being tackled by developing a GA and the implementation issues are discussed in section 4. Preliminary results are shown in section 5, the paper is summarised in section 6 and section 7 highlights future developments.

1. INTRODUCTION

GAs are search procedures based on the mechanics and analogy of natural selection. They were introduced by Holland(1975) - for an introduction to the subject refer to Goldberg(1989). Map assembly is an example of a difficult sequencing problem which requires some form of search to find a good solution from a large problem space of feasible solutions. Traditional GAs are not effective for sequencing problems as illegal solutions can be generated.

Two of the goals of the GENIE project are to develop some form of GA which can successfully tackle sequencing problems and to incorporate an evaluation function based on human assessment of subjective and error prone data. In order to meet these goals, the traditional GA has been modified in several ways to produce a hybrid GA.

The concept of a hybrid GA has been suggested by a number of authors(Bethke(1981); Bosworth (1972); Goldberg(1983); Davis (1991)). A hybrid GA is a GA which incorporates problem specific information or various search techniques. Research conducted in this area has shown a hybrid GA approach to be promising for combinatorial optimisation problems such as the Travelling Salesman Problem (Goldberg(1985), Grefenstette (1985), Oliver(1987), Whitley (1989,1991)) and scheduling problems (Davis(1985), Syswerda (1991)).

In the GENIE project, a hybrid GA approach is being applied to find a good solution to the map assembly problem.

2. MAP BUILDING PROCEDURES

Geneticists worldwide are

attempting to identify and isolate genes and to generate a "map" showing the position and function of each gene on the human DNA .

A current technique which has had a significant impact on genetic mapping is the use of restriction enzymes (REs). REs are used to cut up DNA into fragments. There is a wide range of these enzymes which have different properties and will cut at particular sites. By observing the fragments produced and using already known probes (genetic markers), geneticists can produce restriction maps showing the sites where the enzymes cut and the position of known probes. Using this technique allows regions of DNA to be characterised.

An example of a RE map is shown in figure 2.1 and the data from which it was generated is shown in table 2.1 (Sefton (1990)). The numbers in table 2.1 represent the lengths of the DNA fragments. Each fragment contains a probe and has been cut at either end by the RE shown.

When one RE is used, the fragments obtained are called "single digests" and each end of the fragment has been cut by that RE. When two REs are used, the fragments obtained are called "double digests". Here, each end of the fragment has been cut by one or other of the REs.

RE cut sites are said to be either "complete" or "partial". A complete cut site is a site on the DNA which is always cut by the RE. A partial cut site is a site which is cut on some pieces of DNA, and left intact on others. The occurence of partial cut sites leads to the presence of long fragments. Determining the position of the probes in relation to one another relies on the presence of such fragments. Long fragments are likely to contain two or more probes if the probes are adjacent.

When all fragments present in table 2.1 are combined, taking into account the error present in the lengths, the choice of RE cutting in the double digests and the nature of the cut sites, a RE map is generated which fits the data best. The map reveals the ordering of the probes, the cut sites of the restriction enzymes, and whether or not the cut sites are partial or complete. The map assembled using the data in table 2.1 is shown in figure 2.1.

3. MAP BUILDING CHARACTERISTICS

Map building is essentially an ordering problem - the map is linear and after the number and types of cut sites have been determined, they must then be placed in the correct sequence. There are several aspects of map building which make it a difficult activity.

3.1 Maps do not fit perfectly.

No map ever fits perfectly as there are errors present in the amount and quality of the data used to build a map, due to experimental limitations."Weak" fragments are sometimes obtained. Most weak fragments should fit into a map but as some are due to error, they tend to be ignored when a map is assembled but should in theory fit into the solution. A

Figure 2.1 - RE Map assembled from the data in table 2.1 (not to scale) showing the ordering of the RE cut sites and probes.

```
            PIL    PI                           AACT
             |     |                        s     | SF
 B  SF  M S  |  SF |  sf  B  sf  SF  SF  b  m  S  |  M  S  B
 |   |  | |  |  |  |  |   |  |   |   |   |  |  |  |  |  |  |
-------------*-----*------------------------------*-------------DNA
```

NOTES

1. PIL, PI, AACT - probes
2. B, SF, M, S - REs
3. REs shown in capital letters represent complete cut sites, lower case indicates partial cut sites.

Table 2.1 - Table containing Experimental Data (Sefton (1990)) used to generate the map shown in figure 2.1.

SINGLE DIGEST DATA

B			M			S			SF		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	355	350	350	350	255	255	70	165	[190]	[195]
225	225	275	260	260	80				[10]	[135]	[175]
	[65]	230		[180]						80	135
										65	
										10	

DOUBLE DIGEST DATA

B+M			B+S			B+SF		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	230	125	125	70	165	[190]	135
225	225	105		[65]		[10]	50	105
130	130						10	
	[65]							

M+S			M+SF			S+SF		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
255	255	70	85	80	135		[130]	30
	[180]	30		65	85	70	80	
			[10]	10		(10)	65	
							10	

NOTES

1. PIL, PI, AACT - probes
2. B, SF, M, S - REs
3. Fragments contained in brackets indicate "weak" fragments.

"good" map is one in which the data, including most of the weak fragments, fits well.

There has been a lack of criteria by which the correctness of maps can be assessed. As part of the GENIE project, appropriate criteria have been identified and incorporated into an evaluation function. This is discussed further in section 4.1.3.

3.2 Fragments overlap.

All fragments obtained for a particular probe must overlap the region of the probe. Many orderings of fragments are possible and it is difficult to determine where the cut sites are.

3.3 More than one solution may be possible.

Due to the overlapping nature of the fragments and the error in the fragment lengths, it is possible that many feasible map solutions may be generated from the same data.

4. A GENETIC ALGORITHM APPROACH TO SOLVING THE PROBLEM

The traditional GA has been modified to handle the map assembly problem. The representation and optimisation techniques employed by the expert are used to ensure that the domain knowledge embodied in the encoding is preserved. The GA operators are tailored to apply to the new representation and include domain-based heuristics, and an evaluation function which is based on subjective assessment has been developed. Other problem specific knowledge has guided the development of the overall algorithm.

The modifications made to the traditional GA may appear to be at odds with "pure" GA research, as one of the main aims of GA research has been to develop an algorithm that is robust across a variety of problem domains and operates without problem specific information. It is the intention here to see if the power of the traditional GA can be harnessed and used as the basis of an effective algorithm for a real-world complex problem. Although the traditional GA is robust across a wide range of problems, it is unlikely to be the best algorithm to use for any specific application.

4.1 IMPLEMENTATION ISSUES

A "front-end" menuing system has been developed for the GA which allows for various features to be selected. This has facilitated the implementation of the modified GA and allowed the effect of changing particular features to be observed.

There are several areas which have been addressed during the development of the modified GA and these are discussed in the following sections.

4.1.1 REPRESENTING THE PROBLEM

Traditional GAs generally represent chromosomes as binary vectors. However, having considered several options, it was decided that in the modified GA, the chromosome syntax should be changed to reflect the problem as shown in figure 2.1.

4.1.2 DEVELOPING A SET OF GENETIC OPERATORS

Applying the traditional genetic operators to map assembly has similiar difficulties with applying them to other sequencing problems such as the Travelling Salesman Problem (TSP). In the case of the TSP, there are constraints placed on the symbol string that represents a tour of the cities, in that no city can appear more than once. The traditional recombination operators rearrange symbols on a chromosome independently of each other. When solutions are coded as sequences and the traditional operators applied, cities can appear more than once or not at all.

When developing operators for map assembly, taking into account previous work (Goldberg (1989), Fox and McMahon(1991)), a way of breaking up the chromosomes that was natural for the problem was sought. The traditional operators have been developed to meet the constraints on the chromosomes while preserving the motivating principles behind these operators. Crossover provides an opportunity for the best attributes of both parent strings to be incorporated into the offspring. Mutation is a mechanism for introducing necessary attributes into an individual when those attributes do not already exist within the current population.

4.1.3 DEVELOPING THE EVALUATION FUNCTION

The way in which chromosomes are assessed is critical to the success of any GA as this has a direct influence on the parents of the next generation. It is essential that an evaluation function captures the essence of a good or bad map.

Developing such an evaluation function for the GA is complicated. It is not easy to arrive at a mathematical expression which indicates the correctness of a map.

Historically, there has been a lack of objective criteria by which the quality of existing maps can be assessed or the quality of potential maps can be predicted. Such a facility would allow geneticists to make an objective appraisal of new and old maps.

4.1.4 SETTING PARAMETER VALUES

There are several parameters in a GA that require to be set to appropriate values - population size, number of trials, operator probabilities and evaluation normalisation techniques.

There are established parameter settings described in the literature for GAs using binary representation, binary crossover and mutation (Schaffer(1989)). Finding good settings for non-binary representations is not a trivial task (Davis(1989)) as the techniques available can take a great deal of time. Davis(1989) has devised a system for parameterising operator probabilities for GAs that differ from the traditional type. His technique is being used to measure how effective each of the operators devised are and to obtain appropriate settings.

4.1.5 GENERATING THE INITIAL POPULATION

Traditional GAs generate the initial population at random. However, it has been recognised (DeJong 1988) that if domain specific knowledge is available, it can be usefully exploited in the GA.

The modified GA will use the domain specific information in the form of the experimental data in order to generate maps.

4.1.6 SELECTING A REPRODUCTION TECHNIQUE

Three types of reproduction technique are being investigated - generational replacement, generational replacement with elitism and steady-state replacement without duplicates. (These are reviewed in Davis(1991).)

5. RESULTS

5.1 REPRESENTATION

Chromosomes have been represented as probe maps which denote a simplified version of the complete problem. The choice of representation exemplifies a natural way of splitting up the problem as it is a strategy that the expert may adopt when building maps. A chromosome (probe map) consists of a number of cut sites and the probe, in a particular order as shown below.

eg   B b M m S F * S M F B

The position of the probe is indicated by an asterisk. The length of the chromosome depends on the amount of probe data.

