Inferring DNA Structures from Segmentation Data

Mark Stefik

Computer Science Department, Stanford University, Stanford, CA 94305, U.S.A.

Recommended by N. S. Sridharan

ABSTRACT

The analysis of DNA structure from restriction enzyme segmentation data has been viewed by molecular geneticists as one of their simpler analysis problems. This paper treats segmentation problems as a case study in the selection between data-driven and model-driven hypothesis formation. The main purpose of this paper is to set forth some useful considerations for selecting a problem solving approach according to the characteristics of a domain. The case study illustrates why an exhaustive model-driven approach, which operates primarily by ruling out all the wrong answers, is a good approach for this domain. A program called GA1 solves segmentation problems using techniques similar to those used in the DENDRAL program.

Contents

1. Introduction

A common task in molecular genetics laboratories is the analysis of DNA structure from restriction enzyme segmentation data. This task is one of the simplest, although time-consuming, analysis tasks in molecular genetics. It is described in Section 2.

The thrust of this paper is to examine the import of selecting different problem solving approaches. This paper characterizes the segmentation problem domain and compares a data-driven strategy to a model-driven strategy for forming

Artificial Intelligence **11** (1978), 85–114

hypotheses. The model-driven approach has been used in several Artificial Intelligence applications including chemical analysis [2], machine learning of mass spectroscopy rules [8], and machine vision [9]. This paper seeks:

(1) to quantify some advantages of this approach for this domain,

(2) to identify some general characteristics of problem domains on which the selection of this approach is based and

(3) to discuss some representation and programming ideas which are useful for implementing it.

It will be shown that the best way to determine the answers in segmentation problems is to use an exhaustive model-driven approach which operates primarily by ruling out the wrong answers. This research has led to the development of an application program named GA1[1] which usually solves problems in a few seconds and requires less laboratory work than is customarily done.

A secondary benefit from the explication of the problem-solving knowledge is that some new laboratory techniques have been proposed. They are described in Appendix I.

2. Segmentation Problems

One of the hurdles in understanding Artificial Intelligence applications is the terminology of task domains. An attempt has been made to minimize the amount of technical terminology in the following example so that it will be easy to read. A few genetics terms will be introduced as needed.

2.1. A sample problem

To describe a segmentation problem, it is easiest to begin with the answer. Fig. 1 illustrates a circular DNA structure. Each of the numbers represents the size (in arbitrary units) of a segment in a DNA structure. The labels (e.g. Eco RI) represent enzyme recognition sites. When an enzyme is used to cut or "digest"[2] a DNA structure, it will cut at specific recognition sites.[3] The problem solution describes a sequence of segments separated by recognition sites which best explains the laboratory data. These data are produced by cutting the whole DNA structure with enzymes and measuring the possibly overlapping segments.

An analogy might make the problem easier to visualize. The segment lengths may be likened to lengths of chain and the recognition sites may be likened to locks with keys. The solution to the sample problem is analogous to a circular

[1] The name "GA1" was chosen to indicate that this program is the first in a series of "Geneticist's Assistants."

[2] Digestion refers to the action of breaking apart the DNA molecules. The terminology "enzyme digestion" is a carry over from the first enzymes that were studied—those involved in the digestive system.

[3] This recognition of sites corresponds to a physical template matching process. Each enzyme cuts when it recognizes a specific sequence of nucleotides in the DNA molecule.

chain composed of smaller chains locked together in a particular order (Fig. 2). In these terms the solution to the sample problem has six locks and three different keys (B, H, E). At the beginning of the experiment we are given a box of identical circular chains. We are not permitted to see the chains, but may dispatch an assistant with one or more keys (and possibly a time limit) to go to the box and unlock the corresponding locks. The assistant may report back the lengths of the chain pieces that are left after the unlocking operations. The goal is to infer the

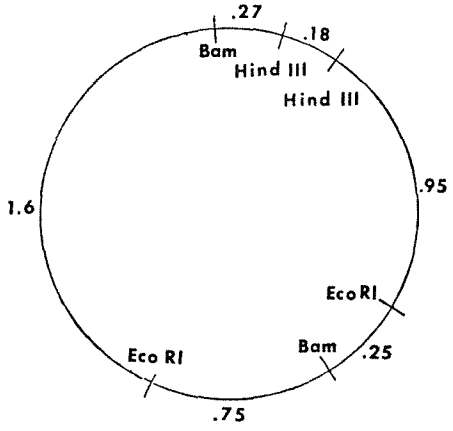


FIG. 1. Solution to the sample problem. The numbers indicate masses of the DNA segments (measured in megadaltons) and the labels (Bam, Hind III, Eco RI) show the recognition and cutting sites for the enzymes.

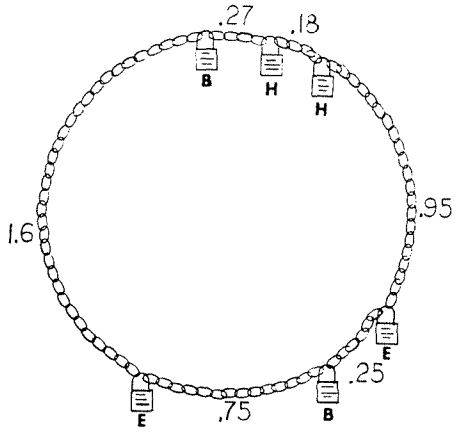


FIG. 2. Locks and chains analogy. The locks sites can be opened with appropriate keys and correspond to enzyme recognition sites. The chains correspond to linear segments of DNA molecules which can be observed when the locks have been opened. The goal is to infer the configuration of the chain from the record of segments observed after different kinds of unlocking operations.

original configuration of the chains from the pieces that are observed after different unlocking operations.

Returning to genetics terminology, the enzymes may be used singly (called 1-enzyme digests) or in combination (n-enzyme digests) to cut an unknown DNA structure. Table 1 shows the segment sizes that were measured after enzyme digestion in the sample problem. These are the input data for the segmentation problem. The goal is to infer plausible DNA structures from the digest data.

TABLE 1. Input Data for the Sample Problem[a]

Enzyme(s)	Segments observed after enzyme digestion
Hind III	3.82 .18
Bam	2.35 1.65
Eco RI	3 1
Hind III & Bam	2.35 1.2 .27 .18
Hind III & Eco RI	1.87 1.0 .95 .18
Bam & Eco RI	1.6 1.4 .75 .25

[a] Segment sizes are expressed in megadaltons—a unit of mass approximately equal to $1.66\ 10^{-18}$ grams. The resolution of measurement for this experiment was one percent. These data were provided by Jerry Feitelson.

Because of combinatorics, the sample problem was considered to be too difficult to solve from the available data. An additional digest involving all three enzymes was subsequently performed and an answer was determined after an hour's work. The manual solution did not include testing the answer for uniqueness. GA1 subsequently solved this problem in about one second and verified that the answer was unique. The program was also able to solve the problem without the additional digest data in approximately three seconds.

2.2. Genetics background

Much of the recent progress in molecular genetics relies on techniques for structural analysis of DNA. Several analytical techniques are available for determining gross or fine structural information. The techniques described in this paper are useful for determining structure at a level that is coarser than nucleotide structure but finer than gene structure. Study of structure on this level is a common step in many experiments involving recombinant DNA [1]. These experiments involve careful splicing together of DNA from different sources to study gene expression in microorganisms. Most of these experiments involve splicing the genes into small linear and circular DNA molecules termed "vectors." Structural analysis is performed both before and after the splicing operations.

Different kinds of enzyme digestions are used for the analyses. An enzyme digestion may be either complete or incomplete. In a complete digestion, all sites in the DNA molecules which are recognized by the enzyme are cut. "Incomplete

digestion" means a limited application of an enzyme. After an incomplete digestion, the sample will contain a mixture of segments resulting from the random cleavage of the DNA molecules at the recognition sites. Under ideal conditions, segments resulting from every possible combination of cuts at the recognition sites can be observed.

