CHAPTER 39

Applications of Combinatorics to Molecular Biology

Michael S. WATERMAN

Departments of Mathematics and Molecular Biology, University of Southern California, Los Angeles, CA 90089-1113, USA

Contents

This work was supported by grants from the National Institutes of Health, the National Science Foundation and the System Development Foundation.

HANDBOOK OF COMBINATORICS
Edited by R. Graham, M. Grötschel and L. Lovász

1. Introduction

The biological sciences have undergone a revolution in the last dozen years. Almost every edition of a major newspaper reports some new discovery in biology, often with medical and/or financial implications. Biologists now have the ability to rapidly read and manipulate DNA, the basic material of life that makes up chromosomes and is the carrier of genetic information. The reading of DNA is called sequencing, since the scientists are determining the linear sequence of bases along the DNA molecule. The bases or alphabet of DNA is adenine (A), guanine (G), cytosine (C), and thymine (T). These bases, joined to a sugar–phosphate backbone, are linked together in a chain to form DNA. Frederick Sanger and Walter Gilbert independently developed procedures for the rapid sequencing of long segments of DNA molecules. They received the Nobel Prize in 1980 for their discoveries. Sanger, incidentally, was earlier the first to determine the amino acid sequence of a protein, insulin.

Rapid DNA sequencing has caused an information explosion. It was only in 1953 that a complementary double-helical structure was postulated for DNA. By 1975 only a few hundred bases had been sequenced. In Spring 1994 DNA sequences are collected in international databases and sequences totaling about 200 million bases are known. These sequences come from various locations in the genomes of a wide variety of organisms. (A genome holds all the genetic information of an organism.) The sequences vary greatly in size. A long continuous sequence that has been determined to date is that of human cytomegalovirus which is 229 354 bases long.

Early in this century, Fisher, Haldane and Wright did fundamental work in proving that the Mendelian model of genetics, with discrete alleles, is rich enough to generate the seemingly continuous range of phenotypes observed in nature. This might seem almost trivial in light of today's emphasis on discrete mathematics, but it was by no means obvious at that time. Their mathematical work in population biology led experimental biology. Today mathematical scientists lag far behind the experimental biologists as they read the basic material of the gene and directly test hypotheses about the nature of life. There has developed a small field of mathematical and computer sciences to assist the molecular biologist in his endeavor. Most of this mathematical development is about discrete structures. See Waterman (1989).

Increasing attention is being given to the mathematical and computational aspects of molecular biology because of the human genome project. This project can be viewed as directed toward sequencing all the DNA of humans and other organisms. While 200 million bases of DNA have been sequenced in pieces that average about 1000 bases long, the genome of even the bacterium *E. coli* is about 5 million bases. Man has a genome of 3 billion bases. Presently, the efforts center on improving the mapping and sequencing technology so that such sequencing projects can be more easily accomplished. Even so, using today's technology, genomes of the size of those of *E. coli* will be sequenced within the next few years. A number of analytical problems are concerning people who are involved in these studies. First of all, the puzzles of assembling map and sequence information from the experimental results are large, combinatorial problems. In addition, the vast quantity of data will severely tax our current methods for finding the relationships between the sequences that are determined

In this chapter some combinatorial aspects of molecular biology will be explored. Section 2 discusses sequence alignments where certain sequence relationships are studied, both by enumeration and algorithms. The next section gives some results on enumeration and algorithms for secondary structures. The final section treats restriction maps of DNA and their relationship with the human genome project. The emphasis is on the description and straightforward solution of some of the related problems. Recently this general area has become increasingly active.

2. Sequence alignments

Evolution is a key concept in biology. To understand living organisms, biologists study the relationships between the organisms and their environments. Important inventions, such as the eye, are maintained and improved on throughout history. When these concerns of understanding the how and why of biology are brought to a molecular level, the evolutionary mode of thinking is extremely important. Certain machinery such as that involved in DNA to protein translation (the genetic code) is present in all organisms and works everywhere in essentially the same way. These mechanisms are so basic to life and so much additional biological activity depends on them that they cannot be modified except in very minor ways.

Other more recent 'inventions' at the molecular level allow us to understand the difference between life forms in terms of their history. For example, organisms with a nucleus (such as humans) are classified as eukaryotes while those without a nucleus (such as *E. coli*) are classified as prokaryotes. Finer and finer distinctions can be made, and classifying organisms goes hand in hand with understanding how they function.

Something of the same approach is taken by biologists in performing DNA sequence analysis. Given a sequence $\boldsymbol{x}$, what known sequences are related to it and what are the relationships? Before this question can be explored we need to understand what evolutionary events can take place during sequence evolution. The simplest event is substitution, where one nucleotide is replaced by another, as when A is replaced by C for example. Nucleotides can be inserted into or deleted from a sequence, either one nucleotide at a time or in blocks. Insertions and deletions greatly complicate the analysis. Inversions and duplications of a block of sequence make things even more difficult.

It is common in molecular biology to try to discover the function of a DNA or protein sequence by relating it to other sequences. Frequently this means a biologist will compare a sequence with a large number of previously analyzed sequences; the comparison is done using a computer using algorithms as described below. These comparisons are usually done with sequences taken two at a time. Often there are families of related sequences where any pair might have a fairly weak relationship. Therefore there is a good deal of interest in comparison of more than two sequences, often in comparison of several hundred sequences.

In most sequence analysis the sequence transformations are restricted to substitutions, insertions or deletions. The biologist represents his findings in an alignment of one sequence written over another, and the sequence transformations can be read from the alignment. For example,

$$\begin{array}{l} ATTA{-}CGG \\ {-}CGACC{-}G \end{array}$$

is an alignment of $ATTACGG$ with $CGACCG$. From the point of view of taking the top sequence as the "original" sequence, this alignment shows the events in an evolution of $\boldsymbol{x}$ to $\boldsymbol{y}$. There has been the deletion of an A and a G, the substitution of C and G for the two T's, and the insertion of a C. There is no history recorded in an alignment, since there is no information about the timing of the events relative to one another nor is it known which sequence "came first". In fact, some other sequence is likely to have been the ancestor of both sequences. In the next section we consider some combinatorics motivated by considering the history of the events, then we discuss sequence alignment combinatorics and algorithms.