5.2 GENETIC OPERATORS

Two reproduction operators have been developed - "side swap", which is a modified type of crossover (crossover occuring at the position of the probe), and "order swap".

Side swap swaps the LHS of one parent with the RHS of the other parent, as shown in figure 5.2.1.

Order swap, as shown in figure 5.2.2, swaps the order of 2 different REs in the child chromosome as long as the swap is legal. A complete cut site cannot be moved closer to the probe than any of the partial cut sites. A partial cut site cannot be moved further away from the probe than its complete cut site.

5.3 EVALUATION FUNCTION

An evaluation function has been developed which generates a value indicating the goodness of fit between the proposed probe map and the experimental data.

From discussions with geneticists on the features of good and bad maps and through the use of a detailed questionnaire, the important characteristics of maps have been identified. A marking system has been developed which provides a means for scoring characteristics of maps including subjective assessments made by the geneticist. The scoring of the various characteristics can be thought of as sub-evaluations. Three types of sub-evaluations are carried out based on the fit of the single digest results, the fit of the double digest results and the fit of the weak fragments. While each sub-evaluation is not considered appropriate as the sole means of evaluating maps, each has a significant contribution to make to the overall evaluation function.

5.4 INITIAL POPULATION

A "probe map builder" has been developed which takes as input items of probe data (as shown in table 2.1) selected at random

```
         B b M S F * S F B M                  B M F S * b M F B S
               parent 1(p1)                       parent 2(p2)

                                   REPRODUCES
                         |                              |
                        \ /                            \ /

         B b M S F * b M F B S                B M F S * S F B M
         child 1 (LHS p1 + RHS p2)            child 2 (LHS p2 + RHS p1)
```

Figure 5.2.1 - Side Swap

```
             B b M S F * b M F B S
position     0 1 2 3 4 5 6 7 8 9 10


order swap positions 2 and 4 -> B b F S M * b M F B S


order swap positions 0 and 3 - ILLEGAL -> B b M S F * b M F B S
                                              map unchanged
```

Figure 5.2.2 - Order Swap

and creates legal probe maps for the initial population.

6. SUMMARY

This paper has outlined the current development and implementation of a modified GA for a difficult sequencing problem.

7. FUTURE DEVELOPMENTS

The evaluation function developed will be used to validate existing published maps. Once all the results from the GENIE project are collected, the maps generated by the modified GA will be assessed both by geneticists and by comparison with current validated maps which will enable the success of the GA to produce "best fit" maps to be determined. The performance and limitations of the modified GA will be analysed and the implications of the results for other sequencing problems and for problems which require a more complex evaluation function will be established.

REFERENCES

[1] Holland J H, "Adaptation in Natural and Artificial Systems", Ann Arbor: The University of Michegan Press, 1975.

[2] Goldberg D E, "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison Wesley, 1989.

[3] Bethke A D "Genetic Algorithms as Function Optimisers", (Doctoral dissertation, University of Michegan), Dissertation Abstracts International 41(9), 3503B (University Microfilms No. 8106101), 1981.

[4] Bosworth J, Foo N, Zeigler B P, "Comparison of Genetic Algorithms with Conjugate Gradient Methods", (CR-2093) Washington, DC: National Aeronautics and Space Administration, 1972.

[5] Goldberg D E, "Computer-aided Gas Pipeline Operation using Genetic Algorithms and Rule Learning" (Doctoral Dissertation, University of Michegan), Dissertation Abstracts International, 44(10), 3174B (University Microfilms No. 8402282), 1983.

[6] Davis L, "Handbook of Genetic Algorithms", Van Nostrand Reinhold Publishers, 1991.

[7] Goldberg D E, Lingle R, "Alleles, Loci, and the Travelling Salesman Problem", Proceedings of an International Conference on GAs and their Applications, 154-159, 1985.

[8] Grefenstette J J, Gopal R, Rosmaita B J, Van Gucht D, "Genetic Algorithms for the Travelling Salesman Problems" Proceedings of an International Conference on GAs and their Applications, 160-168, 1985.

[9] Oliver I M, Smith D J, Holland J R C, "A Study of Permutation Crossover Operators on the Travelling Salesman Problem", Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on GAs, 224-230, 1987.

[10] Whitley D, Starkweather T, Fuquay D, "Scheduling Problems and Travelling Salesman: the Genetic Edge Recombination Operator", Proceedings of the Third International Conference on GAs, San Mateo,CA: Morgan Kaufmann Publishers. pp133-140, 1989.

[11] Whitley D, Starkweather T, Shaner D, "The Travelling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination", Handbook of Genetic Algorithms, ed L. Davis, Van Nostrand Reinhold Publishers, New York, 1991.

[12] Davis L, "Job Shop Scheduling with Genetic Algorithms", Proceedings of an International Conference on GAs and their Applications, 136-140, 1985.

[13] Syswerda G, "Schedule Optimisation Using Genetic Algorithms", Handbook of Genetic Algorithms, ed L.Davis, Van Nostrand Reinhold Publishers, New York, 1991.

[14] Sefton L, Kelsey G, Kearney P, Povey S, Wolfe J, "A Physical Map of the Human PI and AACT Genes", Genomics 7, 382-388, 1990.

[15] Fox B R, McMahon M B, "Genetic Operators for Sequencing Problems", Foundations of GAs, Rawlins G (ed), San Mateo, CA: Morgan Kaufmann Publishers, 1991.

[16] Schaffer J D, Caruana L J, Eshelman L J, Das R, "A Study of Control Parameters Affecting Online Performance of GAs for Function Optimisation", Proceedings of the Third International Conference on GAs pp 51-60. San Mateo, CA:Morgan Kaufmann Publishers, 1989.

[17] Davis L, "Adapting Operator Probabilities in Genetic Algolrithms", Proceedings of the Third International Conference on GAs pp 61-69. San Mateo, CA:Morgan Kaufmann Publishers, 1989.

[18] De Jong K A, "Learning with GAs: an overview" Machine Learning, 3(2), 121-138, 1988.

A Hybrid Genetic Algorithm Application to a Genetics Sequencing Problem.

J.D.Walker, P.E.File, C.J.Miller, W.B.Samson
Department of Mathematical and Computer Sciences
Dundee Institute of Technology
Bell Street
DUNDEE DD1 1HG
UNITED KINGDOM

ABSTRACT

Currently, geneticists are analysing the structure of the DNA of various organisms to determine the location and sequence of genes. A technique which has played a major role in this process is the use of restriction enzymes and radioactively labelled probes to generate maps of the DNA. Building restriction maps from the results of probed partial digest experiments is a time-consuming and lengthy activity which relies on human judgement of inexact data. Map building is an example of a difficult sequencing problem which requires some form of search to find a good solution from a large problem space of feasible solutions. This paper describes the development of a hybrid genetic algorithm (HGA) suitable for tackling the problem. The results of applying the HGA to a set of map data are presented.

1 INTRODUCTION

As part of the Human Genome Programme, geneticists worldwide are attempting to identify and isolate genes and to generate a "map" showing the position and function of each gene on the human DNA. A current technique which has had a significant impact on genetic mapping is the use of restriction enzymes (REs) along with Pulsed Field Gel Electrophoresis to carry out probed partial digest experiments. Maps are generated from the results of these experiments which indicate the sequence and position of genes and restriction enyzme cut sites along the DNA. Map building is a difficult problem as the data is imperfect allowing many plausible reconstructions. Distinguishing amongst these relies on the geneticist's judgement. Agreement as to which is the best map for a set of data is not straightforward as there is no objective method for appraising potential maps.

Map building is considered to be an activity which could benefit from the application of artificial intelligence techniques. Some form of search is required to find a good solution to the problem. Search techniques which exhaustively search the space of possible maps are not feasible due to time constraints. Genetic algorithms (Holland 1975) are search procedures based on the mechanics and analogy of natural selection. This paper describes the current development of a modified form of genetic algorithm as a search technique to tackle the map building problem.

Map building is described in section 2 and the characteristics of the problem are outlined in section 3. The overall approach for tackling the problem is identified in section 4 and the modified form of genetic algorithm which has been developed is described in section 5. The results of applying the technique to map building are presented in section 6 and discussed in section 7. The paper is summarised in section 8.

2 THE MAP BUILDING PROBLEM

Chromosomes contain the instructions for making each cell in the body and are made from deoxyribonucleic acid (DNA). Humans have 46 chromosomes in each cell (apart from the sex cells) organised into 23 pairs. A sequence of DNA that carries the information for a particular function/protein is called a gene. Genes occupy a specific location on a chromosome. Geneticists are working towards producing a map for the human identifying all the genes and finding their position on a chromosome.

One of the problems that geneticists face when trying to map the position of genes on chromosomes is that there is an enormous amount of DNA but only a fraction of the DNA represents genes. In order to isolate the region of DNA of interest, substances called restriction enzymes (REs) can be used. REs recognise particular sites on the DNA and will cut the DNA at that point. There is a wide range of REs which have different properties and will cut at particular sites. By observing the fragments produced and using already known probes (genetic markers), geneticists can produce restriction maps showing the sites where the enzymes cut and the positions of known probes. Using this technique allows regions of DNA to be characterised.

An example of a RE map is shown in figure 2.1 and the data from which it was generated is shown in table 2.1 (Sefton (1990)). The numbers in table 2.1 represent the lengths of the DNA fragments. Each fragment contains a probe and has been cut at either end by the RE shown.

When one RE is used, the fragments obtained are called "single digests" and each end of the fragment has been cut by that RE. When a mixture of two REs is used, the fragments obtained are called "double digests". Here, each end of the fragment has been cut by one or other of the REs.