Complete digests are useful in determining the composition of a structure in terms of enzyme sites and segments. The number of sites can be determined from the number of segments of a complete digest combined with the knowledge of whether the structure is linear or circular. Although a complete digest by one enzyme determines the sizes of the segments and the total molecular weight, it does not determine the order of the segments. An incomplete digest can be helpful in determining the order of segments since segments corresponding to the sum of neighboring segments will appear in the digest. This is explained further in Section 3.2.1.

Digests may also be described as 1-enzyme or n-enzyme digests depending on the number of enzymes involved. In an n-enzyme complete digest, the DNA will be cleaved at all of the sites recognized by any of the restriction enzymes used in the digest. Enzymes are chosen so that particular genes are cut or not cut. When segment sizes are similar in size, digestion data may not determine segment orientation and placement unambiguously. In such cases, additional enzymes are sometimes introduced which may cleave the segments asymmetrically. The 3-enzyme digests constrain the problem by indicating which of the segments in the 2-enzyme digests have uncut sites. Most experiments involve only four or five restriction enzymes and most digestions involve no more than three enzymes.

After the digestions have been carried out, the lengths or masses of the segments are measured. There is always a limiting resolution in this measurement—given as one percent in the sample problem. This may be as high as ten percent in some experiments. Because of the ubiquitous requirement to specify the resolution of measurement, it gets rather tiresome to repeat these qualifications. For the sake of brevity, we will say that two measurements are "equal" when they are "equal within tolerance." Similarly we will refer to the "sum of segments" to mean the sum of their masses or lengths.

2.3. Combinatorics of the sample problem

The number of possible answers in segmentation problems depends on the number of segments in the digests and on the resolution of measurement in the experiment. GA1 has been applied to problems having several billion potential solutions in the solution space. It will be seen that the sample problem, which is much simpler than most of the segmentation problems to which GA1 has been applied, has a solution space of over two million possible answers.

The description of the solution space for the sample problem is given in three steps. First, the 1-enzyme digest results are used to develop a template

(Fig. 3) for the possible answers. Then an argument is given that only segments from the 2-enzyme digests need be used in filling in the template. Finally, it will be shown that some of the segments in the 2-enzyme digests can be ignored.

The 1-enzyme digests in the sample problem each yield two segments after digestion. For circular structures, this means that each enzyme cuts the molecule exactly twice. Since the recognition sites were known to be unique, the sample problem must have six recognition sites—two for each of the three enzymes. (If it had not been known before the experiment that the sites were unique, this could be inferred from the fact that the segments in 1-enzyme digests are different for the different enzymes). For a circular structure, the number of sites must equal the number of segments. This means that the solution to the sample problem must have six sites and six segments as shown in the template in Fig. 3. This template defines the solution space.

$$\begin{array}{cccccc} \text{Seg}_1 & \text{Seg}_2 & \text{Seg}_3 & \text{Seg}_4 & \text{Seg}_5 & \text{Seg}_6 \\ \text{site}_1 & \text{site}_2 & \text{site}_3 & \text{site}_4 & \text{site}_5 & \text{site}_6 \end{array}$$

FIG. 3. Template for hypotheses in the sample problem. The space of possible hypotheses may be generated by assigning sites and segments to this template.

Each segment in a solution is bounded by two recognition sites—one at each end. Thus, every completely digested segment will be severed from the structure when these two enzymes are applied and must appear in some 2-enzyme complete digest—no matter how many enzymes are used in the experiment. In the data from the sample problem, twelve segments appear in the 2-enzyme digests. Thus, every solution corresponds to an arrangement of six of the twelve segments in the six positions of the template.

Finally, it is possible to show that no hypothesis need be considered which uses any of the segments in the sample problem more than once. Reasoning about the uniqueness of segments generally involves the measurement tolerance. In this problem, the only segment which is repeated in the 2-enzyme digests is the 0.18 segment. This segment appears in all of the digests involving the enzyme Hind III. Since this segment appears in the Hind III complete digest, it has a Hind III site at each end. If this segment had an uncut site for some other enzyme, then a smaller segment would appear in one of the 2-enzyme digests. The absence of any smaller segment allows us to trim down the number of possible segments to eleven.

The order of the sites and segments on the template is not known. This yields

$$\frac{11!}{6!5!} \times 6! \times \frac{6!}{2! \times 2! \times 2!} \times \frac{1}{12} = 2\,494\,800 \text{ structures.}$$

The first term is the number of ways to select six of the eleven unique segments; the second term counts the permutations of assigning these segments to Seg 1

through Seg 6; the third term counts permutations of the six sites to Site 1 through Site 6 (allowing for the fact that each enzyme must have two recognition sites). The division by 12 accounts for circular symmetry since each structure is represented twelve times—allowing for six starting places (Seg 1 through Seg 6) and two possible orientations. A similar analysis can be given for problems where the hypothesized molecule is linear.[4]

If an additional laboratory step is performed in the sample problem—the simultaneous digestion by all three enzymes—the number of potential segments to be placed is reduced from eleven to six. This extra laboratory step reduces the number of possible structures to 5400. The manual solution to the sample problem used the extra laboratory step to reduce the combinatorics. The existence of programs like GA1 means that the trade-off can be made in the other order to save laboratory time.

3. The Data-driven Approach

Laboratory geneticists typically do not use a formalized algorithmic approach to solving structural problems. Often they examine the data to see what structures are suggested. The first (and, as it turns out, inadequate) approach, termed the "data-driven" approach, attempts to capture this idea. This approach is described here because its faults help clarify the requirements for a better method.

3.1. Data-driven approaches in general

The term data-driven will be used to describe a strategy for inferring hypotheses from data. This approach is also called bottom-up and it will be contrasted with a model-driven approach which uses a generator (based on a model of the solution space) to propose hypotheses.

3.2. Specifics of the data-driven approach

There are two main methods for inferring structure from data in this domain. The first uses the information from incomplete digests to propose that specific segments are contiguous in the solution. The second method uses 2-enzyme and 3-enzyme digest information for this purpose.

3.2.1. *Reasoning from incomplete-digest data*

Since an incomplete digest employs only a limited application of an enzyme, many structures in the sample will have some of their recognition sites still intact. When the resulting segments from an incomplete digest are compared to those from a complete digest by the same enzyme, many segments can be observed in the

[4] For linear molecules, the ends of the molecule provide unique origins. In such cases the analysis may be simplified by decomposing it into separate analyses for each pair of enzymes. The separate solutions may then be merged. This strategy is seldom useful for circular molecules because such unique origins are not usually available.

incomplete digest which are equal to the sum of various segments from the complete digest. This suggests the incomplete digest rule:

> If a segment from the incomplete digest for enzyme E1 is equal to a sum of a set M of segments from the complete digest by the same enzyme, then the segments in M are probably contiguous in the structure and are separated by sites for enzyme E1.

For example, suppose that the complete digest by an enzyme includes the segments 3, 4, and 6, among others, and the incomplete digest by the same enzyme has a segment of size 13. The incomplete digest rule could be used to propose that the segments 3, 4, and 6 are contiguous.

3.2.2. *Reasoning from n-enzyme digest data*

The incomplete digest rule does not tell how to combine digestion results when several enzymes are involved. Complete digests by different enzymes yield different decompositions of the starting structure. When a structure is digested to completion by several restriction enzymes, the segments produced after digestion by the first enzyme are cut at the sites recognized by each subsequent enzyme.

The simplest case of the *n*-enzyme rule is the 2-enzyme rule.

> If a segment from the 1-enzyme digest for enzyme E1 is equal to a sum of a set M of segments from the 2-enzyme digest by enzymes E1 and E2, then the segments in M are probably contiguous and are separated by sites for E2.

This rule may be explained by an example. Suppose there is a segment of size 12 in the complete digest by enzyme A and segments of sizes 2, 3, and 7 in the complete digest by enzymes A and B. The *n*-enzyme rule says that we may take this as evidence that the small segments are contiguous in the structure and separated by sites for enzyme B. The *n*-enzyme rule does not tell us the order in which the three segments appear.