2.1. Shuffles and alignments

Let $\boldsymbol{x} = x_1x_2\cdots x_n$ and $\boldsymbol{y} = y_1y_2\cdots y_m$ be two sequences. The problem under consideration here is to count the histories for a special type of evolution: delete all the letters of $\boldsymbol{x}$ and insert all the letters of $\boldsymbol{y}$. The deletion/insertion events take place one letter at a time, the events can be performed in any order and it is possible to track each nucleotide. Thus for simplicity it is assumed that all $n+m$ letters are distinct. The results described in this section are from Greene (1988) where material of independent combinatorial interest also appears. While this is a very special case of molecular evolution, the possible histories between two sequences are of much biological interest. Greene's work is the first mathematical study of this complex problem.

Define an order by $\boldsymbol{s} \leqslant \boldsymbol{t}$ if $\boldsymbol{s}$ is a subsequence of $\boldsymbol{t}$. Let $\{\boldsymbol{s}\}$ denote the set of letters in the sequence $\boldsymbol{s}$, and $\boldsymbol{s}|\boldsymbol{t}$ denote the sequence $\boldsymbol{s}$ restricted to the set $\{\boldsymbol{t}\}$. A sequence $\boldsymbol{s}$ is on a path between $\boldsymbol{x}$ and $\boldsymbol{y}$ if $\{\boldsymbol{s}\} \subset \{\boldsymbol{x}\} \cup \{\boldsymbol{y}\}$ with $\boldsymbol{s}|\boldsymbol{x} \leqslant \boldsymbol{x}$ and $\boldsymbol{s}|\boldsymbol{y} \leqslant \boldsymbol{y}$. This set of sequences is denoted $W(\boldsymbol{x},\boldsymbol{y})$ and was noted by Greene to be shuffles of subsequences of $\boldsymbol{x}$ and $\boldsymbol{y}$. If we maintain the idea of going "from" $\boldsymbol{x}$ "to" $\boldsymbol{y}$, there is a natural partial order on $W(\boldsymbol{x},\boldsymbol{y})$:

$$\boldsymbol{s} \leqslant_* \boldsymbol{t} \text{ if } \begin{cases} \boldsymbol{s}|\boldsymbol{x} \geqslant \boldsymbol{t}|\boldsymbol{x}, \\ \boldsymbol{s}|\boldsymbol{y} \leqslant \boldsymbol{t}|\boldsymbol{y}, \\ \boldsymbol{s}|\boldsymbol{t} = \boldsymbol{t}|\boldsymbol{s}. \end{cases}$$

All sequences of the same length have the same order structure so we set $W(\boldsymbol{x},\boldsymbol{y}) = W(n,m)$. For any n and m, $W(n,m)$ is a lattice.

There are some natural combinatorial questions about $W(n,m)$ such as determining $\Omega(n,m)$, the number of elements. Greene answers virtually all of these questions. We set $C_{n,m}$ equal to the number of maximal chains in $W(n,m)$ and define

$$\Phi_{n,m}(x) = \sum_{j\geqslant 0} \binom{m}{j}\binom{n}{j} x^j.$$

This last function is closely related to the Jacobi polynomials and both of the quantities of interest can be expressed in terms of it.

$$\Omega_{n,m} = 2^{n+m}\Phi_{n,m}(1/4),$$
$$C_{n,m} = (n+m)!\,\Phi_{n,m}(1/2),$$

Many interesting cases remain to be studied. Simply changing one sequence to another by deleting all letters of one sequence and inserting all letters of another is not realistic biology. The extension of Greene's work to allow matching and mismatching letters remains to be made; it is likely to be extremely difficult.

2.2. Sequence alignment

In this section we will consider alignment of $\boldsymbol{x} = x_1x_2\cdots x_n$ and $\boldsymbol{y} = y_1y_2\cdots y_n$. The sequences are the same length to avoid non-essential complications of the results. Both algorithms and combinatorics for alignment are easy if no insertions and deletions are allowed. There are simply $2n+1$ ways to align the sequences, one over the other. Each of these alignments can be evaluated for quality of matching by direct examination of the overlapping portions. Therefore the best alignments can be found in $O(n^2)$ time. While this chapter is not intended to be a survey of algorithms for alignments, this area is discussed as it is of great importance in biology. Also, it motivates some useful combinatorics. Insertions and deletions can be included in sequence alignments and best alignments can still be located in $O(n^2)$ time. Reviews of the field have appeared in Kruskal and Sankoff (1983) and in Waterman (1984, 1989).

An alignment can be viewed as a way to extend the sequences to be of the same length L, equal to the overall length of the alignment. The alignment shown above

$$\begin{array}{l} ATTA{-}CGG \\ {-}CGACC{-}G \end{array}$$

has length $L = 8$. Note that the alphabet for the extended sequences has been increased by the symbol "–".

