Chapter 1
Open Combinatorial Problems in Computational Molecular Biology

Pavel A.Pevzner[1]

Department of
Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802

Michael S. Waterman[2]

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

Abstract

In the last few years theoretical computer scientists have found new challenges in computational molecular biology. We discuss recent advances and present some open combinatorial problems in different areas of computational molecular biology such as genome rearrangements, DNA physical mapping, DNA sequencing and sequence comparison.

1 Introduction

Today molecular biology is in the midst of a major paradigm shift, a shift driven in large extent by computer science. Having in many respects already become an *informational* science, molecular biology is rapidly becoming a *computational* science as well.

Computational molecular biology is undergoing rapid development. In 1989 the first "big science" project in biology (*the Human Genome Project*) was launched. It is aimed at determining the 100.000 genes that are contained in three billion letters of the human genome. "Raw" genetic texts are likely to become the main research material of biologists over the next decades.

Many biologists have already realized this shift and are known as "computational molecular biologists". On the other hand, in recent years, theoretical computer scientists have found exciting challenges in DNA sequence analysis, and a growing number of them are trying to contribute to the genome studies. In some classical computer science disciplines (like combinatorial pattern matching), computational molecular biology has already become a major source of new open problems.

The spectrum of computer science techniques used in computational molecular biology ranges from graph theory and computational geometry to flows in networks and computer vision. A number of newly developing experimental techniques are likely to further widen this spectrum. Applications of computational molecular biology include DNA sequence comparison, DNA physical mapping, DNA sequencing, DNA fingerprinting, DNA linguistics, molecular evolution, fast searches of biological databases, predictions of protein and RNA secondary and tertiary structure, prediction of genes in genetic texts, etc. The book by Waterman, 1995 is a survey of modern computational molecular biology techniques and applications.

Computational molecular biology started in the 1970's as an area that aimed to solve the problems arising in the biological labs. An early striking example of a biological discovery made with a computer happened in 1984 when biologists used a simple combinatorial technique to compare a newly discovered cancer-causing gene to all known genes. To their astonishment, the cancer-causing gene matched a normal gene involved in growth and development. From this, it became clear that cancer might be caused by a normal growth gene being switched on at the wrong time.

Almost every area of theoretical computer science started as an attempt to solve applied problems later became more theoretically-oriented. These theoretical aspects might even become more important and scientifically valuable than the original applications that motivated the entire area. (Flows in networks is the most obvious example). In the last few years molec-

[1]Supported by NSF Young Investigator Award, NSF grant CCR-9308567, NIH grant 1R01 HG00987 and DOE grant DE-FG02-94ER61919

[2]Supported by NSF grant DMS 90-05833 and NIH grant GM36230.

0-8186-6915-2/95 $04.00

ular biology motivated a large number of interesting combinatorial problems. In fact, one can assert that *computational molecular biology is a new area of theoretical computer science motivated by molecular biology.*

In this paper we review some open combinatorial problems in computational molecular biology. Many of these problems have already appeared (explicitly or implicitly) in the research papers. Some of them are rather simple while the others have been under attacks for a decade or more. It is our goal to bring a body of interesting and challenging problems to the attention of the community.

The choice of the problems is motivated by the authors' research interests and, as a result, these problems do not uniformly cover all areas of computational molecular biology. Some of them are rather theoretical and have very limited biological applications; the solutions of others might provide valuable tools for biologists or might lead to new biological discoveries. This blend of theoretical and applied problems is a distinctive and fascinating feature of computational molecular biology.

The topics covered in this paper are genome rearrangements, (small-scale) DNA physical mapping, DNA sequencing and sequence comparison. Because of space limitations we do not discuss many other combinatorial topics in computational molecular biology, most notably evolutionary tree reconstruction and large-scale physical mapping (see Farach et al, 1993 and Karp, 1993 for recent reviews). A more complete problem set will be presented in the journal version of this paper.