4. The Model-driven Approach

This section describes the model-driven approach used by GA1 as well as many of the techniques used in implementing it. A comparison of data-driven and model-driven approaches and a discussion about determining their suitability for different domains is given in Section 5.

4.1. Model-driven approaches in general

Feigenbaum [5] uses the term "model-driven"[5] to describe an approach which uses a generator based on a model of the solution space to propose hypotheses. A generator is a procedure for enumerating the elements of the solution space. It is called exhaustive or complete when it exhaustively enumerates the solution space. When the solution space is very large, an exhaustive model-based approach depends on the use of effective methods for eliminating most of the hypotheses proposed by the generator. This approach is also called the Generate-and-Test paradigm. When a generator enumerates the space by refining general hypotheses, the approach may be called top-down.

[5] The term model-driven should not be confused with "model-based", which refers to the use of a model for various purposes in problem solving.

4.2. Specifics of the model-driven approach

The following sections discuss several aspects of the GA1 program. In addition to the generator, which is central to the model-driven approach, data correction, hypothesis elimination, and candidate evaluation will be described. Several of these topics will be considered again in Section 5.

The main steps in GA1 are shown in Fig. 4. The first three steps acquire and correct the data and set up constraints for the exhaustive generation of hypotheses. The recursive generation cycle is where most of the computational work is done. All possible hypotheses are systematically proposed and most of them are eliminated. At each iteration through the loop, a site or segment is placed and the relevant pruning rules are applied. If the class is not pruned, the cycle is repeated for the next site or segment recursively. Backtracking commences whenever a hypothesis is accepted or the current class of candidates can be ruled out. Thus, the algorithm does not propose complete candidates and then rule them out; the rules are applied within the generator to eliminate branches of the solution space so that hypothetical structures are usually discarded before they are completely specified.

1. Acquire input data.
   (Digest data, topology, tolerance, and other constraints.)
2. Check and correct input data.
   (Data checking rules.)
3. Determine generator constraints.
   (Potential sites and segments.)
4. Generation Cycle: Alternately place a site and then a segment. (This assures that every hypothesis is considered.)
5. Apply canonical form rules.
   (This assures that each hypothesis is considered at most once.)
6. Apply pruning rules.
   (Eliminates contradictory classes of candidates.)
7. Evaluation of remaining hypotheses.

FIG. 4. Computational steps in GA1. The generation cycle in steps 4 through 6 recursively assigns sites and segments to a template (see Fig. 3) to specify hypotheses. Backtracking for alternate assignments occurs whenever a hypothesis is accepted or a class of hypotheses is rejected.

In the sections which follow, several ideas will be discussed which are important for model-driven approaches. Among these are the use of canonical forms, the use of pruning rules based on data to limit the generation process, and the use of rules to check and sometimes correct the input data. Many of the ideas and techniques are similar to ideas used in the DENDRAL program [2], which generates chemical structures satisfying chemical constraints.

4.2.1. *Data-checking rules*

Three factors complicate the use of segmentation data from the laboratory.

1. *MISSING DATA.* There are sometimes segments missing from the laboratory data. When an insufficient quantity of a segment is present, it will not be observed. Without a strong model, missing data can be mistaken for negative data.
2. *INSUFFICIENT RESOLUTION.* Segments of nearly identical size may blur together and be observed as a single segment. This is caused by the limited resolution of measurement. This may cause segments to be missing which would appear in an "ideal" digest.
3. *EXTRANEOUS DATA.* These are extra segments which appear in the digest of a structure which would not appear under ideal digest conditions. There are several sources of extra segments in the digest. For example, there may be impurities, there may be a mixture of structures, or a supposedly complete digest may actually be incomplete. Without a strong model, extraneous data can be mistaken for positive data.

This section describes some of the rules which GA1 uses to check and sometimes correct data in segmentation problems.

> If a segment which appears in the complete digest for an enzyme fails to appear in the incomplete digest for that enzyme, it may be added to the list of segments for the incomplete digest.

This is an example of a rule which can repair the input data. Since the ideal results of an incomplete digestion include cutting every combination of the recognition sites for an enzyme, the combination of cutting all of the sites should be included and this corresponds to a complete digest. When these smaller segments fail to be observed, it is usually because the segment separation and measurement parameters have been optimized to accurately measure the larger segments.

Perhaps the most useful test for digest consistency is the molecular weight test. All of the sums of the segments for the complete digests by one or more enzymes should be equal. Each complete digest represents a decomposition of a structure into non-overlapping segments. If the sums are not equal, then there must be some missing or extraneous segments. This observation may be characterized as a conservation law for mass in the experiment. GA1 also has rules pertaining to the conservation of enzyme sites.

Conservation laws provide redundancy for data correction. When a discrepancy arises, it is often possible to use a "majority rules" logic to identify an inconsistent piece of data. The following rule illustrates this idea.

> If most of the complete digests of one or more enzymes yield a summed molecular weight of MW and there is a digest (termed the maverick digest) which predicts a molecular weight of MW′ where MW′ < MW, and if the maverick digest contains a segment of mass equal to MW–MW′, then hypothesize that the segment is an unresolved doublet which should appear twice in the maverick digest.

This rule incorporates the fact that segments of very close mass are sometimes unresolved and a molecular weight discrepancy can often indicate the cause. Even if no segment equal to the mass discrepancy was observed, it would be possible to hypothesize that a segment was missing. In that case, there might be less confidence in the conclusion because there would be no hypothetical unresolved doublet to explain the missing data.

In summary, GA1 uses a number of rules to detect and sometimes correct inconsistencies in the input data. This alleviates problems with extraneous and missing data. Rules which remove extraneous data are used before rules which fill in missing data. More rules for checking and correcting data are listed in Appendix II.

4.2.2. *Determining potential sites and segments*

Before the generation process can begin, GA1 must determine the potential sites and segments from which to build the hypotheses. This process determines constraints for the generator.

The set of potential sites may be inferred from the 1-enzyme complete digest data. Since each complete digest cuts the structure at each of the sites, the number of sites may be determined for each enzyme by counting the segments left after cutting. For circular structures, the number of recognition sites for each enzyme is equal to the number of segments appearing in the 1-enzyme complete digest. For linear structures, the number of sites is one less.

As discussed in Section 2.3, every segment which appears in an n-enzyme complete digest must appear in some 2-enzyme digest. GA1 creates its set of potential segments by collecting all of the segments which appear in any of the 2-enzyme complete digests.

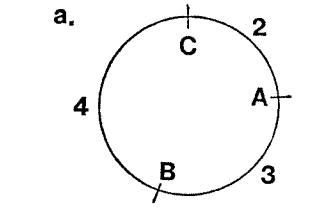
In special cases, this method for determining the set of potential segments is modified. If more than two enzymes are used, then many of the segments in the 2-enzyme complete digests will be unessential because they have internal sites for other enzymes. (It is not usually possible to tell which of the segments are essential without generating the hypothetical structures.) However, if all of the 3-enzyme digests are available, the essential segments will appear in the 3-enzyme digests but many of the non-essential segments will drop out. In other experiments where only one enzyme is used, the segments from the 1-enzyme complete digest are all that are needed. GA1 uses these criteria to select a minimum set of potential segments for the generation process.

An additional consideration is the number of times that a segment may appear in a generated structure. If it can be determined that a segment need appear only once in any hypothesis, the segment is termed "unique" and the size of the set of potential segments can be reduced. Rules for determining the uniqueness of segments are listed in Appendix II.

4.2.3. *Symmetry and a canonical form for structures*

Computing the equivalence of hypotheses is complicated when representation techniques permit more than one representation of a hypothesis. In segmentation problems, "canonical forms" may be defined that specify which of the equivalent representations of a hypothesis will be used.

The DNA structures for which GA1 has been designed have linear or circular topologies.[6] These structures may be conveniently represented as lists, as in Fig. 3, with some notation to indicate whether the structure is closed to form a circle. This representation designates one segment (Seg 1) as a "starting segment" for the structure. If this designation is arbitrary, there will be several equivalent representations for a given physical structure. Fig. 5 shows a simple DNA structure and two of its six possible representations as lists starting with a segment length.



b.

$$2\ A\ 3\ B\ 4\ C$$

$$3\ A\ 2\ C\ 4\ B$$

FIG. 5. (a) A simple circular DNA structure. The letters A, B, and C stand for enzyme recognition sites and the numbers are segment lengths.