We now turn to asymptotics for the number of alignments of two sequences of length n. The first results for this problem related the number of alignments to the Stanton–Cowan numbers (Laquer 1981). One way to count alignments is to identify aligned pairs $\binom{x_i}{y_j}$ and simply to choose subsets of $\boldsymbol{x}$ and $\boldsymbol{y}$ to align. This gives

$$\sum_{k\geqslant 0}\binom{n}{k}\binom{m}{k} = \binom{n+m}{n}$$

alignments if x has n letters and y has m letters. Recent work has generalized these results. Biologists find an alignment more convincing when the matched segments, that is segments without insertions or deletions, occur in larger blocks. Let $g(b,n)$ be the number of alignments where the matched sections are of length at least b. The following appears in Griggs et al. (1986):

For $b \geqslant 1$ define

$$h(x) = (1-x)^2 - 4x(x^b - x + 1)^2$$

and let $\rho = \min\{x\colon h(x) = 0\}$. Then

$$g(b,n) \sim (\gamma_b n^{-1/2})\rho^{-n} \quad \text{as } n \to \infty,$$

where $\gamma_b = (\rho^b - \rho + 1)(-\pi\rho h'(\rho))^{-1/2}$. The proof uses generating functions for $g(b,n)$. We remark that the result of Laquer (1981) is given by the above result with $b = 1$.

Next are some results on $f(k,n)$, the number of alignments of k sequences of length n (Griggs et al. 1990). Using combinatorial argument to give the exponential growth rate:

For fixed $k \geqslant 2$,

$$\lim_{n\to\infty} \ln(f(k,n))/n = \ln(c_k),$$

where $c_k = (2^{1/k} - 1)^{-k}$. It is also possible to show that the asymptotic behavior of c_k is equivalent to that of $2^{-1/2}(\ln 2)^{-k}k^k$.

Employing a saddle point method gives more precise asymptotics for $f(k,n)$. For fixed $k \geqslant 2$ let $r = (2^{1/k} - 1)^k$. Then

$$f(k,n) = [r^{-n}n^{-(k-1)/2}]\left[(r^k\pi^{(k-1)/2}k^{1/2})^{-1}2^{(k^2-1)/2k} + \mathrm{O}(n^{-1/2})\right].$$

From the asymptotics given here it is clear that it is not possible to just look at all possible sequence alignments and pick the preferred ones. It is necessary to define an objective function for "good" alignments. Suppose a function $s(a,b)$ is given to score the alignment of a and b from the sequence alphabet, and that the problem is to find the highest scoring alignments. This score is given by

$$S(\boldsymbol{x},\boldsymbol{y}) = \max_{\text{all alignments}} \sum_{1\leqslant j\leqslant L} s(x_j^*, y_j^*),$$

where x_j^* and y_j^* are the jth members of the extended sequences.

A simple dynamic programming method can be used to find the maximum scoring alignment.

Let $\boldsymbol{x} = x_1x_2\cdots x_n$ and $\boldsymbol{y} = y_1y_2\cdots y_n$. Set $S_{0,j} = \sum_{1\leqslant k\leqslant j} s(-,y_k)$, $S_{0,0} = 0$, $S_{i,0} = \sum_{1\leqslant k\leqslant i} s(x_k,-)$, and $S_{i,j} = S(x_1x_2\cdots x_i, y_1y_2\cdots y_j)$. Then $S(\boldsymbol{x},\boldsymbol{y}) = S_{n,n}$ and

$$S_{i,j} = \max\begin{cases} S_{i-1,j} + s(x_i,-), \\ S_{i-1,j-1} + s(x_i,y_j), \\ S_{i,j-1} + s(-,y_j). \end{cases}$$

The above algorithm aligns two sequences in $O(n^2)$ time and space. Letters are inserted or deleted in blocks in biology. For general weighting of these "gaps" the dynamic programming algorithm has time $O(n^3)$ (Waterman 1986), while linear weighting retains time $O(n^2)$. It can be argued that the weighting should be concave where the comparisons can be made in almost the same time. See Waterman (1989), Miller and Myers (1988) and Galil and Giancarlo (1989).

For the case of k sequences of length n the simple algorithm generalizes to require $O(2^k n^k)$ time and space. This is computationally impractical and several different approaches have been taken to solve this important problem; see Waterman (1986) and Waterman and Jones (1990). Some recent approaches to this important problem are now described.

Carrillo and Lipman (1988) consider the generalization of dynamic programming alignments to k sequences. They observe that the score of the projection of a multiple alignment onto two of the sequences cannot be more than the score of those two sequences aligned by themselves. They exploit this observation to greatly reduce the time and storage of multiple sequence alignment. As many as 9 or 10 sequences might be aligned by their technique.

Another approach to k-sequence alignment is to build up the multiple alignment from two sequence alignments. It is obviously possible to begin with the best-aligned sequence pairs and obtain an unsatisfactory result in the end, but some groups have made useful algorithms based on this approach. In Waterman and Perlwitz (1984) some connections with geometry are explored. Taylor (1987) and Vingron and Argos (1989) have excellent programs along these general lines.

Finally in Waterman (1986) and Waterman and Jones (1990) a different approach is taken. The algorithm matches short words of set length and degree of mismatch. The words can be matched within a fixed amount of position offset and total score is maximized where a score is given to each matching word.

3. Secondary structure

When RNA is transcribed from the DNA template, it is single-stranded. That is, RNA does not possess a matching or self-complementary strand to pair with it. The single-stranded molecule can fold back on itself and when regions of the molecule are complementary they can become double-stranded or helical. The pairing rules for the sequences are analogous to those for DNA except that T becomes U (uracil) in the RNA alphabet: A pairs with U and G pairs with C. In addition, frequently G is thought to pair with U. Biologists call the two-dimensional self-pairing *secondary structure*. Without reference to an actual RNA sequence it is an interesting problem to enumerate the distinct secondary structures that are possible under various restrictions suggested by biology. The number of structures for a sequence of length n satisfies a recursion related to the Catalan numbers. There is even a vector recursion that is "Catalan-like".

Next, a definition of secondary structure is given. Label n points on the x-axis: $1, 2, \ldots, n$. The points correspond to the nucleotide sequence of the RNA. Choose a subset of $2j$ points, $0 \leqslant 2j < n$. The $2j$ points are arranged into j disjoint pairs and the pairs are connected by arcs, subject to the following conditions:

1. Adjacent points are never connected by an arc.
2. Any two points connected by an arc must be separated by at least m points.
3. Arcs cannot intersect.