2 Genome Rearrangements

Genus *Lobelia* comprises over 350 species that range from small, slender herbs to woody, giant-rosette plants. Fig. 1 presents the order of genes in *Tobacco* and *Lobelia fervens* chloroplast genomes with a hypothetical sequence of rearrangement events (Knox et al., 1993) during evolution of *Lobelia fervens* from a tobacco-like ancestral genome.

It is not so easy to verify that the evolutionary events in Fig. 1 represent the *shortest* sequence of inversions transforming the *Tobacco* permutation into the *Lobelia fervens* permutation and until recently there were no computer science results allowing a biologist to analyze genome rearrangements. Since it is often found that the order of genes is much more conserved than the DNA sequence (Franklin,

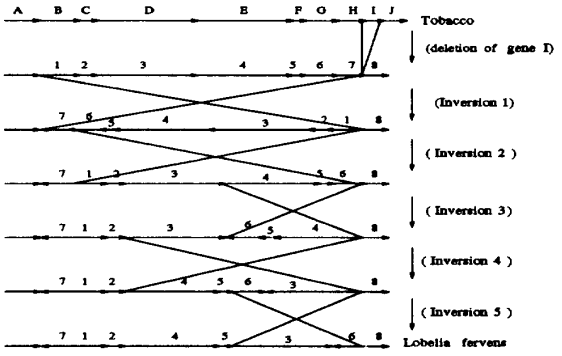


Figure 1: Evolution of *Lobelia fervens*

1985), an approach based on comparison of *gene orders* (Sankoff et al.,1990,1992) versus traditional comparison of *DNA sequences* might be a method of choice for many "hard-to-analyze" genomes.

Studies of gene arrangements in plant organelles were pioneered by Palmer and co-authors who compared gene orders to analyze plant molecular evolution. Every such study involves solving a combinatorial "puzzle" to find a shortest series of genome rearrangements to transform one genome into another. In the cases of genomes consisting of small number of "conserved blocks", Palmer and co-authors were able to find the shortest scenarios for rearrangements. However, for genomes consisting of more than 10 blocks, exhaustive search over all potential solutions is far beyond the possibilities of "pen-and-pencil" methods (Watterson et al., 1982).

With the advent of large-scale DNA mapping and sequencing, the number of biological problems similar to the one presented in Fig. 1 is rapidly growing in different areas, including evolution of plant chloroplast DNA (Raubeson and Jansen, 1992, Hoot and Palmer, 1994), plant mitochondrial DNA(Palmer and Herbon, 1988, Fauron and Havlik, 1989), animal mitochondrial DNA (Hoffman et al., 1992, Sankoff et al., 1992), virology (Koonin and Dolya, 1993, Hannenhalli et al., 1994), *Drosophila* genetics (Carson, 1983, Whiting et al, 1989) and comparative physical mapping (Lyon, 1988).

In the first approximation, the order of genes in two organisms is represented by permutations $\pi = (\pi_1 \pi_2 \ldots \pi_n)$ and $\sigma = (\sigma_1 \sigma_2 \ldots \sigma_n)$. A *reversal* ρ of an interval $[i, j]$ is the permutation

$$\begin{pmatrix} 1 & 2 & \ldots & i-1 & \mathbf{i} & \mathbf{i+1} & \ldots & \mathbf{j-1} & \mathbf{j} & j+1 & \ldots & n \\ 1 & 2 & \ldots & i-1 & \mathbf{j} & \mathbf{j-1} & \ldots & \mathbf{i+1} & \mathbf{i} & j+1 & \ldots & n \end{pmatrix}$$

Clearly $\pi \cdot \rho$ has the effect of reversing genes

$\pi_i, \pi_{i+1}, \ldots \pi_j$.

Given permutations π and σ, the *reversal distance problem* is to find a sequence of reversals $\rho_1, \rho_2, \ldots, \rho_t$ such that $\pi \cdot \rho_1 \cdot \rho_2 \cdots \rho_t = \sigma$ and t is minimum. We call t the *reversal distance* between π and σ. Note that reversal distance between π and σ equals the reversal distance between $\sigma^{-1}\pi$ and the *identity* permutation ι. *Sorting π by reversals* is the problem of finding reversal distance $d(\pi)$, between π and ι.