(b) Two of its six representations as lists starting with a segment length. There are six representations because we can start with any of the three segments and read off the structure in either of two directions.

For linear structures there are two possible orientations—either end may be designated as the starting segment. For circular structures, any segment could be chosen as the starting segment and there are then two possible orientations once a starting segment has been selected.

GA1 uses similar canonical forms for linear and circular structures. Given a linear structure, GA1 must choose between the given list and its reversal. GA1 chooses the representation which minimizes the size of the starting segment. If the first segment is equal to the last segment, GA1 minimizes the second segment.

[6] Although DNA exists in many more complicated shapes, segmentation experiments deal almost exclusively with structures having these topologies or which may be idealized as having these topologies.

This process continues until GA1 has found an asymmetry at a segment or until the structure is completely symmetric in its segments. In the latter case, GA1 starts over again, comparing the alphabetic order of the restriction sites starting from the ends of the structure. Finally, if the structure is completely symmetric in both segments and restriction sites, it does not matter which of the two representations is chosen since they are indistinguishable.

For circular structures the process is similar. GA1 tries to minimize the starting segment. This is done by picking the minimum segment in the structure—except that several segments may be chosen if there is a tie. Once the starting segment is selected, GA1 tries to minimize the subsequent segments. Both orientations of the tied segments must be considered. As with the linear canonical form, ties between segment lengths are resolved by the alphabetic order of the site designations.

Given these definitions of canonical form, two representations in canonical form correspond to the same physical structure if and only if they are equal.

4.2.4. *Using canonical forms to limit generation*

GA1 generates all structural representations that can be put together from a given set of sites and segments. It does this by alternately placing, one at a time, first a site and then a segment—building all possible representations of structures. This recursive process involves bookkeeping of sites and segments during generation.

A desirable attribute of generators is that they be irredundant, that is, that they produce each element of the solution space exactly once. When an hypothesis has several possible representations, an irredundant generator should produce only the canonical one. Unfortunately, it is not always possible to create an irredundant generator. Given a complete generator of all possible representations, it is often possible to detect quite early that a class of hypotheses will not be in canonical form. GA1 does this using a number of canonical form rules which are described below. These rules are very effective at eliminating non-canonical hypotheses, so that they rarely need to be eliminated from the output of the generator. To make GA1 eliminate the remaining hypotheses having rare symmetries would slow down the common cases.

The rules for pruning structures not in canonical form are based on topological considerations for linear and circular structures. The following rule governs the selection of a first segment for a circular structure.

> If circular structures are being generated, only the smallest segment in the list of initial segments should be used for the first segment.

This rule follows immediately from the definition of circular canonical form. The next rule applies when the second segment is being placed.

> If circular structures are being generated and the second segment is about to be placed and it is the largest of the remaining segments, then this branch of the generation can be pruned.

The representations pruned by this rule cannot be in canonical form because the reverse orientation would generate a "more canonical" structure. In other words, consider the representation which would result from reversing the final list of segments and sites starting with the second segment. This corresponds to the same physical structure but the representation would have a smaller second segment and would therefore be more canonical. Considerations of combined rotations and reorientations can be used by rules to prune at deeper points in the generation process. These rules and others for linear structures, are listed in Appendix II.

4.2.5. *Pruning rules*

So far we have discussed how GA1 constrains its generation by limiting the output of the generator to canonical forms. When all redundant representations are eliminated, the output of the generator is still over two million structures for the sample problem. This section describes the use of pruning rules which use digest results to significantly reduce the number of candidates generated.

Experimental data often have missing or extraneous segments due to the difficulties and limitations of measurement. If data are incorrect, pruning rules may prune using incorrect information. For this reason, the sensitivity of each of the rules below to errorful data is important to consider. In the GA1 program, the amount of tolerable contradiction before a candidate is eliminated may be adjusted by the user (although it is typically set to zero). By setting this to a positive number a user may compensate for a small amount of errorful data. This will prevent a class of candidates from being discarded by a single errorful datum. This facility has not been used much in practice because the error-correcting rules have been adequate. For brevity in the rules which follow, we will say "this branch may be pruned" when we mean that the accumulative number of contradictions is increased by one. The current class of hypotheses is then eliminated if this is too great.

The first step in the generation process is the selection of the first segment. The use of the canonical forms to limit the selection of the first segment has already been mentioned. The following rule is also useful in selecting the first segment for linear structures.

> When linear structures are being generated, only segments which appear in some 1-enzyme complete digest should be considered for the first segment of the structure.

The rationale is that the first segment is at an end of a linear structure and therefore has only a single restriction site. Thus the segment should appear not only in the n-enzyme digests, but also in some 1-enzyme digest. This rule will fail to prune extraneous segments which happen to appear in some 1-enzyme digest. It would also prune out a correct structure if the end segment could not be observed in a complete digest. In both cases, however, the data-checking rules would first detect a molecular weight inconsistency.

One of the important structural constraints which can be inferred from the digest data is which sites are permissible for terminating the segments in the

structure. The following definition of "allowable termination sites" and rule illustrate this.

> The allowable termination sites for a segment are the recognition sites for those enzymes in the 2-enzyme complete digests in which the segment appears. (If there is only one enzyme, then only its recognition sites are allowable.)

The following rule uses the list of allowable sites.

> If a segment is being placed and the previous site is not one of the allowable sites for the segment, then this branch of the generation may be pruned.

Similar rules are applicable when a site is about to be placed. A stronger definition—specifying conditions for when certain termination sites must be used—is given in Appendix II.

Pruning rules based on allowable or required termination sites combine knowledge about a previous structural element in the hypothesis with the observed data from the digests to detect a pruning condition. It is possible to include more context, that is, more of the hypothesis in this determination. Several rules in GA1 do this by using the summation of neighboring segments to determine their applicability.

Constraints should generally be tightened as early as possible in the generation process. The following rule involving the first segment of circular structures illustrates this.

> If circular structures are being generated and the first segment in a branch of the generation is unique and appears in a 1-enzyme complete digest for enzyme E1, then a recognition site for E1 can be placed in front of the first segment. (In other words, that site may immediately be placed as the last site in the structure.)

The rationale is that the segment from the 1-enzyme digest has the same recognition site at both ends. One of these sites will eventually have to appear at the far end of the structure during the generation process. Allocating that site immediately reduces by one the number of sites that need to be considered for placement at all of the intermediate steps in the generation.

4.2.6. *Evaluating the candidate structures*

After the generation process there will often be more than one candidate. This is expected when the evidence is insufficient to discriminate among the hypotheses, but there are other possible causes. First, extraneous segments or coincidental sums can cause some of the pruning rules not to be applied. Second, the pruning rules do not use all of the evidence in all possible ways. As we go beyond the pruning rules implemented in GA1, the rules become increasingly complex and specialized. It becomes more difficult to prove the correctness of such rules and to ensure that they are faithfully represented in the program. For these reasons there is an evaluation phase for GA1. For the problems tested so far, the number of candidates left after pruning for evaluation is usually less than five.

For each candidate to be evaluated, GA1 predicts the ideal digest results. Then it compares the predicted results with the observed laboratory results and computes an *ad hoc* score to characterize the quality of the match.

This score is a weighted sum over the segments in all of the available laboratory digests.

$$\text{SCORE} = \sum_{\substack{\text{All of the segments} \\ \text{in the digests}}} 2 \times (\text{AGREE} - \text{MISSED}) - \text{EXTRA}$$

The scoring function above assigns credits for predictions which agree with the lab digest and penalties for disagreements. The weighting reflects the fact that segments are more likely to be missing from a laboratory measurement than to be extraneous. Thus, a hypothesis should be penalized more for failing to predict an observed segment than for predicting an unobserved segment.