RE cut sites are said to be either "complete" or "partial". A complete cut site is a site on the DNA which is always cut by the RE. A partial cut site is a site which is cut on some pieces of DNA and left intact on others. The occurrence of partial cut sites leads to the presence of long fragments. Determining the position of the probes in relation to one another relies on the presence of such fragments. Long fragments are likely to contain two or more probes if the probes are adjacent.

When all fragments present in table 2.1 are combined, taking into account the uncertainty present in the lengths, the choice of RE cutting in the double digests and the nature of the cut sites, a RE map is generated which fits the data best. The map reveals the ordering of the probes, the cut sites of the restriction enzymes, and whether or not the cut sites are partial or complete. The map assembled manually by Sefton(1990) using the data in table 2.1 is shown in figure 2.1.

Figure 2.1 - RE Map assembled from the data in table 2.1 (not to scale) showing the ordering of the RE cut sites and probes.

```
            PIL    PI                           AACT
             |     |                        s    |  SF
 B  SF  M S  |  SF |  sf  B  sf  SF  SF  b  m  S |  M  S  B
 |  |   | |  |  |  |  |   |  |   |   |   |  |  | |  |  |  |
-------------*-----*-----------------------------*-----------DNA
```

NOTES

1. PIL, PI, AACT - probes
2. B, SF, M, S   - REs
3. REs shown in capital letters represent complete cut sites, lower case indicates partial cut sites.

Table 2.1 - Table containing Experimental Data (Sefton (1990)) used to generate the map shown in figure 2.1.

SINGLE DIGEST DATA

B			M			S			SF		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	355	350	350	350	255	255	70	165	[190]	[195]
225	225	275	260	260	80				[10]	[135]	[175]
	[65]	230		[180]						80	135
										65	
										10	

DOUBLE DIGEST DATA

B+M			B+S			B+SF		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
250	250	230	125	125	70	165	[190]	135
225	225	105		[65]		[10]	50	105
130	130						10	
	[65]							

M+S			M+SF			S+SF		
PIL	PI	AACT	PIL	PI	AACT	PIL	PI	AACT
255	255	70	85	80	135		[130]	30
	[180]	30		65	85	70	80	
			[10]	10		[10]	65	
							10	

NOTES

1. PIL, PI, AACT - probes
2. B, SF, M, S - REs
3. Fragments contained in brackets indicate "weak" fragments.

3 MAP BUILDING CHARACTERISTICS

Map building is essentially an ordering problem - the map is linear and after the number and types of cut sites have been determined, they must then be placed in the correct sequence. There are several aspects of map building which make it a difficult activity.

3.1 Maps do not fit perfectly.

Due to experimental limitations, there are errors present in the number of fragments used to build a map and the lengths of the fragments. It is possible that there are areas in the pulsed field gels which appear to represent fragments. Sometimes it is difficult to determine if these are proper fragments or not. When there is ambiguity over the existence of a fragment it is referred to as a "weak fragment". Most weak fragments should fit into a map but some are due to error. They are not used to assemble a map but should in theory fit into the solution. A "good" map is one in which the data, including most of the weak fragments, fits well.

3.2 No objective means of assessing maps.

There has been a lack of objective criteria by which the correctness of maps can be assessed. As part of the project, an attempt has been made to identify appropriate criteria and to incorporate these into an evaluation function. This is discussed further in section 5.2.3.

3.3 Fragments overlap.

All fragments obtained for a particular probe must overlap the region of the probe. Many orderings of fragments are possible and it is difficult to determine where the cut sites are.

3.4 More than one solution may be possible.

Due to the overlapping nature of the fragments and the error in the fragment lengths, it is possible that many feasible map solutions may be generated from the same data. For example, it can be shown that there are at least 7.2 x 10e14 permutations of the cut sites for the data shown in table 2.1 and this is considered to be small data set and a conservative estimate.

4 STRATEGY FOR TACKLING THE PROBLEM

As the number of possible maps for a set of data is very large, using any search technique which requires all maps to be exhaustively generated is not appropriate.

A two stage approach was taken to tackle the map building problem. The overall problem of generating the map was decomposed into the smaller problem of generating single probe maps. Using the example shown in table 2.1, this reduced the number of possible maps for PIL to 15,840, for PI to 224,640 and for AACT to 34,560.

In order to arrive at a set of feasible maps for each probe, it was proposed that a genetic algorithm be developed. Genetic algorithms have been used successfully as search techniques across a range of problems; particularly combinatoric problems.

Once a number of good maps have been generated for each probe, an optimisation algorithm will be developed to merge the separate probe maps together to arrive at a complete map.

5 GENETIC ALGORITHMS

Genetic Algorithms (GAs) are search procedures inspired by the mechanics of natural selection and have been developed to be robust over a variety of problem domains. Goldberg(1989) is the standard introductory text to GAs. In a GA, a potential solution is referred to as a chromosome. A number of chromosomes are generated at random to produce what is called an initial population. Each chromosome is assigned a score indicating how good it is, which is referred to as its fitness. A new population is produced by performing operations patterned after genetic operations such as sexual recombination (crossover) and fitness proportionate reproduction (Darwinian survival of the fittest). The more fit chromosomes (the better solutions), reproduce (combine together)in an attempt to generate more highly adapted individuals (solutions that are better still). This process is repeated and each successive population is called a generation. After a fixed number of generations (trials), the fittest chromosome represents the solution.

The steps in a traditional genetic algorithm are shown in figure 5.1.

```
1. Generate an initial population
2. Evaluate the population
repeat
     3. Reproduce and generate a new population
     4. Evaluate the new population
until the number of trials is up
```

Figure 5.1 - A Traditional Genetic Algorithm

In order to apply a genetic algorithm to a particular problem, there are 4 issues that must be considered.

1. REPRESENTATION - this is the way in which the problem is be "represented" in genetic algorithm notation.

2. GENETIC OPERATORS - during reproduction, operators analogous to the biological operators of crossover and mutation are applied. Traditionally, the crossover operator involves exchanging strings at random, combining parts of good individuals in an attempt to create a more fit individual. This is illustrated in figure 5.2. The mutation operator is applied less frequently and plays a secondary role. It involves changing a single value in an individual in order to generate some unexpected variation into the population and is illustrated in figure 5.3.

3. FITNESS FUNCTION - this is the means by which potential solutions can be evaluated and a fitness value can be assigned. In

order to apply a genetic algorithm to a problem, there must be some method of measuring the goodness of potential solutions. The fitness function must capture what is good about a potential solution.

4. PARAMETER SETTINGS - there are various parameters that must be set for the genetic algorithm such as population size, frequency of application of the genetic operators and number of generations to run.

[Individuals are represented in binary notation.]

```
parent 1 = 1 1 1 1 1 0 1       parent 2 = 1 0 1 1 0 0 0

             |                                |
            \ /                              \ /

          Reproduction and crossover between position 4 - 7

                        |
                       \ /

           child = 1 1 1 1 0 0 0
```

Figure 5.2 The Crossover Operator in Action

```
                  child = 1 1 1 1 0 0 0

                              |
                             \ /

                mutation of bit position 7

                              |
                             \ /

                        1 1 1 1 0 0 1
```

Figure 5.3 The Mutation Operator in Action

5.1 HYBRID GENETIC ALGORITHMS

As traditional GAs are not effective for sequencing problems (illegal solutions can be generated), it was proposed that a modified GA be developed to generate probe maps. The concept of a hybrid GA (HGA) has been suggested by a number of authors(Bethke(1981); Bosworth (1972); Goldberg(1983); Davis (1991)). An HGA is a GA which incorporates problem specific information or various search techniques. Research conducted in this area has shown an HGA approach to be promising for combinatorial optimisation problems such as the Travelling Salesman Problem (Goldberg(1985), Grefenstette (1985), Oliver(1987), Whitley (1989,1991)) and scheduling problems (Davis(1985), Syswerda (1991)).

In the HGA developed for the probe maps, the representation and optimisation techniques employed by the expert are used to ensure that the domain knowledge embodied in the encoding is preserved. The HGA operators are tailored to apply to the new representation and include domain-based heuristics. An evaluation function which is based on subjective assessment has also been developed. Other problem specific knowledge has guided the development of the overall algorithm.

One of the main aims of GA research has been to develop an algorithm that is robust across a variety of problem domains and operates without problem specific information. There is a well established theory which can account for the success of the traditional GA as a search procedure. It is the intention here to see if the power of the traditional GA can be harnessed and used as the basis of an effective algorithm for a real-world complex problem. In modifying the traditional GA, the theoretical foundation is left behind. It has been recognised that experimentation is leading the field in this area of research and that the existing theory can only provide guiding principles (Whitley 1993).

5.2 IMPLEMENTATION ISSUES

A "front-end" menuing system has been developed for the HGA which allows for various features to be selected. This has facilitated the implementation of the HGA and allowed the effect of changing particular features to be observed.

There are several areas which have been addressed during the development of the HGA and these are discussed in the following sections.

5.2.1 REPRESENTING THE PROBLEM

Traditional GAs generally represent chromosomes as binary vectors. However, having considered several options, it was decided that in the HGA, the chromosome syntax should be changed to reflect the problem, as shown in figure 2.1.

Chromosomes have been represented as probe maps which consist of a number of cut sites and the probe, in a particular order as shown below.

eg   B b M m S F * S M F B

The position of the probe is indicated by an asterisk. The length of the chromosome depends on the amount of probe data.

5.2.2 DEVELOPING A SET OF GENETIC OPERATORS

The difficulties encountered when applying the traditional genetic operators to map assembly are similiar to those found when applying them to other sequencing problems such as the Travelling Salesman Problem (TSP). In the case of the TSP, there are constraints placed on the symbol string that represents a tour of the cities, in that no city can appear more than once. The traditional recombination operators rearrange symbols on a chromosome independently of each other. When the traditional operators are applied to city sequences, cities can appear more than once or not at all.