Condition 2 comes from restrictions on the bending of the sugar–phosphate backbone. In RNA $m = 3$ or 4 is realistic. Condition 3 comes from eliminating structures with "knotted" loops. There are a few examples in biology where condition 3 is violated and no combinatorics has yet been done for those cases.

In the next section, computer prediction of secondary structures for RNA sequences is briefly discussed. As was the case for sequence alignment, the associated dynamic programming algorithms are closely related to enumeration of the configurations.

3.1. Prediction of secondary structure

Several attempts were made on the secondary structure "problem" before dynamic programming was first proposed. The basic problem is to find the minimum free-energy structure where negative free energy is assigned to the base pairs and positive energy is assigned to end loops, unpaired bases in helical regions, and so on. The energy rules are not too well understood. The subject is reviewed in Zuker and Sankoff (1984) and here a very simple version of the problem is solved: find the secondary structures that have the maximum number of base pairs.

Theorem 3.1. *Let $x = x_1x_2 \cdots x_n$ be a sequence over $\{A, C, G, U\}$, $1 \leqslant m$, and $p : \{A, C, G, U\} \times \{A, C, G, U\} \to \{0, 1\}$. Define $F(i, j)$ = maximum number of base pairs of all secondary structures over $x_i \cdots x_j$, where a pair can be formed if and only if $p(\cdot, \cdot) = 1$. Set $F(i, j) = 0$ whenever $j \leqslant m + i$. Then*

$$F(i,j) = \max\{F(i,j-1), [F(i,k-1) + F(k+1,j-1) + 1]p(x_k,x_j);\\ 1 + k + m \leqslant j\}.$$

Proof. The proof of the recursion is based on the observation that either x_j is unpaired or it is paired with a base x_k. To satisfy the constraints, $m \leqslant j - k - 1$. The boundary conditions simply reflect the fact that no pairs can form unless the constraint is satisfied. □

The recursion can be performed in $O(n^2)$ time and space. Unfortunately the structures predicted by this algorithm usually do not correspond to those known to exist in nature and more complicated algorithms must be employed. A very useful algorithm has been devised, again based on dynamic programming, that takes time $O(n^3)$ and $O(n^2)$ space (Zuker and Sankoff 1984). This method employs a shortcut and until recently no polynomial-time solution was known for the general problem. In Watermand and Smith (1986) a general solution was given that takes time $O(n^4)$ and space $O(n^3)$. Since sequences of interest are often 5000 long and range up to 20000, there is a need for more work in this area. See Galil and Giancarlo (1989).

3.2. Counting secondary structures

Let $S_n(m) = S_n$ be the number of secondary structures possible for a string or sequence of length n. For this discussion the structures need only satisfy the conditions stated above – no sequence-specific pairing is considered. The results are taken from Stein and Waterman (1978) and Howell et al. (1980).

Theorem 3.2. *For* $1 \leqslant m$, $S_0 = S_1 = \cdots = S_{m-1} = 0$ *and* $S_m = 1$ *are boundary values. Then*

$$S_{m+j} = S_{m+j-1} + S_{m+j-2} + \ldots + S_{j-1} + \sum_{0 \leqslant i \leqslant m+j-2} S_i S_{m+j-2-i}$$

Proof. The proof is similar to the proof of the algorithm given for Theorem 3.1. Consider adding the base $m+j$. If base $m+j$ does not pair we have S_{m+j-1} structures. Otherwise the base pairs with a base with subscript $i+1$ from 1 to $m+j-2$ and the number of structures is the product of the possibilities from the strings $1 \cdots i$ and $i+2 \cdots m+j-1$. The boundary conditions give the recursion in the above form. □

If we set $m = 0$ and consider the above recursion,

$$S_n(0) = S_n = S_{n-1} + \sum_{0 \leqslant j \leqslant n-2} S_j S_{n-2-j}$$

where $S_n(0) = 1$. This recursion generates the Motzkin numbers (Sloan 1973) and they have an explicit solution: $S_n(0) = \sum_{j \geqslant 0} c_{j+1} \binom{n}{2j}$. This formula is a consequence of Theorem 3.3 below, where we define the Catalan numbers c_{j+1} by

$$c_{j+1} = (j+1)^{-1} \binom{2j}{j}.$$

Next define the convolved Fibonacci numbers $f_n(r,k)$ by

$$(1 - x - x^2 - \cdots - x^r)^{-k} = \sum_{n \geqslant 0} f_n(r,k) x^n.$$

Theorem 3.3. *For* $1 \leqslant m$, *set*

$$g(x) = \sum_{n \geqslant 0} S_n x^n.$$

Then

$$g(x) = \sum_{j \geqslant 0} c_{j+1} x^{(m+2)j+m} \sum_{n \geqslant 0} f_n(m+1, 2j+1) x^n.$$

This relation can be derived by squaring the generating function, using the recurrence relation with some manipulations. Next asymptotics for S_n are presented. The proof is based on the folklore theorem of Bender (1974).

Theorem 3.4. *Define* $F(r,s) = r^2s^2 - (1 - r - r^2 - \cdots - r^{m+1})s + r^m$. *Let* $r > 0, s > S_0$ *be the unique real solutions of the system* $F(r,s) = 0, F_y(r,s) = 0$. *Then*

$$S_n \sim (rF_x(r,s)/(2\pi F_{yy}(r,s)))^{1/2} n^{-3/2} r^{-n}.$$

The following special cases hold:
1. $S_n(0) \sim \sqrt{3/(4\pi)} n^{-3/2} 3^n$.
2. $S_n(1) \sim \sqrt{(15 + 7\sqrt{5})/(8\pi)} n^{-3/2}((3 + \sqrt{5})/2)^n$.
3. $S_n(2) \sim \sqrt{(1 + \sqrt{2})/\pi} n^{-3/2}(1 + \sqrt{2})^n$.