We have not done extensive tests with the scoring function in an attempt to optimize it. The quality of the laboratory data is such that it has very close agreement with the ideal data for a correct structure. In addition, the number of disagreements between the ideal digests of two different structures is usually considerable and increases rapidly as the differences are increased. Careful tuning of the evaluation coefficients has not been necessary.

5. AI Issues: Choosing a Solution Method

One of the motivations for this case study is to gain a perspective on how the problem-solving goals and the logic of a domain determine the applicability of alternative problem-solving methods. Some important general characteristics of domains and methods are listed in Table 2. This section examines characteristics of the segmentation problem domain and shows how they are important in the selection of a problem-solving approach.

In this section two approaches to hypothesis formation—model-driven and data-driven will be contrasted. In Section 4 we introduced the term model-driven to describe the method of using an hypothesis generator based on a model of the solution space. The term data-driven has been used to describe a bottom-up strategy data for inferring hypotheses from data without a generator. The idea is to propose only those candidates which are suggested by the data. One way to view the difference between these two approaches is in terms of indexing: the data-driven approach steps through the data-space to propose hypotheses and the model-driven approach steps through the solution-space checking hypotheses against the data.

The model-driven and data-driven approaches may be viewed as extreme points in a spectrum of approaches. Many problem-solving programs contain elements of both approaches. A data-driven approach to problem solving is often intuitively

TABLE 2. Characteristics of domains important in selecting problem solving approaches[a]

Characteristics of Data	
* Positive data (Data are observed)	* Negative data (Absence of data is observed)
Error-free data	* Errorful data
* Redundant data	Sparse data
Characteristics of Inferences	
* Positive inferences (Hypothesis-forming)	* Negative inferences (Hypothesis eliminating)
* General inferences (Broad inferences)	Specific inferences (Narrow inferences)
* Strong inferences (Conclusive)	* Weak inferences (Suggestive)
Unique chain of inferences for each hypothesis	* Several chains of inference yield the same hypotheses
Goal Characteristics	
* Find all solutions Consistency for every solution	Find one solution * Plausibility for every solution

[a] In some cases the left and right columns may be viewed as extremes of a spectrum. Asterisks indicate those attributes which are characteristic of the segmentation domain. Both positive and negative inferences in this domain tend to be general. Negative inferences tend to be strong and positive inferences tend to be weak. Although the raw data are errorful, they are also redundant and can be reliably corrected.

appealing and the first considered. A model-driven approach sometimes seems less intuitive—perhaps because it operates predominantly by ruling out all of the wrong answers.

In choosing a problem-solving method, it is important to examine the character of both the data and the inference rules which will be used. In this section we will see some difficulties with using a data-driven approach, using the two rules previously described.

5.1. Sensitivity to missing data

The performance of any problem-solving method is affected by missing data. In the extreme case—no data at all—methods must generate an unconstrained set of solutions. For segmentation problems, there is not enough *a priori* information to afford any practical enumeration of potential structures. (With no digest data at all, the solution space would be infinite since the number of sites in a structure and the sizes of the segments would be indeterminate.) Choice of the generation scheme and inference rules determines consequences of errors in the data.

The inference rules in the data-driven generation approach are sensitive to

errors in all of the digests. If segments are missing in any complete or incomplete digest, some possible hypotheses may not be considered. The hypothesis generator for the model-driven approach is less sensitive to missing data because it requires data from only the 1-enzyme and 2-enzyme complete digests. If data are missing from the other digests, at worst a few too many candidates may be generated; in many cases a few missing segments will have no ill effect at all.

5.2. Efficient use of the data

A second difficulty with a purely data-driven approach for segmentation problems is that redundancy in the data causes unnecessary computational work. This effect increases with problem size and with the amount of the data.

The number of segments in an ideal incomplete digest[7] for circular structures is

$$N(N-1)+1$$

and for linear structures is

$$\frac{N(N+1)}{2}$$

where N is the number of segments in the corresponding complete digest. Hence, the number of segments in an incomplete digest increases quadratically with the number of segments in the corresponding complete digest.

Most of these segments (except N of them which correspond to the individual segments from the complete digest) could be used by the incomplete digest rule. However, in most cases only N of them are required to determine segment placement. The incomplete digest rule does not say how to select from this excessive data. Criteria for data selection would provide a middle ground between data-driven and model-driven approaches.

Unfortunately, it turns out that when data are missing, selection is harder. A specification which would cover all combinations of missing data would be quite complicated. *Without this additional knowledge, a purely data-driven approach to generation must consider all of the evidence in order to avoid missing a solution.* The cost of completeness is that it will do the most work when none of the data are missing.

Finally, the number of inferences that can be drawn by each application of either of the data-driven rules increases exponentially with the number of segments in the summation set. Each application of a rule to a segment may be used to propose any permutation of the segments in the summation set. Thus, the number of inferences is unwieldy for rather modest sets. Typically, the same hypotheses will be generated several times by applying the two rules to different parts of the data. In contrast, the model-driven approach considers each candidate at most once.

[7] We are considering only those linear segments corresponding to the model of digestion discussed previously. In some circumstances (e.g. "super-coiled" DNA) additional segments may appear, but these are essentially laboratory artifacts which are removed from the data before computation. Hence, they are not included in the discussion.

5.3. Generality and the use of negative inference

Two types of inference may be distinguished—positive and negative. By "positive inference" we mean using data to propose hypotheses; by "negative inference" we mean using data to rule out hypotheses. For every domain two basic questions should be answered.

> How much evidence is needed to infer that a hypothesis is true?
>
> How much evidence is needed to infer that a hypothesis is false?

Several researchers have been concerned with quantifying the strength of inferences and the certainty of hypotheses. This work seeks to quantify and use measures of certainty or belief in hypotheses according to available evidence. Duda [4] reviews some recent work in this area. Since the pruning rules used by GA1 are all very reliable, sophisticated techniques for the accounting of certainty have not been necessary.

The *n*-enzyme and incomplete digest rules are examples of positive inference because they propose structural hypotheses. Neither rule can unambiguously determine molecular structure. Like the pruning rules in Section 4.2.5, these rules are general because each application refers to a class of hypothetical structures. For example, the *n*-enzyme digest rule states that some permutation of a set of segments is a substructure of a hypothesis; it does not determine the order of the segments.

The generality of these inferences has important implications for the accumulation of certainty. Since each positive inference leaves open the possibility of many different substructures (the number depends on the size of the set), it is possible to create examples for which there is an arbitrarily large amount of positive evidence confirming a false hypothesis. Thus these inferences are weak and certainty for hypotheses does not accumulate with successive positive inferences.

In contrast, many of the negative inference rules in the segmentation domain are strong. These rules are based on a theory of segmentation which predicts what data should be observed given a hypothesis. If the data observed differs from the predictions, the hypothesis may be ruled out.

For negative inference, generality is an advantage because it permits ruling out classes of candidates. It is appropriate to ask how much negative evidence is needed to rule out a hypothesis. A single contradiction is usually not enough when the data are errorful. If errors are rare in the data, the case against a particular hypothesis becomes significantly stronger with each successive negative inference.

The observations about the strength of rules for positive and negative inference may be summarized as follows.

> Rules for positive inference have the most utility when they are highly specific; rules for negative inference have the most utility when they are general.

Since all of the rules in the segmentation domain are general, we are led to an alternative strategy to the data-driven approach. Instead of using so much data to drive positive inferences, we can use it instead to drive negative inferences.

Candidates are selected by a process of elimination; *we become certain that they are correct through our certainty that the other possible answers are wrong*. This is the model-driven strategy of using an exhaustive generator of hypotheses and effective rules for pruning its output.

5.4. Use of negative data

In the data for segmentation problems, two kinds of data may be recognized—a segment may be observed or it may not be observed. Thus, the absence of a segment in a digest may be considered as evidence. Both kinds of data may be used for both types of inference.

Reliability of data depends on what kinds of errors are common. If missing data are common, then negative data will be unreliable because missing data can be misinterpreted as negative data. If extraneous data are common, then positive data will be unreliable because extraneous data can be misinterpreted as positive data.