Reversals generate the *symmetric group* S_n. Given an arbitrary permutation π from S_n, we seek a shortest product of *generators* $\rho_1 \cdot \rho_2 \cdots \rho_t$ that equals π. Even and Goldreich, 1981 show that given a set of generators of a permutation group and a permutation π, determining the shortest product of generators that equals π is NP-hard. Jerrum, 1985 proves that the problem is PSPACE-complete, and remains so, when restricted to two generators. In our problem, the generator set is fixed. However Kececioglu and Sankoff, 1993 conjectured that

Problem 1 *Sorting by reversals is NP-hard.*

Let $i \sim j$ if $|i - j| = 1$. Extend a permutation $\pi = \pi_1\pi_2 \ldots \pi_n$ by adding $\pi_0 = 0$ and $\pi_{n+1} = n + 1$. We call a pair of consecutive elements π_i and π_{i+1}, $0 \leq i \leq n$, of π an *adjacency* if $\pi_i \sim \pi_{i+1}$, and a *breakpoint* if $\pi_i \not\sim \pi_{i+1}$. Denote the number of breakpoints in π as $b(\pi)$. For example, the *Lobelia fervens* permutation π = 71245368 has 6 breakpoints (0,7), (7,1), (2,4), (5,3), (3,6), (6,8) and 3 adjacencies (1,2), (4,5), (8,9).

A reversal ρ affects the breakpoints of π only at the endpoints of ρ and therefore every reversal can eliminate *at most* two breakpoints. It implies $d(\pi) \geq \frac{b(\pi)}{2}$. Kececioglu and Sankoff, 1993 conjectured that even a much weaker version of sorting by reversals problem is NP-complete.

Problem 2 *Deciding $d(\pi) = \frac{b(\pi)}{2}$ is NP-complete.*

Kececioglu and Sankoff, 1993 found an approximation algorithm for sorting by reversals with performance guarantee 2. They also devised efficient bounds, allowing them to solve the reversal distance problem optimally or almost optimally for n up to 50. This range covers the biologically important case of mitochondrial genomes. Later Bafna and Pevzner, 1993 achieved a performance guarantee of $\frac{7}{4}$ for sorting by reversals.

Problem 3 *Improve the performance guarantee of sorting by reversals*

When the sequence of genes contains duplications, sorting *permutations* by reversals is transformed into the problem of sorting *words* by reversals. For example, the shortest sequence of reversals to transform the word 43132143 into the word 42341314 involves two inversions 43132143 $\rightarrow$ 42313143 $\rightarrow$ 42341314. No performance guarantee algorithms for this problem are known.

Problem 4 *Devise a performance guarantee algorithm for sorting words by reversals.*

Define a *strip* of a permutation π as an interval $[i, j]$ such that $(i-1, i)$ and $(j, j+1)$ are breakpoints, and no breakpoint lies between them. It is natural to suspect that in a shortest series of reversals we never have to consider reversals that cut strips. This however is false; for example the permutation 3412 requires three reversals if we do not cut strips, yet it can be sorted with two. Nevertheless Kececioglu and Sankoff, 1993 made the following conjecture:

Problem 5 *For every permutation there exists a shortest series of reversals that does not cut strips of three or more elements.*

The following, even stronger conjecture was also proposed by Kececioglu and Sankoff, 1993

Problem 6 *For every permutation there exists a shortest series of reversals that never increases the number of breakpoints.*

Define $D(n) = \max_{\pi \in S_n} d(\pi)$ to be the *reversal diameter* of the symmetric group of order n. Gollan conjectured that $D(n) = n - 1$ and that only one permutation γ_n, and its inverse, γ_n^{-1} require $n - 1$ reversals to be sorted. For $n \leq 11$, Gollan verified this conjecture using extensive computations. Kececioglu and Sankoff, 1993 developed lower bounds for reversal distance, allowing them to verify Gollan's conjecture for $n \leq 200$ for n mod 3 = 1. Finally Bafna and Pevzner, 1993 proved Gollan's conjecture and further, studied the problem of expected reversal distance $E(d)$ between two random permutations. They demonstrated that reversal distance between two random permutations is very close to the reversal diameter of the symmetric group, i.e. $E(d) \geq \left(1 - \frac{4.5}{\log n}\right) n$. This result indicates that reversal distance provides a good separation between related and non-related sequences in molecular evolution studies.