The behavior of $S_n(m)$ is governed by $r(m)^{-n}$ and $r(m)$ can only be numerically determined for $m \geqslant 3$. Still, it can be shown that $r(m)$ is monotonically increasing and $r(m) \to 1/2$ as $m \to \infty$.

Next a closer examination of $S_n(m) = S_n$ is made. Set

$$\boldsymbol{R}^n = (r_0^n, r_1^n, r_2^n, \ldots)$$

where $r_0^n = 1$ and r_i^n is the number of secondary structures with i base pairs for a sequence of length n. Also define $\boldsymbol{a} * \boldsymbol{b} = \boldsymbol{c}$ by

$$c_k = \sum_{0 \leqslant i \leqslant k} a_i b_{k-i},$$

and let $\zeta(a_0, a_1, \ldots) = (0, a_0, a_1, \ldots)$.

Theorem 3.5. *Set* $\boldsymbol{R}_0 = \boldsymbol{R}_1 = \boldsymbol{R}_2 = (1, 0, 0, \ldots)$. *Then for* $n \geqslant 2$,

$$\boldsymbol{R}_{n+1} = \boldsymbol{R}_n + \sum_{1 \leqslant j \leqslant n-1} \zeta[\boldsymbol{R}_{j-1} * \boldsymbol{R}_{n-j}].$$

Proof. There can be no pairs for $n \leqslant 2$, so the boundary conditions hold. Next, the number of structures with $i + 1$ pairs for a sequence of length $n + 1$ is derived. If base $n + 1$ is unpaired, s_{i+1}^n structures exist with $i + 1$ pairs. If instead base $n + 1$ is paired with base j, then to have $i + 1$ pairs i additional pairs are needed. If k pairs exist for bases 1 to $j - 1$, then $i - k$ pairs must exist for bases $j + 1$ to n. □

4. Maps of DNA

For many years maps of the relative location of genes on chromosomes have been constructed by a technique known as linkage analysis. Before 1980 it was required that the genes have observable mutations available as genetic markers. Since fruit flies have many visibly distinguishable mutants, the relative locations of many of the corresponding genes have been mapped. Bacteria and yeast have also been extensively mapped. In 1980 it was realized that measurable changes in the DNA itself can be used for genetic mapping (Botstein et al. 1980). The changes are in the lengths of restriction fragments and the resulting mapping techniques have been very important in localizing genes associated with major diseases.

Site-specific restriction enzymes were discovered in 1970 in bacteria (Nathans and Smith 1975). These enzymes cut double-stranded DNA at the locations of short specific patterns, usually from four to six letters in length. The restriction enzyme HhaI cuts at $GCGC$ while EcoRI cuts at $GAATTC$. Mutations at a single letter of DNA can cause the appearance or disappearance of a restriction site. It is easy to see that insertions and deletions of a segment of DNA can also cause variation in the restriction sites. The fragment lengths then become the genetic markers for linkage analysis. This is a quite active research area and there is currently some mathematical activity in devising efficient algorithms (Lander and Botstein 1986).

In linkage analysis, map distance might not relate linearly to physical distance or number of bases. Soon after the discovery of site-specific restriction enzymes biologists learned to construct another type of map known as a restriction map. In restriction maps all the enzyme sites are approximately located on the DNA. Such maps usually cover a few thousand bases of DNA but much longer stretches of DNA have been mapped (Isono et al. 1987). This section will discuss in some detail the difficulties in restriction map construction. First, restriction maps are related to interval graphs.

4.1. Maps as interval graphs

Interval graph theory began with Benzer's study of the structure of genes in bacteria (Benzer 1959). Benzer was able to obtain data on the overlap between pairs of fragments of DNA from a gene. He was successful in arranging the overlap data in a way that implied the linear nature of the gene. Soon after this, Fulkerson and Gross (Golumbic 1980) studied interval graphs and incidence matrices; that study is closely related to Benzer's analysis. Today the linear nature of the gene is well established but interval graphs also arise in connection with restriction maps (Waterman and Griggs 1986).

Representing an interval of DNA as a line segment, the biologist indicates the location of restriction sites along the line segment. Circular DNA does occur in nature but this discussion is restricted to the linear maps of two restriction enzymes. Next the two restriction enzymes are designated by A and B. Figure 1a gives an example of a two-enzyme A/B map while the single-enzyme maps appear in fig. 1b. Biologists are able to measure the lengths but not the order of the intervals between sites, so they are labeled arbitrarily. The intervals are called restriction fragments and form the nodes of our graphs.

Label the ith fragment from enzyme A (B) by A_i (B_i). Define the incidence matrices $I(A,B)$, $I(A,A/B)$, and $I(B,A/B)$ where, for example, $I(A,B)_{i,j}=1$ if $A_i \cap B_j \neq \emptyset$ and 0 otherwise. It is sometimes experimentally feasible to determine $I(A,A/B)$ and $I(B,A/B)$. It is then a simple matter to compute $I(A,B)$.