In segmentation problems, extraneous segments are rare while data are fairly often missing. Thus, for uncorrected digest data, positive data are more reliable than negative data. The data correction and checking rules described in Section 4.2.1 use the natural redundancy of digest data to greatly improve the reliability of both positive and negative data.

5.5. Solution requirements

The following requirements summarize some conclusions from the above considerations for segmentation problems.

1. The solution method should generate all solutions which are consistent with the data. In this case, this means requiring all solutions which are not ruled out by the data.
2. An improvement or increase in data should not degrade program performance measured in terms of correctness or time requirements.
3. The consequences of errorful data should be minimized.
4. A solution method should make efficient use of the evidence.

6. Performance Evaluation

GA1 has proven to be fast and effective for most of the segmentation problems to which it has been applied. In some cases it has found solutions overlooked by human problem solvers. Sometimes it has reported extra solutions which could be ruled out by information not given to the program. The facility for checking and correcting input data has been useful in catching typing errors and has often surprised GA1's users. As GA1 has been applied to larger problems, it has been necessary to extend the kinds of structural constraints which can be specified.

Table 3 summarizes some parameters of the performance of GA1 on three laboratory experiments. The column labeled "Canonical Structures" indicates the number of canonical structures—computed as in the sample problem. In these examples, only one candidate was generated because the pruning rules were very effective. In other problems, several candidates reach final evaluation.

The amount of time required by GA1 to do the calculations has been broken down into generation time and evaluation time. These figures show that the computation time is not linearly proportional to the number of possible hypotheses. Generation time is the time required to generate the candidates—with simultaneous pruning by the pruning rules discussed already. This will vary on different problems even if they have the same number of canonical structures because the rules are effective at different levels for different problems. Generation time increases when the tolerable number of contradictions is increased and when the measuring tolerance is increased. Evaluation time is mostly the time taken to predict the laboratory results but also includes the ranking of candidates. Variations in time of twenty to thirty percent, have been noticed and are due to the time-shared paging environment in which GA1 is run.[8]

TABLE 3. Performance of GA1 on three problems[a]

Case	Canonical structures	Time (seconds) Generation	Evaluation
1	5 400	0.94	0.80
2	2 494 800	3.14	0.71
3	133 660 800	25.6	1.2

[a] Time requirements depend on the number of possible structures, the available lab results, the resolution of measurement, and known structural constraints.

GA1's "sites and segments" structural model is a well-established part of the theory of molecular genetics. The limitation to linear and circular topologies is due to the nature of the plasmids and viral DNA on which these segmentation experiments are performed.

The GA1 program has been tested on about thirty problems. New pruning rules and data-correction rules have been added as difficulties have been encountered in laboratory data.

As GA1 has been applied to more complex segmentation problems, it has become clear that the human problem solvers use information not present in the digest data. Thus, although GA1 can solve in seconds smaller problems which require minutes to hours for humans to solve, it could not narrow down the possibilities on some of the larger problems. The reason for this is that the geneticists have biological information which provides additional structural constraints

[8] GA1 is written in *INTERLISP* and is run on the SUMEX facility at Stanford. SUMEX has a dual KI-10 computer with 512K of memory and uses the TENEX operating system.

not known to GA1. For example, previous experiments or independent biological evidence may imply that certain segments must be adjacent or that a particular segment is at the end of a linear molecule. GA1 is being extended so that additional constraints about segment placement can be specified.

7. Summary and Comparison to DENDRAL

Segmentation problems have been used for a case study of alternative generation strategies. This section attempts to abstract and summarize the ideas which are broadly applicable to many other domains. First, however, a comparison to the conceptually similar DENDRAL program [2] will be given. The two domains and programs will be examined in order to highlight their similarities and differences. Then the main lessons from the case study will be summarized.

7.1. Comparison to DENDRAL

GA1 is similar in several ways to the DENDRAL program. Both programs infer structures from fragmentation data and generate the structures using an exhaustive generator and a Generate-and-Test paradigm. Both programs evaluate the candidates using models of the fragmentation process and a scoring function. Mixtures present special problems in both domains. A general facility for dealing with mixtures [10] was developed in DENDRAL and demonstrated for the case of mixtures of estrogens. Both programs have evolved from using constraints based solely on fragmentation data towards using general structural constraints.

Simplicity of the hypotheses in the segmentation problem domain is largely responsible for the relative simplicity of the GA1 program as compared to DENDRAL. An early version of the DENDRAL program (acyclic DENDRAL) considers structures which are strictly trees. A newer DENDRAL program, CONGEN, can handle arbitrary cyclic graphs. (This advance in the DENDRAL project was considered a major breakthrough.) GA1 considers structures which are strictly linear or circular. The generator in acyclic DENDRAL is complete and irredundant; the generators in GA1 and in CONGEN are complete, but some non-canonical candidates are discovered only after they have been completely generated. The pieces used by the DENDRAL program are all atoms, have various chemical valences, and can be put together in many ways which are chemically stable; the pieces used by GA1 are either DNA segments or recognition sites, have "valence" one or two, and must be put together so that segments and sites alternate. There is no model of stability in GA1 because all orderings of segments are stable. The difference in hypothesis structure simplifies symmetry considerations and is largely responsible for the fact that cyclic-DENDRAL (CONGEN) required several man-years to develop and GA1 required less than two man-months.

The fragmentation rules in mass spectrometry yield the same kinds of conclusions as those in segmentation problems, although there are some key differences

between the domains. The rules for fragmentation in DENDRAL vary with the chemical class but there is essentially one type of fragmentation—analogous to an incomplete digest. A mass spectrum corresponds to incomplete digest information because the fragments may overlap. Mass spectrometry has no analog to a complete digest or to a multiple-enzyme digest. Thus, there is no technique in mass spectroscopy to break a molecule at one set of "sites" and then change some parameter to break it at some other set of distinct sites. In contrast, GA1's fragmentation rules depend on enzymes with different recognition sites to yield independent decompositions of the molecule. (When DENDRAL combines information from several spectroscopic techniques, these do not each give a complete decomposition of the molecule into fragments.) These independent decompositions make it relatively easy to determine molecular weight—a quite complicated problem in the mass spectrometry domain [3]. The digests also provide redundancy for a wide range of data correction capabilities. This makes the use of negative data reliable for candidate elimination in segmentation problems.

To summarize, GA1 uses the same general framework that DENDRAL uses—Generate-and-Test. The simplicity of the symmetry considerations in GA1 was a large factor in the overall simplicity of the program. Finally, the redundancy of data in segmentation problems makes data correction possible and permits the reliable use of negative data.

7.2. Lessons from the case study

The choice between model-driven and data-driven approaches depends both on the problem solving goals and the logic of the domain.

We have considered the choice between a data-driven and model-driven approach for segmentation problems. One factor in the selection is that the goal is to find all of the solutions that are reasonably consistent with the data. A second factor is that the rules of inference in this domain are general (i.e. non-specific); this has important implications on how the inferences should be applied in order to accumulate certainty. A third factor is that reliable rules for negative inference were available. If the goal were to find one solution—given error-free data and highly specific rules of positive inference—a data-driven approach would probably have an advantage of speed.

It was also particularly convenient that

(1) the constraints for the hypothesis generator could be easily inferred from the digest data and

(2) that the representation of the hypotheses could be uniform.

For model-driven generation an increase in data leads to an improvement in performance.

That an increase in data should not degrade performance is always appealing but less often achieved. In the model-driven approach to segmentation problems the rules of inference are organized so that increased data yields increased pruning—thus satisfying this criterion. In the data driven approach to this problem, additional

data may yield (after additional computational work) only the same hypotheses again. The addition of redundant data does not degrade performance in a model-driven approach because it solves the problem by indexing through the solution space instead of through the data.

Errorful data appears even in some simple domains.

This is perhaps obvious to anyone who has built a program which uses real data, but it bears repeating. The rules used by GA1 are an illustration of the possibilities for categorizing errors. The simple topological model and the inherent redundancy of the data create considerable opportunity for data correction and detection.