When developing operators for map assembly, taking into account previous work (Goldberg (1989), Fox and McMahon(1991)), a way of breaking up the chromosomes that was natural for the problem was sought. The traditional operators have been modified to meet the constraints on the chromosomes while preserving the motivating principles behind these operators. Crossover provides an opportunity for the best attributes of both parent strings to be incorporated into the offspring. Mutation is a mechanism for introducing necessary attributes into an individual when those attributes do not already exist within the current population.

Two operators have been developed - "side swap", which is a modified type of crossover (crossover occuring at the position of the probe), and "order swap".

Side swap swaps the LHS of one parent with the RHS of the other parent, as shown in figure 5.2.2.1. Order swap, as shown in figure 5.2.2.2, is a unary operator which swaps the order of 2 different REs in the child chromosome as long as the swap is legal. A complete cut site cannot be moved closer to the probe than any of the partial cut sites. A partial cut site cannot be moved further away from the probe than its complete cut site.

```
        B b M S F * S F B M                B M F S * b M F B S
            parent 1(p1)                       parent 2(p2)

                               REPRODUCES
                    |                          |
                   \ /                        \ /

        B b M S F * b M F B S              B M F S * S F B M
      child  1 (LHS p1 + RHS p2)     child 2 (LHS p2 +  RHS  p1)
```

Figure 5.2.2.1 - Side Swap

```
             B b M S F * b M F B S
position     0 1 2 3 4 5 6 7 8 9 10


order swap positions 2 and 4 -> B b F S M * b M F B S


order swap positions 0 and 3 - ILLEGAL -> B b M S F * b M F B S
                                             map unchanged
```

Figure 5.2.2.2 - Order Swap

5.2.3 DEVELOPING THE EVALUATION FUNCTION

The way in which chromosomes are assessed is critical to the success of any GA as this has a direct influence on the parents of the next generation. It is essential that an evaluation function captures the essence of a good or bad map. Developing such an evaluation function for the HGA is complicated. It is not easy to arrive at a general expression which indicates the correctness of a map.

Historically, there has been a lack of objective criteria by which the quality of existing maps can be assessed or the quality of potential maps can be predicted. Such a facility would allow geneticists to make an objective appraisal of new and old maps.

From discussions with geneticists on the features of good and bad maps and through the use of a detailed questionnaire, an attempt has been made to identify the important characteristics of maps. A marking system has been developed which provides a means for scoring characteristics of maps including subjective assessments made by the geneticist. The scoring of the various characteristics can be thought of as sub-evaluations. Currently two types of sub-evaluations are carried out based on the fit of the single digest results and the fit of the double digest results. While each sub-evaluation is not considered appropriate as the sole means of evaluating maps, each has a significant contribution to make to the overall evaluation function. The evaluation function generates a score indicating the goodness of fit between the proposed probe map and the experimental data. A map which gains the highest score is called an "ideal map".

5.2.4 SETTING PARAMETER VALUES

There are several parameters in a GA that require to be set to appropriate values - population size, number of trials, operator probabilities and evaluation normalisation techniques. There are established parameter settings described in the literature for traditional GAs which use binary representation, binary crossover and mutation (Schaffer(1989)). However, finding good settings for non-binary representations is not a trivial task (Davis(1989)). It is recommended that appropriate settings for non standard GAs be derived by experimentation (Syswerda 1991) and this procedure is being followed.

5.2.5 GENERATING THE INITIAL POPULATION

Traditional GAs generate the initial population at random. However, it has been recognised (DeJong 1988) that if domain specific knowledge is available, it can be usefully exploited in the GA. The HGA uses the domain specific information in the form of the experimental data in order to generate maps. A "probe map builder" has been developed which takes as input items of probe data (as shown in table 2.1) selected at random and creates legal probe maps for the initial population.

5.2.6 SELECTING A REPRODUCTION TECHNIQUE

Three types of reproduction technique are being investigated - generational replacement, generational replacement with elitism and steady-state replacement without duplicates. (These are reviewed in Davis 1991).

6 RESULTS

The data presented in table 2.1 was chosen as a test set as it represents a small, manageable number of probes and it is self-contained, in that all combinations of double digests were performed. The results of applying the HGA to build an "ideal map" for the AACT probe are shown. An ideal map is a map which is awarded the highest score possible from the evaluation function.

The size of the search space for the probe was calculated and is shown below. As the search space was relatively small, it was possible to run an exhaustive search to see how many "ideal maps" were present in the search space. In order to see how well the evalution function was operating, each of the sub-evaluations was carried out separately then applied together. Maps were evaluated on how well the single digest data fitted, how well the double digest data fitted and how well both the single and double digest fitted when taken together.

The parameters for the HGA were held constant for each of the runs.

	AACT
Total no. of maps possible	34,560
No. of maps generated by the HGA	20,000

EVALUATING ON THE FIT OF THE SINGLE DIGEST DATA

Total no. of "ideal maps" possible	4256
No. of "ideal maps" generated by the HGA	2062

EVALUATING ON THE FIT OF THE DOUBLE DIGEST DATA

Total no. of "ideal maps" possible	60
No. of "best maps" generated by the HGA	41

EVALUATING ON THE FIT OF THE SINGLE AND DOUBLE DIGEST DATA

Total no. of "ideal maps" possible	26
No. of "ideal maps" generated by the HGA	25

7 DISCUSSION

Evaluating on the fit of the single digest data is a very broad evaluation in that 12.3% of all AACT maps would be considered ideal maps based on this evaluation alone. It is harder for maps to score well evaluating on the fit of the double digest data and harder still to score well when both evaluations are considered together. There are 60 maps which score as ideal maps for AACT, evaluating on the fit of the double digest data and 26 maps applying both evaluations together. The HGA managed to generate 25 of the 26 ideal maps. Out of the 26 ideal maps there were 13 map mirror images and the HGA found all of these. As a search technique, the HGA would appear to be functioning well. When the number of cut sites in the ideal maps was compared with the number of cut sites in the published map, a critical error on the published map was highlighted in the number and position of Bss cut sites. The published map has been revised to correct this error. When the 25 ideal maps were analysed in more detail, it was found that further information concerning fragment lengths must be taken into account to identify optimal maps.

Preliminary results obtained using the PIL and PI probe data indicate there are no maps in the search space which can score perfectly evaluating on the fit of the double digest data. The scoring mechanism implemented appears to have over constrained the problem and this is being examined.

The results highlight the areas in which the current evaluation function needs to be refined to include more problem specific information to allow optimal maps to be identified.

Further experimentation is under way to test the sensitivity of the parameter settings in the HGA.

Once acceptable sets of maps can be obtained for the individual probe maps, work will commence on the second stage. This will involve developing a suitable optimisation algorithm which will be used to merge the separate probe maps together.

8 SUMMARY

This paper has outlined the approach that has been taken to apply artificial intelligence techniques to the difficult problem of DNA map building. The current development and implementation of an HGA for generating probe maps is presented and the results obtained to date appear promising.

REFERENCES

Bethke A D "Genetic Algorithms as Function Optimisers", (Doctoral dissertation, University of Michegan), Dissertation Abstracts International 41(9), 3503B (University Microfilms No. 8106101), 1981.

Bosworth J, Foo N, Zeigler B P, "Comparison of Genetic Algorithms with Conjugate Gradient Methods", (CR-2093) Washington, DC: National Aeronautics and Space Administration, 1972.

Davis L, "Job Shop Scheduling with Genetic Algorithms", Proceedings of an International Conference on GAs and their Applications, 136-140, 1985.

Davis L, "Adapting Operator Probabilities in Genetic Algolrithms", Proceedings of the Third International Conference on GAs pp 61-69. San Mateo, CA:Morgan Kaufmann Publishers, 1989.

Davis L, "Handbook of Genetic Algorithms", Van Nostrand Reinhold Publishers, 1991.

De Jong K A, "Learning with GAs: an overview" Machine Learning, 3(2), 121-138, 1988.

Fox B R, McMahon M B, "Genetic Operators for Sequencing Problems", Foundations of GAs, Rawlins G (ed), San Mateo, CA: Morgan Kaufmann Publishers, 1991.

Goldberg D E, "Computer-aided Gas Pipeline Operation using Genetic Algorithms and Rule Learning" (Doctoral Dissertation, University of Michigan), Dissertation Abstracts International, 44(10), 3174B (University Microfilms No. 8402282), 1983.

Goldberg D E, Lingle R, "Alleles, Loci, and the Travelling Salesman Problem", Proceedings of an International Conference on GAs and their Applications, 154-159, 1985.

Goldberg D E, "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison Wesley, 1989.

Grefenstette J J, Gopal R, Rosmaita B J, Van Gucht D, "Genetic Algorithms for the Travelling Salesman Problems" Proceedings of an International Conference on GAs and their Applications, 160-168, 1985.

Holland J H, "Adaptation in Natural and Artificial Systems", Ann Arbor: The University of Michegan Press, 1975.

Oliver I M, Smith D J, Holland J R C, "A Study of Permutation Crossover Operators on the Travelling Salesman Problem", Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on GAs, 224-230, 1987.

Schaffer J D, Caruana L J, Eshelman L J, Das R, "A Study of Control Parameters Affecting Online Performance of GAs for Function Optimisation", Proceedings of the Third International Conference on GAs pp 51-60. San Mateo, CA:Morgan Kaufmann Publishers, 1989.

Sefton L, Kelsey G, Kearney P, Povey S, Wolfe J, "A Physical Map of the Human PI and AACT Genes", Genomics 7, 382-388, 1990.

Syswerda G, "Schedule Optimisation Using Genetic Algorithms", Handbook of Genetic Algorithms, ed L.Davis, Van Nostrand Reinhold Publishers, New York, 1991.

Whitley D, Starkweather T, Fuquay D, "Scheduling Problems and Travelling Salesman: the Genetic Edge Recombination Operator", Proceedings of the Third International Conference on GAs, San Mateo,CA: Morgan Kaufmann Publishers. pp133-140, 1989.

Whitley D, Starkweather T, Shaner D, "The Travelling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination", Handbook of Genetic Algorithms, ed L. Davis, Van Nostrand Reinhold Publishers, New York, 1991.