Proposition 4.1. *If I^{T} is the transpose of I, then*

$$I(A,B) = I(A,A/B)I^{\mathrm{T}}(B,A/B).$$

Proof. The result follows from the observation that the (i,j)th element of the matrix product is equal to the number of A/B intervals in both A_i and B_j. □

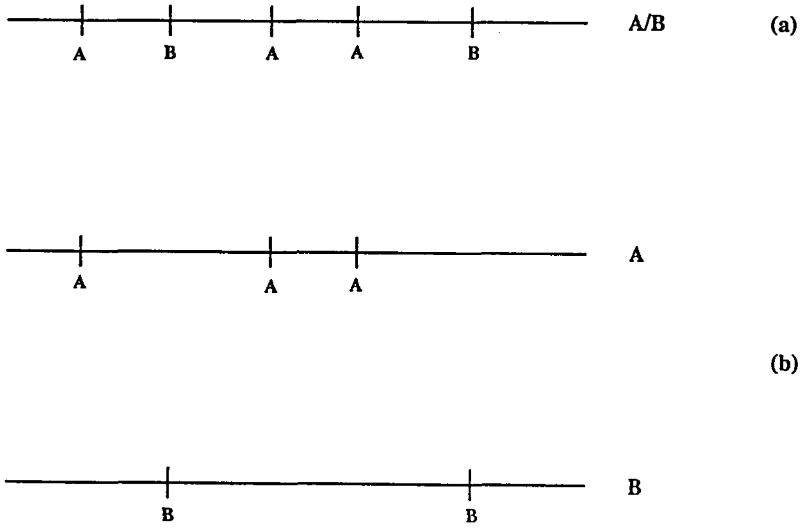

Figure 1. The two enzyme A/B map (a) and the single enzyme (A and B) maps (b).

Constructing a restriction map from $I(A,B)$ is equivalent to finding an interval representation of a bipartite graph $G(A,B)$ defined in a natural way. The vertex set $V(A,B)$ is the union of the set of A intervals with the set of B intervals; the edge set $E(A,B)$ consists of those sets $\{A_i, B_j\}$ where $A_i \cap B_j \neq \emptyset$. If we delete the endpoints of the fragments from the line segment, we obtain an open-interval representation of $G(A,B)$. Restriction maps can be characterized by known results on interval graphs (Golumbic 1980).

Theorem 4.2. *The following are equivalent:*

1. *$G(A,B)$ is a bipartite graph constructed from a restriction map.*
2. *$G(A,B)$ is a bipartite interval graph with no isolated edges.*
3. *$I(A,B)$ can be transformed by row and column permutations into a staircase form with each row or column having 1's in precisely one of the steps.*

With the identification of $G(A,B)$ as an interval graph, it is routine to adapt a general algorithm of Booth and Leuker to recognize G as an interval graph and to give its representation in linear time (Booth and Leuker 1976, Waterman and Griggs 1986). As is the case in many problems in biology, the overlap data is usually given with errors. Then the problem of finding the "interval graph" becomes much harder.

4.2. Constructing maps

It is experimentally possible to apply restriction enzymes singly or in combination, and to estimate the lengths of the resulting fragments of DNA. The problem is to construct the map of location of the enzyme sites along the DNA from this fragment-length data. The results are from Goldstein and Waterman (1987).

4.2.1. Simulated annealing

Here we consider the simplest problem of interest that involves linear DNA, two restriction enzymes, and no measurement error. We will refer to this problem as the double-digest problem or problem DDP. A restriction enzyme cuts a piece of DNA of length L at all occurrences of a short specific pattern and the lengths of the resulting fragments are recorded. In the double-digest problem we have as data the list of fragment lengths when each enzyme is used singly, say,

$$\boldsymbol{A} = \{a_i\colon 1 \leqslant i \leqslant n \text{ from the first digest}\},$$
$$\boldsymbol{B} = \{b_i\colon 1 \leqslant i \leqslant m \text{ from the second digest}\},$$

as well as a list of double-digest fragment lengths when the restriction enzymes are used in combination and the DNA is cut at all occurrences specific to both patterns, say

$$\boldsymbol{C} = \{c_i\colon 1 \leqslant i \leqslant n_{1,2}\};$$

only length information is obtained. In general $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ will be multisets; that is, there may be values of fragment lengths that occur more than once. We adopt the convention that the sets $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are ordered; that is, $a_i \leqslant a_j$ for $i \leqslant j$, and similarly for the sets $\boldsymbol{B}$ and $\boldsymbol{C}$. Of course

$$\sum_{1 \leqslant i \leqslant n} a_i = \sum_{1 \leqslant i \leqslant m} b_i = \sum_{1 \leqslant i \leqslant n_{1,2}} c_i = L,$$

since we are assuming that fragment lengths are measured in number of bases with no errors.

Given the above data, the problem is to find orderings for the sets $\boldsymbol{A}$ and $\boldsymbol{B}$ such that the double-digest implied by these orderings is, in a sense made precise below, $\boldsymbol{C}$. This is a mathematical statement of a problem originally solved by exhaustive search.

The double-digest problem can be stated more precisely as follows. For permutations $\sigma \in (12\cdots n)$, $\mu \in (12\cdots m)$, call (σ, μ) a configuration. By ordering $\boldsymbol{A}$ and $\boldsymbol{B}$ according to σ and μ, respectively, the set of locations of cut sites is obtained:

$$S = \left\{ s\colon s = \sum_{1 \leqslant j \leqslant r} a_{\sigma(j)} \text{ or } s = \sum_{1 \leqslant j \leqslant t} b_{\mu(j)};\ 0 \leqslant r \leqslant n,\ 0 \leqslant t \leqslant m \right\}.$$

The set S is not allowed repetitions; that is, S is not a multiset. Now label the elements of S such that

$$S = \{s_j\colon 0 \leqslant j \leqslant n_{1,2}\} \quad \text{with } s_i \leqslant s_j \text{ for } i \leqslant j.$$

The double-digest implied by the configuration (σ, μ) can be defined by

$$C(\sigma, \mu) = \{c_i(\sigma, \mu)\colon\ c_i(\sigma, \mu) = s_j - s_{j-1} \text{ for some } 1 \leqslant j \leqslant n_{1,2}\},$$

where it is assumed as usual that the set is ordered in the index i. The problem then is to find a configuration (σ, μ) such that $\boldsymbol{C} = C(\sigma, \mu)$. As discussed below, this problem lies in the class of NP-complete problems conjectured to have no polynomial-time solution.