This study has discussed some general techniques but GA1 is not a general program.

Some common techniques and considerations used both in GA1 and DENDRAL have general applicability. For example, the techniques that were useful in conjunction with model-based approach include:

(1) the use of canonical forms for efficiency in generation,

(2) pruning rules to eliminate classes of candidates and

(3) rules for the detection and correction of errorful data.

Given the striking similarities between the programs and problem solving goals, it is reasonable to ask whether a general program could be written which would cover both (and possibly other) domains. What seems to be lacking is a program to create the irredundant generator of hypotheses—given a description of the nodes and arcs of the graphs and symmetry considerations. The fact that the development of the theory behind the generator in DENDRAL required several years suggests that this would be a difficult task.

ACKNOWLEDGMENTS

Special thanks to Joshua Lederberg for his thought-provoking suggestions and encouragement on this problem, to Jerry Feitelson for suggesting the problem and to Larry Kedes for his patience. Thanks also to Bruce Buchanan, Randy Davis, Ed Feigenbaum, Peter Friedland, and Nancy Martin for many helpful discussions on these and other matters.

Data was provided by the laboratories of Professor Lederberg and Professor Kedes. This research was done as part of the MOLGEN[9] project. MOLGEN is supported by NSF grants MCS76-11935 and MCS76-11649. Computing support is provided by the SUMEX computing facility, sponsored by the NIH Biotechnology Resources Program under NIH grant RR-785.

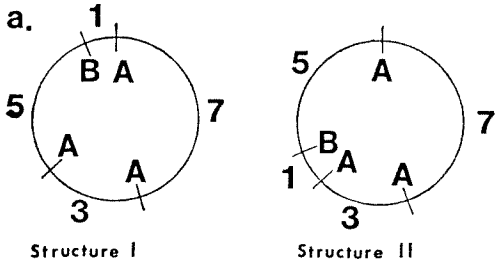
Appendix I. Proposing New Laboratory Techniques

Sometimes the explication of the logic of a domain for a computer program has additional benefits beyond the creation of the application program. This section describes a new laboratory technique involving complete digestion by one enzyme

[9] MOLGEN [7] is a joint project between the Heuristic Programming Project and Genetics department of Stanford University, and the Computer Science Department of the University of New Mexico.

and incomplete digestion by another. In some cases, this new laboratory technique can resolve structural ambiguities without the necessity of using an additional enzyme. The idea for this technique came out at a discussion about why GA1 could not discriminate between two structures.

Fig. 6 illustrates two simple DNA structures for which conventional digestion techniques yield the same segments. However, an unconventional approach using complete digestion by enzyme B followed by incomplete digestion by enzyme A yields discriminatory results as shown. The first structure may be converted into the second structure by rotating the 1 and 5 segments around the recognition site for enzyme B. Conventional digest results do not change when this rotation is performed. The unconventional approach avoids this rotational ambiguity by cutting the B site before the incomplete digest.



b. Complete Digests / Incomplete Digests

	Complete Digests	Incomplete Digests
A:	7 6 3	A: 16 13 10 9 7 6 3
B:	16	B: 16
A&B:	7 5 3 1	

c.

Structure	Unconventional Digests
I.	16 15 11 10 8 7 5 3 1
II.	16 15 12 11 10 7 5 4 3 1

FIG. 6. (a) Two simple DNA structures. The labels A and B stand for enzyme recognition sites. Structure II is like structure I except that the segments of length 1 and 5 are reversed.

(b) Conventional digests results. These results are the same for both structures.

(c) Unconventional digest results using complete digestion by enzyme B followed by incomplete digestion by enzyme A. This technique can distinguish between structure I and structure II without introducing an additional enzyme.

The technique could have been suggested by the process of filling out all combinations of digest parameters as shown in Table 4. This table shows that the new technique may be viewed as the combination of using an incomplete digest with more than one enzyme.

There are many generalizations of the above technique involving multiple partial digests with multiple enzymes. (The resolution of current gels for separating and measuring segments is a limiting factor here.) Another variety of segmentation evidence requiring more sophisticated equipment is two-dimensional analysis. This involves the digestion of a sample by one enzyme and separation of the segments along one axis, followed by the application of a second enzyme and separation along a perpendicular axis. Lederberg has speculated on the utility of combining complete and incomplete digestion with the use of 2-dimensional analysis. For example, an incomplete digestion by one enzyme followed by complete digestion by the same enzyme along the second axis yields a type of experiment data analogous to metastable data for mass spectrometry.

TABLE 4. Proposed new digestion techniques

	1-dimensional Enzyme Analysis[a]	
	Complete Digest	Incomplete Digest
1-enzyme	conventional	conventional
n-enzyme	conventional	New

	2-dimensional Enzyme Analysis[b]	
	Complete Digest	Incomplete Digest
1-enzyme	conventional	New (analogous to metastables in mass spectroscopy)
n-enzyme	conventional	New

[a] This table illustrates the combinations of using complete or incomplete digestion with one or more enzymes in the 1-dimensional analysis techniques discussed in this paper. The unconventional digest is the combination of incomplete digestion with use of more than one enzyme.

[b] The same combination of techniques can be used in 2-dimensional enzyme analysis. This leads to two new techniques for digestion. One of these techniques is analogous to "metastable scanning" in mass spectroscopy.

One of the long term goals of the MOLGEN project is to explore techniques for invention and discovery. A working paper [6] discusses some examples of this in the context of a recent genetics experiment.

Appendix II. Rules for Segmentation Problems

This appendix lists the rules which make up the problem-solving knowledge in the GA1 program. The term "rules" is used because of the production rule format

in which these entries may be described. Each entry describes the conditions under which an inference may be made. Several of these rules have already been described in the text. The genetics justification of these rules will not be given here. All of the rules except for the data-driven (D) rules are implemented in the GA1 program as *LISP* expressions.

Rules from the Data-Driven Approach

Rule D1. If a segment from the incomplete digest for enzyme E1 equals the sum of a set M of segments from the complete digest by the same enzyme, then the segments in M are probably contiguous in the structure and are separated by sites for enzyme E1.

Rule D2. If a segment from the 1-enzyme digest for enzyme E1 is equal to a sum of a set M of segments from the 2-enzyme digest by enzymes E1 and E2, then the segments in M are probably contiguous in the structure and are separated by recognition sites for enzyme E2.

Rule D3. If a segment from the 2-enzyme digest for enzymes E1 and E2 is equal to a sum of a set M of segments from the 3-enzyme digest by enzymes E1, E2, and E3, then the segments in M are probably contiguous in the structure and are separated by recognition sites for enzyme E3.

Rules for Segment Uniqueness

Rule U1. If no segment appears more than once in the list of potential segments at the beginning of the generation process, then every segment is a unique segment.

Rule U2. Every segment larger than half the molecular weight is unique.

Rule U3. If a segment appears more than once in any 2-enzyme or 3-enzyme complete digest, then that segment is not unique. (It may be repeated in the structure.)

Rule U4. If a segment which is repeated in the list of potential segments appears in the 1-enzyme digest for enzyme A and only in *n*-enzyme digests which include A (but is not repeated more than once in any of these), then the segment is unique.

Data Checking and Correcting Rules

Rule C1. If a segment which appears in the complete digest for an enzyme fails to appear in the incomplete digest for that enzyme, it may be added to the list of segments for the incomplete digest.

Rule C2. Every segment which appears in a complete digest involving three or more enzymes must appear in some complete digest involving only two of the enzymes.

Rule C3. All of the sums of the segments for the complete digests by one or more enzymes should be equal. Let MW be the molecular weight (sum of segments) predicted by the majority of the complete digests.

Rule C4. For circular molecules, the number of segments expected in every *n*-enzyme complete digest is the sum of the numbers of segments from the 1-enzyme digests; for linear molecules, the number is one less. (This rule is one form of a law for the conservation of recognition sites.)

Rule C5. If the sums of all the complete digests are equal but the number of segments in some *n*-enzyme complete digest is less than the number of segments predicted by Rule C4, then hypothesize the existence of the required number of small unobserved segments.