Whitley D (1993), Introduction to Foundations of Genetic Algorithms 2, edited by L.Darrell Whitley, Morgan Kaufmann Publishers, San Mateo, California.

Building DNA Maps : A Genetic Algorithm Based Approach

J.D.Walker, P.E.File, C.J.Miller, W.B.Samson
Department of Mathematical and Computer Sciences
Dundee Institute of Technology
Bell Street
DUNDEE DD1 1HG
UNITED KINGDOM

Abstract. Geneticists are investigating the structure of the human DNA and are working towards producing a map identifying the location of all of the genes on the chromosomes. One of the techniques employed to generate maps uses restriction enzymes to break up the DNA and radioactively labelled probes to mark regions of interest. The fragments of DNA obtained from such experiments are pieced together in what is considered to be the best order to create a restriction map. Building maps is a time consuming and problematic process as many solutions may be feasible and evaluating between competing solutions is carried out subjectively. This paper describes the continuing development of an artificial intelligence technique based on a genetic algorithm for building restriction maps from the results of probed partial digest experiments. An objective system for assessing maps is developed which is used by the genetic algorithm to search for map solutions which fit the experimental fragments well.

1 Introduction

There has been a considerable amount of interest shown in the development and application of Artificial Intelligence (AI) techniques to biological problems, specifically at the molecular level [1],[2]. This is partly in response to the vast amount of data produced by the Human Genome Programme which has presented a variety of stimulating new challenges to researchers working in the field of AI. The problem of generating DNA restriction maps from the results of partial digest experiments has been studied by several groups eg [3,4,5,6].

Most of the applications reported rely on the use of cloned DNA, which reduces the complexity of the problem considerably. The work described here concentrates on the harder problem of using total DNA and restriction enzymes (REs) along with Pulsed Field Gel Electrophoresis to carry out probed partial digest experiments. The problem is more difficult due to limitations in the experimental process. Error in the data allows for many plausible map reconstructions. Distinguishing amongst these relies on the geneticist's judgement. Agreement as to which is the best map for a set of data is not straightforward as there is no objective method for appraising potential maps.

Map building is considered to be an activity which could benefit from the application of AI techniques. In order to generate a map which fits the experimental data well, some form of search is required to find a good solution to the problem. Search techniques which exhaustively search the space of possible maps are not feasible due to time constraints. Genetic algorithms (GAs)[7] are search procedures based on the mechanics of natural selection. Previous work [8] describes the preliminary development of a form of GA suitable for tackling the map building problem. Recent results on a constrained version of the problem [9] suggest that the approach appears promising. This paper describes the continuing development of an objective system for evaluating maps and of the incorporation of that system into a modified form of genetic algorithm.

The paper is organised into the following sections. Map building is described in section 2 and the characteristics of the problem are outlined in section 3. The overall approach for tackling the problem is identified in section 4. Genetic algorithms are introduced in section 5 and the modified form of genetic algorithm which has been developed is described in section 6. The results of applying the technique to map building are presented in section 7 and discussed in section 8. The paper is summarised in section 9.

2 Description Of The Map Building Problem

Geneticists are attempting to identify and isolate genes and to generate a "map" showing the position and function of each gene on the human DNA. A current technique which has had a significant impact on genetic mapping is the use of restriction enzymes (REs) along with Pulsed Field Gel Electrophoresis to carry out probed partial digest experiments. Maps are generated from the results of these experiments which indicate the sequence and position of genes and RE cut sites along the DNA.

Chromosomes contain the instructions for making each cell in the body and are made from deoxyribonucleic acid (DNA). Humans have 46 chromosomes in each cell (apart from the sex cells) organised into 23 pairs. A sequence of DNA that carries the information for a particular function/protein is called a gene. Genes occupy a specific location on a chromosome.

One of the problems that geneticists face when trying to map the position of genes on chromosomes is that there is an enormous amount of DNA but only a fraction of the DNA represents genes. In order to isolate the region of DNA of interest, REs can be used. REs recognise particular sites on the DNA and will cut the DNA at that point. There is a wide range of REs which have different properties and will cut at particular sites. By observing the fragments produced and using already known probes (genetic markers), geneticists can produce restriction maps showing the sites where the enzymes cut and the positions of known probes. Using this technique allows regions of DNA to be characterised.

An example of a RE map is shown in figure 1 and the data from which it was generated is shown in table 1 (taken from [10]). The numbers in table 1 represent the lengths of the DNA fragments. Each fragment contains a probe and has been cut at either end by the RE shown.

When one RE is used, the fragments obtained are called "single digests" and each end of the fragment has been cut by that RE. When a mixture of two REs is used, the fragments obtained are called "double digests". Here, each end of the fragment has been cut by one or other of the REs.

RE cut sites are said to be either "complete" or "partial". A complete cut site is a site on the DNA which is always cut by the RE. A partial cut site is a site which is cut on some pieces of DNA and left intact on others. The occurrence of partial cut sites leads to the presence of long fragments. Determining the position of the probes in relation to one another relies on the presence of such fragments. Long fragments are likely to contain two or more probes if the probes are adjacent.

When all fragments present in table 1 are combined, taking into account the uncertainty present in the lengths, the choice of RE cutting in the double digests and the nature of the cut sites, a RE map is generated which fits the data best. The map reveals the ordering of the probes, the cut sites of the REs, and whether or not the cut sites are partial or complete. The map assembled manually by [10] using the data in table 1 is shown in figure 1.

Figure 1 - RE Map assembled from the data in table 1 (not to scale) showing the ordering of the RE cut sites and probes.

```
           PIL   PI                            AACT
            |     |                       s    | SF
B SF  M S   | SF  | sf  B sf  SF  SF   b  m  S | M  S  B
| |   | |   | |   | |   | |   |   |    |  |  | | |  |  |
------------*-----*----------------------------*----------DNA
```

NOTES

1. PIL, PI, AACT - probes
2. B, SF, M, S - REs
3. REs shown in capital letters represent complete cut sites, lower case indicates partial cut sites.

Table 1 - Table containing Experimental Data taken from [10] used to generate the map shown in figure 1.

SINGLE DIGEST DATA

B

PIL	PI	AACT
250	250	355
225	225	275
	[65]	230

M

PIL	PI	AACT
350	350	350
260	260	80
	[180]	

S

PIL	PI	AACT
255	255	70

SF

PIL	PI	AACT
165	[190]	[195]
[10]	[135]	[175]
	80	135
	65	
	10	

DOUBLE DIGEST DATA

B+M

PIL	PI	AACT
250	250	230
225	225	105
130	130	
	[65]	

B+S

PIL	PI	AACT
125	125	70
	[65]	

B+SF

PIL	PI	AACT
165	[190]	135
[10]	50	105
	10	

M+S

PIL	PI	AACT
255	255	70
	[180]	30

M+SF

PIL	PI	AACT
85	80	135
	65	85
	[10]	10

S+SF

PIL	PI	AACT
	[130]	30
70	80	
[10]	65	
	10	

NOTES

1. PIL, PI, AACT - probes
2. B, SF, M, S - REs
3. Fragments contained in brackets indicate "weak" fragments.

3 Characteristics Of Map Building

Map building is an example of a sequencing problem. The map is linear and after the number and types of cut sites have been determined, they must be placed in the correct order. There are several aspects of map building which make it a difficult activity.

3.1 Experimental Data Tends To Contain Inaccuracies.

Due to limitations in the experimental process, there are errors present in the number of fragments used to build a map and the lengths of the fragments. It is possible that there are areas in the pulsed field gels which appear to represent fragments. Sometimes it is difficult to determine if these are proper fragments or not. When there is ambiguity over the existence of a fragment it is referred to as a "weak fragment". Most weak fragments should fit into a map but some are due to error. They are not used to assemble a map but should in theory fit into the solution. A "good" map is one in which the data, including most of the weak fragments, fits well.

3.2 Map Assessment Is Subjective.

Currently, maps are assessed subjectively by the geneticist. Where several possible maps are proposed using the same data the geneticist applies their expert knowledge and judgement to determine which one is best.

As part of the project, an attempt has been made to identify appropriate objective criteria by which maps can be assessed. These criteria have been incorporated into an evaluation function. This is discussed further in section 6.1.2.

3.3 DNA Fragments Overlap.

All the DNA fragments obtained for a particular probe must overlap the region of the probe. It is possible for the fragments to be ordered in many different ways and it is difficult to determine the number and position of cut sites.

3.4 More Than One Solution May Be Possible.

Due to the overlapping nature of the fragments and the error in the fragment lengths, it is possible that many feasible map solutions may be generated from the same data. For example, it can be shown that there are at least 2.06 x 10e21 permutations of the cut sites for the data shown in table 1 and this is considered to be small data set and a conservative estimate.

4 Strategy For Tackling The Problem

As the number of possible maps for a set of data is very large, using any search technique which requires all maps to be exhaustively generated is not considered appropriate due to the length of time it would take. It was decided that some type of heuristic method would be employed which would hopefully find a near optimal solution to the problem in a reasonable length of time.

To tackle the map building problem, a two stage approach was adopted. The overall problem of generating the map was decomposed into the smaller problem of generating single probe maps. Using the example shown in table 1, this reduced the number of possible maps for PIL to 4,055,040, for PI to 57,507,840 and for AACT to 8,847,360.

In order to arrive at a set of feasible maps for each probe, it was proposed that a genetic algorithm be developed [3]. Preliminary results on a constrained version of the map building problem suggested that the approach appeared promising [4]. A genetic algorithm is an example of a heuristic method which has been used successfully as a search technique across a range of problems; particularly combinatoric problems. Once a number of good maps have been generated for each probe, an optimisation algorithm will be developed to merge the separate probe maps together to arrive at a complete map.