Problem 7 *Improve the lower bound and derive a non-trivial upper bound for expected reversal distance. Estimate the variance of the reversal distance.*

A gene is a piece of DNA sequence and the sequence has a *reading direction*. This comment leads to the sorting by reversals problem for *signed* permutations. The *Lobelia fervens* permutation corresponds to the signed permutation $(-7,+1,+2,+4,+5,+3,-6,+8)$. In the biologically more relevant signed case, every reversal of fragment $[i,j]$ changes the signs of the elements within that fragment. We are interested in the minimum number of reversals to transform the signed permutation π into the identity signed permutation $(+1,+2,\ldots,+n)$. Bafna and Pevzner, 1993 devised an approximation algorithm for sorting signed permutations by reversals with performance ratio $\frac{3}{2}$. In the case of signed permutations the performance guarantee algorithms approximate the reversal distance with an astonishing accuracy for both simulated data (Kececioglu and Sankoff, 1994) and biological applications (Bafna and Pevzner, 1995b).

Problem 8 *Improve the performance guarantee for sorting of signed permutations by reversals.*

Kececioglu and Sankoff, 1994 studied the bounds for the diameter $D_s(n)$ in the case of signed permutations and proved that

$$n-1 \leq D_s(n) \leq n$$

They also conjectured

Problem 9 *For signed circular permutations, $D_s(n) = n$ for sufficiently large n.*

A characterization of "hard-to-sort" signed permutations (i.e. n-element permutations π such that $d(\pi) = D_s(n)$) is still unknown.

Problem 10 *Characterize the set of signed permutations on n elements such than $d(\pi) = D(n)$.*

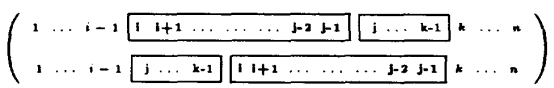
Gates and Papadimitriou, 1979 and Gyori and Turan, 1979 studied *sorting by prefix reversals* problem (also known as the *pancake flipping problem*): Given an arbitrary permutation π, find $d_{pref}(\pi)$, the minimum number of reversals of the form $[1,i]$ that sort π. Note that all reversals in Fig. 1 are in fact originated at one point (left-side of the fragment 8) and are essentially prefix reversals corresponding to a *hot point of mutations*. Gates and Papadimitriou, 1979 are concerned with bounds on the *prefix reversal diameter* of the symmetric group, $D_{pref}(n) = \max_{\pi \in S_n} d_{pref}(\pi)$. They show that $D_{pref}(n) \leq \frac{5}{3}n + \frac{5}{3}$, and that for infinitely many n, $D_{pref}(n) \geq \frac{17}{16}n$. They also conjectured that a particular *Gates-Papadimitriou* permutation on n elements requires at least $\frac{19}{16}n$ reversals to be sorted. Heidari and Sudborough, 1993 disproved their conjecture by describing $\frac{18}{16}n + 2$ reversals sorting the Gates-Papadimitriou permutation.

Problem 11 *Improve lower and upper bounds for sorting by prefix reversals.*

Problem 12 *Find the prefix reversal diameter of the symmetric group.*

Genomes evolve not only by inversions but by transpositions, deletions, insertions and duplications of fragments. As a first approximation, transpositions in genome rearrangements can be modelled in the following way.