In order to implement a simulated annealing algorithm, an energy function and a neighborhood structure are required. The energy function is a chi-square-like function

$$f(\sigma, \mu) = \sum_{1\leqslant i\leqslant n_{1,2}} (c_i(\sigma, \mu) - c_i)^2/c_i.$$

Note that if all measurements are free of error then f attains its global minimum value of zero for at least one choice (σ, μ). Following Goldstein and Waterman (1987), we define the set of neighbors of a configuration (σ, μ) by

$$N(\sigma, \mu) = \{(\tau, \mu)\colon\ \tau \in N(\sigma)\} \cup \{(\sigma, \nu)\colon\ \nu \in N(\mu)\},$$

where $N(\rho)$ are the neighbors used in studies of the travelling salesman problem (Bonomi and Lutton 1984).

With these ingredients, the algorithm was tested on exact, known data from the bacteriophage lambda with restriction enzymes BamHI and EcoRI, yielding a problem size of $|A|! \times |B|! = 6!6! = 518\,400$. See Daniels et al. (1983) for the complete sequence and map information about lambda. Temperature was not lowered at the rate $c/\log(n)$ as suggested by the theorem in Geman and Geman (1984), but for reasons of practicality was instead lowered exponentially. On three separate trials using various annealing schedules the solution was located after 29 702, 6895, and 3670 iterations from random initial configurations.

4.2.2. Multiplicity of solutions

In many instances, the solution to the double-digest problem is not unique. Consider, for example,

$$\boldsymbol{A} = \{1, 3, 3, 12\},$$
$$\boldsymbol{B} = \{1, 2, 3, 3, 4, 6, \},$$
$$\boldsymbol{C} = \{1, 1, 1, 1, 2, 2, 2, 3, 6\}.$$

This problem, of size $4!6!/2!2! = 4320$, admits 208 distinct solutions. That is, there are 208 distinct orders which produce $\boldsymbol{C}$. We now demonstrate that this phenomenon is far from isolated.

Below, we use the Kingman subadditive ergodic theorem to prove that the number of solutions to the double-digest problem increases exponentially as a function of length under the probability model stated below.

For reference, a version of the subadditive ergodic theorem is given here (Kingman 1973). For s, t non-negative integers with $0 \leqslant s \leqslant t$, let $X_{s,t}$ be a collection of random variables which satisfy

1. whenever $s < t < u$, $X_{s,u} \leqslant X_{s,t} + X_{t,u}$;
2. the joint distribution of $\{X_{s,t}\}$ is the same as that of $\{X_{s+1,t+1}\}$;
3. the expectation $g_t = E[X_{0,t}]$ exists and satisfies $g_t \geqslant -Kt$ for some constant K and all $t > 1$.

Then the finite $\lim_{t\to\infty} X_{0,t}/t = \lambda$ exists with probability one and in the mean.

Theorem 4.3. *Assume the sites for two restriction enzymes are independently distributed with cut probabilities p_1, p_2 respectively and $p_i \in (0,1)$. Let $Y_{s,t}$ be the number of solutions between the sth and the tth coincident sites. Then there is a constant $\lambda > 0$ such that*

$$\lim_{t\to\infty} \frac{\log(Y_{0,t})}{t} = \lambda.$$

Proof. Let a coincidence be defined to be the event that a site is cut by both restriction enzymes; such an event occurs at each site independently with probability $p_1p_2 > 0$, and at site 0 by definition. On the sites $1,2,3,\ldots$, there will be an infinite number of such events. For $s,u = 0,1,2,\ldots$, $0 \leqslant s \leqslant u$ we may consider the double-digest problem for only that segment located between the sth and uth coincidences. Let $Y_{s,u}$ denote the number of solutions to the double-digest problem for this segment.

Suppose $s < t < u$. A solution for the segment between the sth and tth coincidences and a solution for the segment between the tth and uth coincidences can be combined to yield a solution for the segment between the sth and uth coincidences. Thus

$$Y_{s,u} \geqslant Y_{s,t} Y_{t,u}.$$

We note that the inequality may be strict as $Y_{s,u}$ counts solutions given by orderings where fragments initially between, say, the sth and tth coincidences now appear in the solution between the tth and uth coincidences. Letting

$$X_{s,t} = -\log Y_{s,t},$$

we have $s \leqslant t \leqslant u$ implies $X_{s,u} \leqslant X_{s,t} + X_{t,u}$.

Additional technical details can be established to complete the proof of the theorem. □

4.2.3. Computational complexity

Given the definition of a restriction map as permutations of the various digest fragments, it is no surprise that the double-digest problem is NP-complete. See Garey and Johnson (1979) for definitions.

Theorem 4.4. *The double-digest problem is NP-complete.*

Proof. It is clear that the DDP described above is in the class NP, as a nondeterministic algorithm need only guess a configuration (σ, μ) and check in polynomial time if $C(\sigma, \mu) = C$. The number of steps to check this is in fact linear. To show that DDP is NP-complete, the partition problem is transformed to DDP.

In the partition problem, known to be NP-complete (Garey and Johnson 1979), a finite set Q, say $|Q| = n$ is given along with a positive integer $s(q)$ for each $q \in Q$, and we wish to determine whether there exists a subset $Q' \subset Q$ such that

$$\sum_{q \in Q'} s(q) = \sum_{q \in Q - Q'} s(q).$$

If $\sum_{q \in Q} s(q) = J$ is not divisible by two, there can be no such subset Q'. Otherwise, input to problem DDP the data

$$\begin{aligned} \boldsymbol{A} &= \{s(a_k) : 1 \leqslant k \leqslant n\}, \\ \boldsymbol{B} &= \{J/2, J/2\}, \\ \boldsymbol{C} &= \{s(q) : q \in Q\}. \end{aligned}$$

It is clear that any solution to problem DDP with this data yields a solution to the partition problem through the order of the implied digest $\boldsymbol{C}$. Therefore DDP is NP-complete. $\square$

Biologists have routinely been solving DDPs for inexact length measurements. They of course are generally unaware of the results presented here, and search for the length and enzymes that allow a solution. To contribute usefully to this field the challenge is to find algorithms that will extend their capabilities.