5 Genetic Algorithms

Genetic algorithms (GAs) were introduced by Holland [7] as adaptive procedures based on the mechanics and analogy of natural selection. They have generated much interest in recent years and their use is being investigated in areas as diverse as machine learning, machine vision, function optimisation and NP-hard problems. A GA is a type of search procedure which takes as input several possible solutions to a problem and attempts to progressively improve upon them through a sequence of changes which mimics the process of natural selection. Holland recognised that natural systems are more robust than artificial systems and attempted to achieve robustness by developing an algorithm which emulated natural selection. In natural selection, the individuals which are best adapted to their environment tend to have the greatest chance of survival and reproduce more, passing on their genes to the next population. GAs have been developed as search procedures which are population based and proceed over a number of generations. The criteria of "survival of the fittest" provides evolutionary pressure for populations to develop increasingly fit individuals.

In a GA, a potential solution to the problem is represented as an ordered list of values. A number of potential solutions are generated at random to produce what is called an initial population. The fitness of the population is evaluated by assessing the fitness of each individual (or potential solution) in the population.

A new population is produced by performing operations patterned after genetic operations such as sexual recombination (crossover) and fitness proportionate reproduction (Darwinian survival of the fittest). The more fit individuals (the better solutions), reproduce (combine together)in an attempt to generate more highly adapted individuals (solutions that are better still.) This process is repeated. Each successive population is called a generation. After a fixed number of generations (trials), the fittest individual represents the solution.

The steps in a traditional genetic algorithm are shown in figure 2.

Figure 2 - A Traditional Genetic Algorithm

```
 Generate an initial population;

 Evaluate the population;

REPEAT

     Reproduce and generate a new population;

     Evaluate the new population;

UNTIL (the number of trials is up);
```

The GA is a "parallel" algorithm in that it transforms a population of individual objects into a new population. During reproduction, parents are selected to mate, the recombination operators are applied and the children are inserted into the new population. Selecting parents to reproduce in proportion to fitness ensures that above average parents are selected to reproduce more frequently. Various methods for selecting parents have been proposed [11,12], one of the most commonly used is called "roulette wheel" parent selection which involves randomly selecting parents to reproduce using a roulette wheel biased in proportion to the parents fitness. In a standard GA, a whole new population of individuals are created, saving the best one from the previous generation (known as Generational Replacement with Elitism). This ensures that when the best solution is found it is not lost through disruption from crossover or mutation.

There is a well established theory which has been develped by Holland to explain why GAs work and it is based on a binary representation and the notion of a schema. A description of the Schema Theorem and an introduction to GAs in general is contained in [13].

5.1 Implementating A Genetic Algorithm

In order to apply a GA to a particular problem, there are various issues that must be addressed such as representation, generating the initial population, selecting an evaluation function, choice of operators and settings for parameter values. These are briefly discussed in the following sections.

5.1.1 Generating The Initial Population

Some technique must be chosen to generate the intial population for the GA. At the simplest level, potential solutions can be generated at random however it has been recognised that if there is problem specific information available, it is beneficial to make use of it [14].

5.1.2 Representing The Problem.

This is the way in which the problem is be "represented" in genetic algorithm notation. Traditionally a fixed string binary representation is used, however, it has been recognised that the binary string representation is not the most suitable for certain types of problems. In some cases, the most natural representation involves more complex data structures and by linearising the data structure into a string representation limits the window by which the system observes the world.

5.1.3 Choice Of Recombination Operators.

During reproduction, recombination operators analogous to the biological operators of crossover and mutation are applied. Traditionally, the crossover operator involves exchanging strings at random, combining parts of good individuals in an attempt to create a more fit individual. This is illustrated in figure 3. The role of crossover is to provide an opportunity for the best attributes of both parent strings to be incorporated into the offspring. Mutation is a mechanism for introducing necessary attributes into an individual when those attributes do not already exist within the current population. The mutation operator is applied less frequently and is considered to play a secondary role. It involves changing a single value in an individual in order to generate some unexpected variation into the population and is illustrated in figure 4.

Figure 3 The Crossover Operator in Action

[Individuals are represented in binary notation.]

```
 parent 1 = 1 1 1 1 1 0 1          parent 2 = 1 0 1 1 0 0 0
position    0 1 2 3 4 5 6         position    0 1 2 3 4 5 6
                |                                   |
               \ /                                 \ /

        Reproduction and crossover between position 4 - 6

                              |
                             \ /

                   child = 1 1 1 1 0 0 0
                  position  0 1 2 3 4 5 6
```

Figure 4 The Mutation Operator in Action

```
                 child = 1 1 1 1 0 0 0
                position  0 1 2 3 4 5 6
                            |
                           \ /

                mutation of bit position 6

                            |
                           \ /

                      1 1 1 1 0 0 1
```

5.1.4 Selecting An Appropriate Evaluation Function

The evaluation function in a GA plays the role of the environment by rating potential solutions in terms of their fitness. In order to apply a GA to a problem, there must be some method of measuring the goodness of potential solutions. The evaluation function must capture what is good about a potential solution. The evaluation function is the one area where the traditional GA requires problem specific knowledge. For some types of problems, the choice of evaluation function is obvious but for other problems it is not as straightforward. For example, if a GA is to be used to optimise a particular function, the fitness of a potential solution can be assessed by inserting the value into the function and computing the function.

In a combinatorial optimisation problem such as the Travelling Salesman Problem (TSP), a potential solution represents the distance of a tour and the object is to minimise that distance. A table can be maintained containing the distance between cities. For other problems, such as the conjunctive normal form (CNF) - satisfiability problem, the choice of evaluation function is not clear. In the CNF-satisfiability problem, a logical expression made up of clauses of logical variables, is represented in CNF. The problem is to find a truth assignment for the variables in the expression so that the whole expression evaluates to TRUE. Specific truth assignments result in the whole expression evaluating to TRUE or FALSE. During the search for the solution to the problem, the expression will evaluate to FALSE unless a solution is found and it is not possible to discriminate between "good" and "bad" solutions. DeJong [15] discusses some of the problems associated with developing evaluation functions.

5.1.5 Setting Parameter Values

There are several parameters in a GA that require to be set to appropriate values such as population size, number of trials, operator probabilities and evaluation normalisation techniques. There are established parameter settings described in the literature for traditional GAs which use binary representation, binary crossover and mutation [16].

6 Non-Standard Genetic Algorithms

The traditional GA is one which uses binary representation, one-point crossover and mutation, generational replacement normally with elitism and some fitness normalisation process. The traditional GA has the theoretical underpinnings of the Schema Theorem. A central goal in GA research has been to develop an algorithm that is robust which can perform well across a variety of problem domains with no problem specific knowledge. There are many researchers who are still pursuing this goal and working with the traditional GA. There has also been considerable interest in the application of GAs to real-world problems. From an AI perspective, the standard GA can be classed as a "weak method" as it makes few assumptions about the problem domain and is widely applicable. As a weak method, the traditional GA is unlikely to be the best algorithm to use for any particular problem. Various modifications to the traditional GA have been proposed. Some of these have been motivated by a wish to improve the performance of a GA for particular problems. Other researchers have been interested in developing the GA to handle constraint problems.

For the problem of generating probe maps, it was proposed that a modified GA be developed as the traditional GA is not effective for sequencing problems as illegal solutions can be generated. The concept of a hybrid GA (HGA) has been suggested by a number of authors [17,18,19]. Davis's [20] description involves hybridising the GA with the best features of alternative techniques for solving a particular problem. The representation should reflect a natural way of stating the problem and problem specific genetic operators should be developed to work with the new representation. Research conducted in this area has shown an HGA approach to be promising for combinatorial optimisation problems such as the TSP [21,22,23,24,25] and scheduling problems [26,27].

In the HGA developed for the probe maps, the representation and optimisation techniques employed by the expert are used to ensure that the domain knowledge embodied in the encoding is preserved. The HGA operators are tailored to apply to the new representation and include domain-based heuristics. An evaluation function which is based on subjective assessment has also been developed. Other problem specific knowledge has guided the development of the overall algorithm. In modifying the traditional GA, the theoretical foundation is left behind. It has been recognised that experimentation is leading the field in this area of research and that the existing theory can only provide guiding principles [28].

The HGA developed to build probe maps is described in the following sections.

6.1 Problem Representation

The traditional GA represents chromosomes as fixed length binary strings however it is not the most suitable for certain types of problems. For numerical optimisation problems which require a high degree of precision, real number representation tends to give solutions with greater accuracy and in fewer generations than equivalent implementations using binary (eg [20],[29]). Other representations have used character strings, gray coding, integers and matrices.

Having considered several options, it was decided that in the HGA, the chromosome syntax should be changed to reflect the problem, as shown in figure 1. Chromosomes have been represented as probe maps which consist of a number of cut sites and the probe, in a particular order as shown below.

eg   B b M m S F * S M F B

The position of the probe is indicated by an asterisk. The length of the chromosome depends on the amount of probe data.

6.1.1 Developing Recombination Operators

The difficulties encountered when applying the traditional genetic operators to map assembly are similiar to those found when applying them to other sequencing problems such as the TSP. In the case of the TSP, there are constraints placed on the symbol string that represents a tour of the cities, in that no city can appear more than once. The traditional recombination operators rearrange symbols on a chromosome independently of each other. When the traditional operators are applied to city sequences, cities can appear more than once or not at all.

When developing operators for map assembly, taking into account previous work [13,30], a way of breaking up the chromosomes that was natural for the problem was sought. The traditional operators have been modified to meet the constraints on the chromosomes while preserving the motivating principles behind these operators. Crossover provides an opportunity for the best attributes of both parent strings to be incorporated into the offspring. Mutation is a mechanism for introducing necessary attributes into an individual when those attributes do not already exist within the current population.