For a permutation π, a *transposition* $\rho(i,j,k)$ (defined for all $1 \leq i < j \leq n+1$ and all $1 \leq k \leq n+1$ such that $k \notin [i,j]$) "inserts" an interval $[i,j-1]$ of π between π_{k-1} and π_k , i.e. $\rho(i,j,k)$ corresponds to a permutation



Given permutations π and σ, the *transposition distance problem* is to find a series of transpositions $\rho_1, \rho_2, \ldots, \rho_t$ such that $\pi \cdot \rho_1 \cdot \rho_2 \cdots \rho_t = \sigma$ and t is minimum. We call t the *transposition distance* between π and σ. *Sorting π by transpositions* is the problem of finding transposition distance $d(\pi)$, between π and the identity permutation ι. A well-known polynomially-solvable variant of sorting by transpositions is sorting by transpositions $\rho(i,i+1,i+2)$ where the operation is an exchange of adjacent elements. For this problem polynomial algorithms exist for both linear (bubble sort) and circular permutations (Jerrum, 1985). Aigner and West, 1987 found a simple algorithm for sorting by transpositions $\rho(1,2,i)$ when the operation is reinsertion of the first element,

Sorting by transpositions is a somehow harder combinatorial problem than the previously studied sorting by reversals; in particular the transposition diameter $D_t(n)$ of the symmetric group is still unknown. Bafna and Pevzner, 1995a devised a 1.5 performance guarantee algorithm for sorting by transpositions and demonstrated that

$$\frac{n}{2} \leq D_t(n) \leq \frac{3n}{4}$$

Problem 13 *Find the transposition diameter of the symmetric group.*

Problem 14 *Devise an improved performance guarantee algorithm for sorting by transpositions.*

Problem 15 *Devise an approximation algorithm for sorting by reversals and transpositions.*

Eucaryotic genomes typically consist of many chromosomes. Chromosomal rearrangements include not only inversions and transpositions but *translocations* as well. Translocations exchange genetic material between different chromosomes.

Recent advances in large-scale comparative genetic mapping offer exciting prospects for understanding mammalian genome evolution. The large number of conserved segments in the maps of man and mouse suggest that multiple chromosomal rearrangements have occurred since the divergence of lineages leading to humans and mice. Nadeau and Taylor, 1984 estimated that just 178 ± 39 chromosomal rearrangements have occurred since this divergence. This estimate survived a ten-fold increase in the amount of the comparative man/mouse mapping information; the new estimate based on the latest data (Copeland et al., 1993) changed little compared to Nadeau and Taylor, 1984.

However, the Nadeau and Taylor, 1984 analysis is non-constructive and does not lead to a scenario of chromosomal rearrangements.

Problem 16 *Find a most parsimonious scenario of chromosomal rearrangements transforming the mouse genome into the human genome.*

Problem 17 *Devise an algorithm to analyze chromosomal rearrangements.*

See Kececioglu and Ravi, 1995 for a discussion of recent progress on the last problem.

3 DNA physical mapping

Hamilton Smith discovered in 1970 that the *restriction enzyme Hind* II cleaves DNA molecules at every occurrence of a sequence GTGCAC or GTTAAC. Since that time, physical maps presenting a DNA molecule with the points of cleavage (*sites*) by restriction enzymes have become a fundamental data structure in molecular biology.

Several experimental approaches to small-scale DNA physical mapping exist, each with its own advantages and disadvantages. Most of them correspond to the following combinatorial problem arising also in crystallography. If $X \subset R^1$ is a multiset (a finite set with repetitions allowed) let ΔX denote the multiset of *all* pairwise distances between points in X: $\Delta X = \{|x_1 - x_2| : x_1, x_2 \in X\}$. In the small-scale DNA physical mapping some subset $E \subset \Delta X$ corresponding to the experimental data about fragment lengths is given, and the problem is to reconstruct X from the knowledge of E alone.

In the (computationally) simplest case $E = \Delta X$, corresponding to the *Partial Digest Problem (PDP)*, the experiment provides data about *all* pairwise distances between restriction sites. The problem is known also as *turnpike* problem in computer science (Weiss,1992). Rosenblatt and Seymour, 1982 gave a pseudo-polynomial algorithm for PDP based on factorization of polynomials. Later Skiena et al., 1990 gave a simple backtracking algorithm which performs well in practice. Weiss,1992 even asked whether this backtracking algorithm runs in polynomial time on any instance of PDP. However Zhang,1994 constructed an exponential example for the backtracking algorithm.