Rule C6. If there is a digest (termed the maverick digest) which predicts a molecular weight of MW′ where $MW' < MW$ and if the maverick digest contains a segment of mass equal to MW–MW′, then hypothesize that the segment is an unresolved doublet which should appear twice in the maverick digest.

Rule C7. If there is an *n*-enzyme complete digest (termed the maverick digest) which predicts a molecular weight of MW′ where MW′ < MW and if the number of segments in the maverick digest is less than the number predicted by Rule C4, then hypothesize an unobserved segment equal to MW–MW′.

Rule C8. If circular structures are being generated and there is a segment of size X in an incomplete digest (and X is not known to be an extraneous segment), then there should also be a segment whose size is the difference MW–X.

Rule C9. Rules for removing extraneous segments should be used before those for inserting missing segments.

Rule C10. If there is a maverick digest which predicts a molecular weight of MW′ where MW′ > MW and if there is a segment in the maverick digest whose mass is equal to MW′–MW, then hypothesize that the segment is extraneous.

Rule C11. If there is a maverick digest which predicts a molecular weight of MW′ where MW′ > MW and if there are segments in the maverick digest equal to the sum of other segments in the digest, then hypothesize that the digestion is incomplete. (If removal of some of these segments will correct the molecular weight discrepancy while preserving the expected number of segments, then remove the segments.)

Rule C12. If a segment appears in the complete digest of N enzymes which is larger than any segment which appears in the digests of any N–1 of the enzymes, then hypothesize that the segment is extraneous.

Rule C13. If a segment of size X appears in the complete digest for enzyme E1 and if there is a 2-enzyme complete digest for enzymes E1 and E2 in which there is no set of segments whose sum is X, then report that either the tolerance is too small or the digests are inconsistent.

Rule C14. If a segment of size X appears in the incomplete digest for enzyme E1 and if there is no set of segments in the complete digest for enzyme E1 whose sum is equal to X, then either the tolerance is too small or X is an extraneous segment.

Rule C15. If linear structures are being generated, then every 2-enzyme complete digest must contain two segments which also appear in one or the other of the corresponding 1-enzyme complete digests. (These are the end segments.)

Rule C16. No incomplete digest should contain any segment which is smaller than all of the segments from the corresponding 1-enzyme complete digest.

Canonical Form Rules

Rule F1. If a linear structure is being generated, the largest segment in the initial list should not be used as the first segment in the structure.

Rule F2. If linear structures are being generated and all of the remaining segments to be placed are less than the first segment and the mass of the structure is less than the molecular weight (so that at least one of them will be placed in the structure), then this branch of the generation process can be pruned.

Rule F3. If circular structures are being generated, only the smallest segment in the list of initial segments should be used for the first segment.

Rule F4. If circular structures are being generated and the second segment is about to be placed and there are several segments to be placed and the segment is the largest of the remaining segments, then this branch of the generation can be pruned.

Rule F5. If circular structures are being generated and a segment equal to the first segment is about to be placed and the total mass is less than the molecular weight (so that at least one more segment will be placed) and all remaining segments are less than the second segment of the structure, then this branch of the generation may be pruned.

Rule F6. If circular structures are being generated and a segment equal to the first segment is about to be placed and the previous segment is less than the second segment, then this branch of the generation may be pruned.

Pruning Rules

Rule P1. If linear structures are being generated, then only segments which appear in some 1-enzyme complete digest should be used as the first segment of the structure.

Rule P2. If linear structures are being generated and the user has indicated a set of end segments, then the minimum segment in this set should be used as the first segment in the structure.

Definition P3. Allowable sites for segments. Recognition sites are allowable for terminating a segment only if the segment appears in the 2-enzyme complete digests for the corresponding enzymes. (If there is only one enzyme in the experiment, then only its sites are allowable.)

Rule P4. If a segment is about to be placed and the previous site is not one of the allowable sites for this segment, then this branch of the generation may be pruned.

Rule P5. If a site is about to be placed and it is not an allowable site for the previous segment, then this branch of the generation may be pruned.

Definition P6. Required termination sites for segments. If only one enzyme was used in the experiment, then the site for that enzyme is required for every segment. If two enzymes were used, then for each segment which does not appear in a 1-enzyme digest, both enzyme sites are required. If three or more enzymes were used, then for each segment which appears in exactly one 2-enzyme complete digest, the sites for the enzymes involved in that digest are both required.

Rule P7. If a segment having required sites is about to be placed and the previous site is not one of them, then this branch of the generation may be pruned.

Rule P8. If a site is about to be placed and the previous segment has required sites and this site is not one of them, then this branch of the generation may be pruned.

Rule P9. If a site is about to be placed and the previous segment has two required sites and the previous site is one of the two required sites but this site is not the other one, then this branch of the generation may be pruned.

Rule P10. If a segment is about to be placed which would increase the mass of the current structure to be greater than the expected molecular weight and there are more sites to be placed, then this branch of the generation may be pruned.

Rule P11. If circular structures are being generated and the first segment is unique and appears in the 1-enzyme complete digest for enzyme E1, then a recognition site for E1 can be placed in front of the first segment.

Rule P12. If linear structures are being generated and a segment is about to be placed which is on the list of user-supplied end-segments and there are more sites to be placed, then this branch of the generation may be pruned.

Definition P13. Allowable inter-site segments. For recognition sites E1 and E2, a segment is said to be allowable between E1 and E2 when it appears in the appropriate digests. Specifically, if E1 is distinct from E2, the segment must appear in the 2-enzyme complete digest involving E1 and E2. Otherwise it must appear in the 1-enzyme complete digest for E1.

Rule P14. If a site E1 is about to be placed and there is another site E2 preceding it in the structure (and there is no site equal to E1 or E2 between them) and the sum of the intermediate segments is not an allowable segment for E1 and E2, then this branch of the generation may be pruned.

Definition P15. An incomplete digest is said to be "ideal" if it has the right number of segments and each of the segments equals the sum of a set of segments from the corresponding complete digest. (The number of segments expected in an ideal incomplete digest is given in Section 5.2.)

Rule P16. If the incomplete digest for enzyme E1 is ideal and there is a sum of segments between E1 sites which does not appear in the ideal digest, then this branch of the generation may be pruned.

REFERENCES

1. Beers, R. F. and Bassett, E. G. (Eds.), *Recombinant Molecules: Impact on Science and Society*, Raven Press, New York, New York (1977).
2. Buchanan, B. G. and Feigenbaum E. A., DENDRAL and Meta-DENDRAL: Their applications dimension, *Artificial Intelligence* (this issue) (1978).
3. Dromey, R. G., Buchanan, B. G., Lederberg, J. and Djerassi, C., Applications of artificial intelligence for chemical inference, XIV. A general method for predicting molecular ions in mass spectra, *Journal of Organic Chemistry* **40** (1975), 770.
4. Duda, R. O., Hart, P. E. and Nilsson, N. J., Subjective Bayesian methods for rule-based inference systems, *Proceedings of the 1976 National Computer Conference (AFIPS Conference Proceedings)* **45** (1976) 1075–1082.
5. Feigenbaum, E. A., The art of artificial intelligence: I. Themes and case studies of knowledge engineering, *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (1977) 1014–1029.
6. Feitelson, J. and Stefik, M., A case study of the reasoning in a genetics experiment, Heuristic Programming Project Working Paper 77–18, Computer Science Department, Stanford University (April 1977).
7. Martin, N., Friedland, P., King, J. and Stefik, M., Knowledge base management for experiment planning in molecular genetics, *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (1977) 882–887.
8. Mitchell, T., Version spaces: A candidate elimination approach to rule learning, *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (1977) 305–310.
9. Perkins, W. A., Model-based vision system for scenes containing multiple parts, *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (1977) 678–684.
10. Smith, D. H., Buchanan, B. G., Engelmore, R. S., Adlercreutz, H. and Djerassi, C., Applications of artificial intelligence. IX. Analysis of mixtures without prior separation as illustrated for estrogens, *Journal of the American Chemical Society* **95** (1973) 6078.