Three operators have been developed - "side swap", which is a modified type of crossover (crossover occuring at the position of the probe), "order swap" and "site swap". Using these three operators enables all legal maps in the search space to be reached. Side swap swaps the LHS of one parent with the RHS of the other parent, as shown in figure 5. Order swap, as shown in figure 6, is a unary operator which swaps the order of two different REs in the child chromosome as long as the swap is legal. A complete cut site cannot be moved closer to the probe than any of the partial cut sites. A partial cut site cannot be moved further away from the probe than its complete cut site. Site swap is a unary operator which swaps the **value** of a cut site as shown in figure 7. The value of a cut site is whether or not it is a partial cut site or a complete cut site. A RE is selected at random from the LHS or RHS of the map and the value of the outermost cut site for that RE is swapped.

Figure 5 - Side Swap

```
   B b M S F * S F B M              B M F S * b M F B S
        parent 1(p1)                     parent 2(p2)

                         REPRODUCES
             |                        |
            \ /                      \ /

     B b M S F * b M F B S            B M F S * S F B M
   child 1 (LHS p1 + RHS p2)     child 2 (LHS p2 + RHS p1)
```

Figure 6 - Order Swap

```
                   B b M S F * b M F B S
        position   0 1 2 3 4 5 6 7 8 9 10


    order swap positions 2 and 4 -> B b F S M * b M F B S


 order swap posns 0 and 3 - ILLEGAL -> B b M S F * b M F B S
                                           map unchanged
```

Figure 7 - Site Swap

```
                   B b M S F * b M F B S
        position   0 1 2 3 4 5 6 7 8 9 10


site swap RE position 2  -> B b m S F * b M F B S
```

6.1.2 Developing The Evaluation Function

The way in which potential solutions are assessed is critical to the success of any GA as this has a direct influence on the parents of the next generation. It is essential that the evaluation function captures the essence of a good or bad map. Developing such an evaluation function for the map building problem is complicated as map evaluation is a highly judgemental activity relying on the geneticists intuition and expert knowledge. It is not easy to arrive at a general expression which indicates the correctness of a map. An objective method of evaluating maps would enable the quality of existing maps to be assessed and the quality of potential maps to be predicted. Such a facility would allow geneticists to make an objective appraisal of new and old maps.

In order to develop a system for assessing maps, four main tasks were undertaken. Firstly, a series of discussions and informal interviews were held with geneticists in an attempt to identify what they regard as the "good" and "bad" features of maps and to find out what rules of thumb they use for map assembly. After the knowledge elicitation stage several published maps and the data from which they were generated were analysed to gain insight into the number and types of discrepencies present in maps. Given the published maps, the data expected from them was calculated. The expected data was compared with the observed experimental data and anomalies were highlighted.

Having taken these two approaches to gather information regarding map characteristics, some mechanism was required to place a score on the characteristics and quantify the process. This step was carried out with the aid of a questionnaire. In the questionnaire the geneticicst was asked to rate the severity of different kinds of map discrepancies on a scale from one (no problem) to five (critical problem). The geneticist was then asked to rank maps in terms of the number of discrepencies that would be tolerated in "ideal","good","adequate" and "unacceptable" maps. On completion of this exercise, it was apparent that there were several criteria by which maps were judged. It was proposed that the evaluation of a map should be made up of a series of sub-evaluations covering as much of the information contained in the questionnaire as possible.

Three types of sub-evaluations have been developed. Evaluation 1 assesses a map on how well the expected single digest data fits into the proposed map. Evaluation 2 assesses a map on how well the expected double digest data fits into the proposed map. The third evaluation considers the proposed map as a whole and assesses how well the lengths of the double digest data fit into the map. While each sub-evaluation is not considered appropriate as the sole means of evaluating maps, each has a significant contribution to make to the overall evaluation function. The evaluation function generates a score indicating the goodness of fit between the proposed probe map and the experimental data. A map which gains the highest score is called an "ideal" map. "Good" maps and "adequate" maps are maps that contain a certain amount of error but the error is tolerable given the limitations of the experimental process. Ideal, good and adequate maps are all maps that the geneticist would wish to consider. Maps classed as "unacceptable" contain too many errors and would not be considered.

6.1.3 Setting Parameter Values

There are established parameter settings described in the literature for traditional GAs [16], however, finding good settings for non-binary representations is not a trivial task [31]. It is recommended that appropriate settings for non standard GAs be derived by experimentation [27].

6.1.4 Generating The Initial Population

The HGA uses the domain specific information in the form of the experimental data in order to generate maps. A "probe map builder" has been developed which takes as input items of probe data (as shown in table 1) selected at random and creates legal probe maps for the initial population.

6.1.5 Selecting A Reproduction Technique

Various reproduction techniques were investigated, the best results were obtained using a steady-state replacement without duplicates [32]. Here, solutions generated which are already present in the population are discarded. This ensures that all members of a population are different.

7 Results

The HGA was applied to the experimental data contained in table 1. (This data was chosen as a test set as it represents a small, manageable number of probes and it is self-contained, in that all combinations of double digests were performed). The number of "acceptable" maps generated for the PI probe is shown in figure 8. (An acceptable map is a map which scores as "ideal", "good" or "adequate" using the evaluation function.) In order to see how well the evalution function was operating, each of the sub-evaluations was carried out separately (E1,E2,E3) then applied together (E123). Maps were evaluated on how well the single digest data fitted, how well the double digest data fitted and how well the total data fitted. The average fitness of the population over the 100 trials is shown in figure 9 along with the average best map score.

The parameters for the HGA were held constant for each of the runs. The results shown indicate the average of twenty runs.

8 Discussion

Looking at the results for PI shown in figure 8, evaluating on the fit of the single digest data (evaluation 1) is a very broad evaluation. Almost all of the maps generated after 100 trials would be considered as acceptable maps based on this evaluation alone. It is slightly harder for maps to score well using evaluation 2, evaluating on the fit of the double digest data. In this case only 90% of the maps generated after 100 trials would be considered acceptable. Evaluation 3, evaluating on the fit of the total data, resulted in a further 10% reduction in the number of acceptable maps. In all three cases, further information concerning fragment lengths was required to be taken into account to identify feasible maps. For example, an ideal map according to evaluation 1 may turn out to be an unacceptable map according to evaluation 3. Each of the evaluations scores different map characteristics and in order to generate feasible maps, all three evaluations must be applied together. When all three evaluations were applied together the number of acceptable maps for the PI probe after 100 trials was approximately 20% of the total population.

Figure 8 - Graph showing the number of acceptable maps for PI generated using evaluations1, 2, 3.

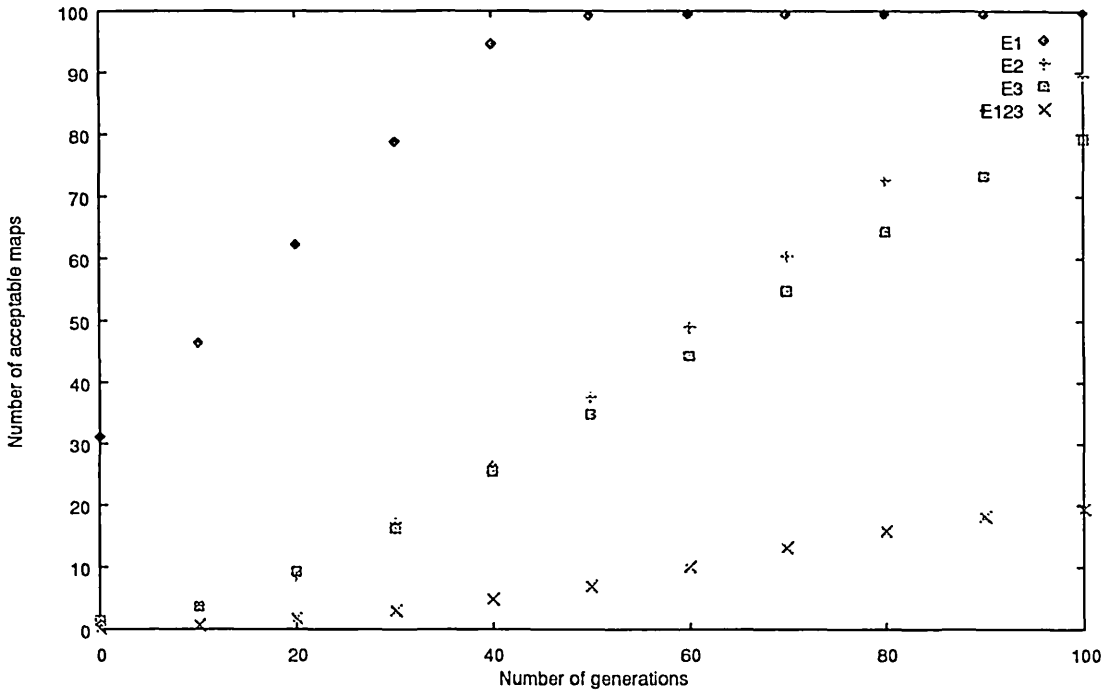
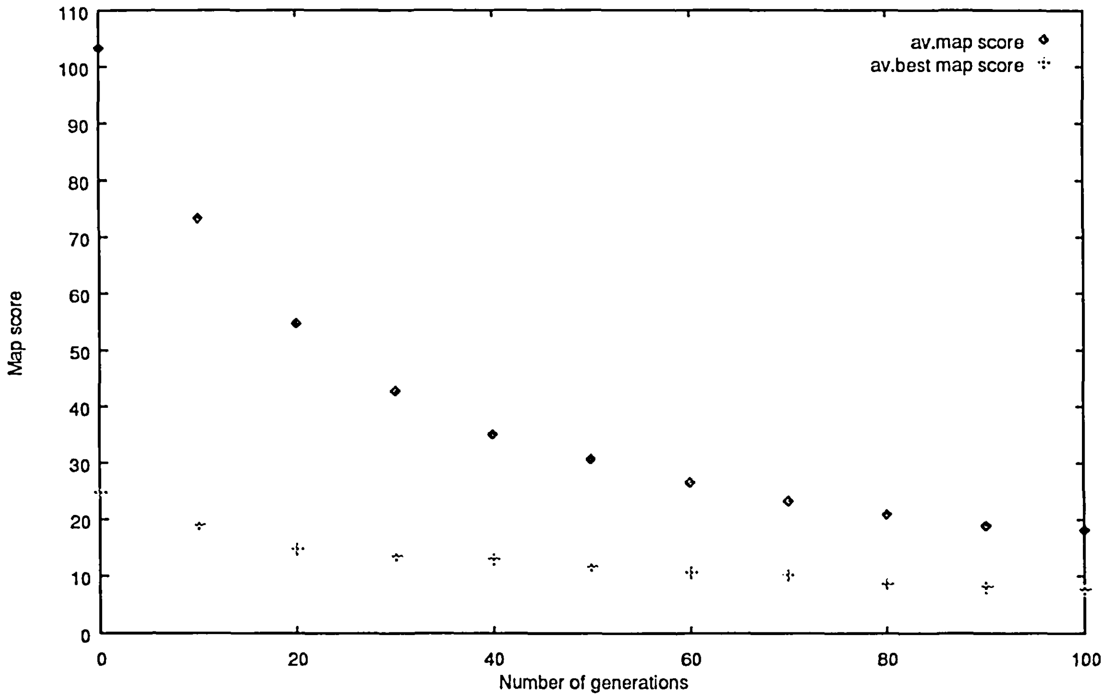


Figure 9 - Graph showing the av. map score and av. best map score for PI using evaluations 1,2 and 3 together.