References

Bender, E.A.
[1974] Asymptotic methods in enumeration, *SIAM Rev.* **16**, 485–515.

Benzer, S.
[1959] On the topology of genetic fine structure, *Proc. Nat. Acad. Sci. U.S.A.* **45**, 1607–1620.

Bonomi, E., and J.L. Lutton
[1984] The N-city travelling salesman problem: Statistical mechanics and the Metropolis algorithm, *SIAM Rev.* 26, 551–568.

Booth, K.S., and G.S. Leuker
[1976] Testing for the consecutive ones property, interval graphs, and graph planarity using PO algorithms, *J. Comput. Syst. Sci.* **13**, 335–379.

Botstein, D., R.L. White, M. Skolnick and R. Davis
[1980] Construction of a genetic linkage map in man using restriction fragment length polymorphisms, *Amer. J. Human Genet.* **32**, 314–331.

Carrillo, H., and D. Lipman
[1988] The multiple sequence alignment problem in biology, *SIAM J. Appl. Math.* **48**, 1073–1082.

Daniels, D., J. Schroeder, W. Szybalski, F. Sanger, A. Coulson, G. Hong, D. Hill, G. Peterson and F. Blattner
[1983] Complete annotated lambda sequence, in: *Lambda II*, eds. R.W. Hedrix, J.W. Roberts and F.W. Weisberg (Cold Spring Harbor Laboratory, Cold Spring Harbor, NY).

Galil, Z., and R. Giancarlo
[1989] Speeding up dynamic programming with applications to molecular biology, *Theor. Comput. Sci.* **64**, 107–118.

Garey, M.R., and D.S. Johnson
[1979] *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W.H. Freeman, San Francisco, CA).

Geman, S., and D. Geman
[1984] Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721–741.

Goldstein, L., and M.S. Waterman
[1987] Mapping DNA by stochastic relaxation, *Adv. in Appl. Math.* **8**, 194–207.

Golumbic, M.C.
[1980] *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York).

Greene, C.
[1988] Posets of shuffles, *J. Combin. Theory A* **47**, 191–206.

Griggs, J.R., P.J. Hanlon and M.S. Waterman
[1986] Sequence alignments with matched sections, *SIAM J. Algebraic Discrete Methods* **7**, 604–608.

Griggs, J.R., P.J. Hanlon, A.M. Odlyzko and M.S. Waterman
[1990] On the number of alignments of k sequences, *Graphs Combin.* **6**, 133–146.

Howell, J.A., T.F. Smith and M.S. Waterman
[1980] Computation of generating functions for biological molecules, *SIAM J. Appl. Math.* **39**, 119–133.

Isono, K., Y. Kohara and K. Akiyama
[1987] The physical map of the whole *E. coli* chromosome: application of a new strategy for rapid analysis and sorting of a large genomic library, *Cell* **50**, 495–508.

Kingman, J.F.C.
[1973] Subadditive ergodic theory, *Ann. Probab.* **1**, 883–909.

Kruskal, J.B., and D. Sankoff
[1983] *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison* (Addison-Wesley, Reading, MA).

Lander, E., and D. Botstein
[1986] Strategies for studying heterogeneous genetic traits in humans by using a linkage map of restriction fragment length polymorphisms, *Proc. Nat. Acad. Sci. U.S.A.* **83**, 73–53.

Laquer, H.T.
[1981] Asymptotic limits for a two-dimensional recursion, *Studia Appl. Math.* **64**, 271–277.

Miller, W., and E.W. Myers
[1988] Sequence comparison with concave weighting functions, *Bull. Math. Biol.* **50**, 97–120.

Nathans, D., and H.O. Smith
[1975] Restriction endonucleases in the analysis and restructuring of DNA molecules, *Ann. Rev. Biochem.* **44**, 273–293.

Sloan, N.J.A.
[1973] *A Handbook of Integer Sequences* (Academic Press, New York).

Stein, P.R., and M.S. Waterman
[1978] On some new sequences generalizing the Catalan and Motzkin numbers, *Discrete Math.* **26**, 261–272.

Taylor, W.R.
[1987] Multiple sequence alignment by a pairwise algorithm, *Comput. Appl. Biosci.* **3**, 81–87.

Vingron, M., and P. Argos
[1989] A fast sensitive multiple sequence alignment algorithm, *Comput. Appl. Biosci.* **5**, 115–121.

Waterman, M.S.
[1984] General methods of sequence comparison, *Bull. Math. Biol.* **46**, 473–500.
[1986] Multiple sequence alignment by consensus, *Nucleic Acids Res.* **14**, 9095–9102.
[1989] *Mathematical Methods for DNA Sequences*, ed. M.S. Waterman (CRC Press, Boca Raton, FL).

Waterman, M.S., and J.R. Griggs
[1986] Interval graphs and maps of DNA, *Bull. Math. Biol.* **48**, 189–195.

Waterman, M.S., and R. Jones
[1990] Consensus methods for DNA and protein sequence alignments, in: *Methods in Enzymology*, Vol. 183, ed. R. Doolittle (Academic Press, New York).

Waterman, M.S., and M.D. Perlwitz
[1984] Line geometries for sequence comparisons, *Bull. Math. Biol.* **46**, 567–577.

Waterman, M.S., and T.F. Smith
[1986] Rapid dynamic programming algorithms for RNA secondary structure, *Adv. in Appl. Math.* **7**, 455–464.

Zuker, M., and D. Sankoff
[1984] RNA secondary structures and their prediction, *Bull. Math. Biol.* **46**, 591–622.