The average map score and average best map score shown in figure 9 indicate how the fitness of the population improved over time. An acceptable map is a map which scored 9 or less. An ideal map is a map which scored 0.

The total number of acceptable maps generated over the twenty runs was calculated for each of the three probes and is shown in table 2.

Table 2 - The total number of acceptable maps generated for PIL, PI and AACT using evaluations 1, 2 and 3 together.

	Probe		
	AACT	PIL	PI
No. of acc. maps	134	143	426

One of the aims of using a HGA based system was to generate a small subset of feasible maps. The number of acceptable maps found using all three evaluations was still large. When all the acceptable maps were analysed, it was found that many maps had the same number of cut sites in the same positions however the value of the cut site (ie whether it was a complete cut site or a partial cut site) differed. It was proposed that maps could be reduced to a template form where a map template represented a precise number and ordering of cut sites but did not take into account the value of the cut sites. By using templates, the number of acceptable maps for each of the three probes should be reduced. The concept of map templates was introduced into the HGA through the reproduction technique. The reproduction technique was modified to disallow map templates which were duplicates of current templates in the population. The steady-state-without-duplicates technique became a steady-state-without-duplicate-templates(SSWDT) reproduction technique.

When the HGA was run using the SSWDT technique and the same parameters as set before, the total number of acceptable templates found is shown in table 3. Templates of the best maps arrived at by the geneticist were amongst the acceptable maps found for the probes. Some of the ideal templates generated by the HGA appear to represent a better fit of the experimental data than the geneticists map and this is being investigated.

Table 3 - The total number of acceptable map templates (ideal, good and adequate) generated for PIL, PI and AACT using evaluations 1, 2 and 3 together.

HEAV3 [illegible]

	Probe		
	AACT	PIL	PI
No. of ideal templates	4	6	5
No. of good templates	11	12	12
No. of adequate templates	41	23	32
Total no. of maps to be considered	56	41	49

The objective evaluation function appears to be functioning well in generating groups of acceptable maps for the individual probes. Work is commencing on the second stage of the project which involves developing a suitable optimisation algorithm to merge the separate probe maps together.

9 Summary

This paper has described the development of a hybrid GA approach to the problem of building DNA maps from the results of experimental data. An objective system for scoring maps has been devised. Results have been presented which indicate that it is successful in discriminating between good and bad maps. The objective scoring system has been used by a modified form of GA to generate a group of acceptable probe maps for a particular data set.

10 References

[1] L.Hunter(Ed), Artificial Intelligence and Molecular Biology, AAAI Press 1993.

[2] Molecular Bioinformatics, Proc. IEE Colloquium on Molecular Bioinformatics. Digest No. 1994/029, IEE London, 1994.

[3] M.Stefik, Inferring DNA Structures from Segmentation Data, Artificial Intelligence, Vol 11, Ps 85-114, 1978.

[4] W.R.Pearson, Automatic Construction of Restriction Site Maps, Nucleic Acids Research, Vol 10, No. 1, 1982.

[5] M.Shifman, P.Nadkarni, P.Miller, Interactive, Graphical, Computer-Based Tools for Storing and Constructing Pulse Field Gel Maps, Abstracts of Papers Presented at the 1992 Meeting on Genome Mapping and Sequencing, Cold Harbor Laboratory, New York, Published by Cold Spring Harbor Press, 1992.

[6] J.W.Fickett, M.J.Cinkosky, Optimizing Maps to Fit Experimental Data, Abstracts of Papers Presented at the 1992 Meeting on Genome Mapping and Sequencing, Cold Harbor Laboratory, New York, Published by Cold Spring Harbor Press, 1992.

[7] J.H.Holland, Adaptation in Natural and Artificial Systems, Ann Arbor: The University of Michegan Press, 1975.

[8] J.D. Walker, P.E. File, C.J. Miller, W.B. Samson, The GENIE Project: A GA Application To A Sequencing Problem In The Biological Domain, Proceedings of the International Conference On Artificial Neural Nets and Genetic Algorithms held in Innsbruck, ed. R.F. Albrecht, C.R. Reeves, N.C. Steele, 1993.

[9] J.D. Walker, P.E. File, C.J. Miller, W.B. Samson, A Hybrid GA Application To A Genetics Sequencing Problem, in [2].

[10] L.Sefton, G.Kelsey, P.Kearney, S.Povey, J.Wolfe, A Physical Map of the Human PI and AACT Genes, Genomics 7, 382-388, 1990.

[11] K.A.DeJong, An Analysis of the Behaviour of a Class of Genetic Adaptive Systems, (Doctoral Dissertation, Department of Computer and Communication Sciences, University of Michigan). Dissertation Abstracts International 36(10),5140B. (University Microfilms No. 76-9381), 1975.

[12] A. Brindle, Genetic Algorithms for Function Optimisation, Doctoral dissertation, University of Alberta, Edmonton, 1981.

[13] D.E.Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning, Addison Wesley, 1989.

[14] K.A.DeJong, Learning with Genetic Algorithms: an overview, Machine Learning, 3(2), 121-138, 1988.

[15] K.A.DeJong, W.M.Spears, Using Genetic Algorithms to Solve NP-Complete Problems, Proceedings of the Third International Conference on Genetic Algorithms, ed. J.D.Schaffer, ps 124-133,1989.

[16] J.D.Schaffer, L.J.Caruana, L.J.Eshelman, R.Das, A Study of Control Parameters Affecting Online Performance of GAs for Function Optimisation, Proceedings of the Third International Conference on GAs pp 51-60. San Mateo, CA:Morgan Kaufmann Publishers, 1989.

[17] A.D.Bethke, Genetic Algorithms as Function Optimisers,(Doctoral dissertation, University of Michegan), Dissertation Abstracts International 41(9), 3503B (University Microfilms No. 8106101), 1981.

[18] J.Bosworth, N.Foo, B.P.Zeigler, Comparison of Genetic Algorithms with Conjugate Gradient Methods, (CR-2093) Washington, DC: National Aeronautics and Space Administration, 1972.

[19] D.E.Goldberg, Computer-aided Gas Pipeline Operation using Genetic Algorithms and Rule Learning, (Doctoral Dissertation, University of Michigan), Dissertation Abstracts International, 44(10), 3174B (University Microfilms No. 8402282), 1983.

[20] L.Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold Publishers, 1991.

[21] D.E.Goldberg, R.Lingle, Alleles, Loci, and the Travelling Salesman Problem, Proceedings of an International Conference on GAs and their Applications, 154-159, 1985.

[22] J.J.Grefenstette, R.Gopal, B.J.Rosmaita, D.Van Gucht, Genetic Algorithms for the Travelling Salesman Problems,Proceedings of an International Conference on GAs and their Applications, 160-168, 1985.

[23] I.M.Oliver, D.J.Smith, J.R.C.Holland, A Study of Permutation Crossover Operators on the Travelling Salesman Problem, Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on GAs, 224-230, 1987.

[24] D.Whitley, T.Starkweather, D.Fuquay, Scheduling Problems and Travelling Salesman: the Genetic Edge Recombination Operator, Proceedings of the Third International Conference on GAs, San Mateo,CA: Morgan Kaufmann Publishers. pp133-140, 1989.

[25] D.Whitley, T.Starkweather, D.Shaner, The Travelling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination, Handbook of Genetic Algorithms, ed L. Davis, Van Nostrand Reinhold Publishers, New York, 1991.

[26] L.Davis, Job Shop Scheduling with Genetic Algorithms, Proceedings of an International Conference on GAs and their Applications, 136-140, 1985.

[27] G.Syswerda, Schedule Optimisation Using Genetic Algorithms, Handbook of Genetic Algorithms, ed L.Davis, Van Nostrand Reinhold Publishers, New York, 1991.

[28] D.Whitley, Introduction to Foundations of Genetic Algorithms 2, edited by L.Darrell Whitley, Morgan Kaufmann Publishers, San Mateo, California,1993.

[29] Z.Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1992.

[30] B.R.Fox, M.B.McMahon, Genetic Operators for Sequencing Problems, Foundations of GAs, Rawlins G (ed), San Mateo, CA: Morgan Kaufmann Publishers, 1991.

[31] L.Davis, Adapting Operator Probabilities in Genetic Algolrithms, Proceedings of the Third International Conference on GAs pp 61-69. San Mateo, CA:Morgan Kaufmann Publishers, 1989.

[32] D.Whitley, GENITOR : a different GA, Proceedings of the Rocky Mountain Conference on AI. Denver, Colorado,1988.