Problem 18 *Does a polynomial algorithm exist for PDP?*

Given ΔX, it is not possible to completely reconstruct X. For example, for any solution X the sets $-X$ (*reflection*) and $X + a$ for every $a \epsilon R^1$ (*translation*) are also solutions. There are less trivial examples of this nonuniqueness, We say that two multisets X and Y are homometric if $\Delta X = \Delta Y$. For example, the sets $\{0, 1, 3, 8, 9, 11, 13, 15\}$ and $\{0, 1, 3, 4, 5, 7, 12, 13, 15\}$ are homometric and are not transformed into each other by reflections and translations (*strongly homometric sets*). Skiena et al., 1990 proved that the maximum number $H(n)$ of strongly homometric sets on n elements is bounded by $\frac{1}{2}n^{0.6309} \leq H(n) \leq \frac{1}{2}n^{2.5}$. The upper bound seems to be very pessimistic.

Problem 19 *(Skiena et al., 1990) Derive tighter bounds for $H(n)$.*

Another way to construct a physical map is to measure the fragment lengths from a digestion of the DNA by each of the two enzymes A and B singly, and then by the two enzymes applied together. The problem of determining the positions of the cuts from fragment length data is known as the *Double Digest Problem* or *DDP* .

For an arbitrary set X of n elements denote $\delta X \subset \Delta X$ to be the set of $n-1$ distances between *consecutive* elements of X. In DDP the multiset $X \subset [0, t]$ is partitioned into two subsets $X = A \bigcup B$ with $0 \epsilon A, B$ and $t \epsilon A, B$. The experiment provides three sets of lengths: $\delta A, \delta B$ and δX. Double Digest problem is to reconstruct A and B from these data. The real

problem is complicated by the experimental errors in $\delta A, \delta B$ and δX.

Different combinatorial approaches to DDP are known (Waterman and Griggs,1986; Goldstein and Waterman,1987; Allison and Yee, 1988; Pevzner, 1992, 1994) and a number of computer programs for DDP have been suggested (Pearson,1982; Fitch et al., 1983; Pevzner and Mironov,1987; Krawczak,1988, Ho et al.,1990; Grigorjev and Mironov,1990; Yap, 1991). However, despite significant efforts, DDP still cannot be considered resolved. All of the algorithms encounter significant difficulties when applied to finding physical maps with more than 7 or 8 sites for each restriction enzyme. Devising a fast computer program for small-scale DNA physical mapping is still an open problem.

Problem 20 *Devise a fast algorithm and a practical computer program for the Double Digest problem allowing a biologist to build physical maps with up to 20 sites in the presence of experimental errors.*

Goldstein and Waterman, 1987 proved that DDP is NP-hard. They also showed that when the enzyme sites are modelled by a random process, the number of solutions that produce the same single and double digests increases exponentially as the length of DNA increases. Exponential growth of the number of solutions of DDP forces biologists to use $k > 2$ enzymes in experiments. They further combine pairwise maps into a multiple map with k enzymes (Pevzner, 1990). Another approach would be to have three enzymes A, B and C and to get experimental data about the lengths of the fragments in *single* digestions A, B, C, *double* digestions AB, BC, CA and *triple* digestion ABC.

Problem 21 *Devise an algorithm to construct physical maps of three enzymes A, B, C provided seven sets of experimental data (digestions A, B, C, AB, BC, CA and ABC) are given.*

Of course NP-hardness and exponential growth of the number of solutions hamper the development of a fast mapping program. Nevertheless, a new approach allowing reduction of the computational time of mapping algorithms has been suggested by Schmitt and Waterman,1991.

Although the number of solutions grows very fast as the number of sites grows, the majority of solutions are very similar (could be transformed into each other by rearrangements of various regions). Since most of time during map construction is wasted generating 'very similar maps', it would seem reasonable to partition the entire set of physical maps into equivalence classes and to generate only one *basic* map in every equivalence class. Further, all solutions could be generated from basic maps using simple transformations. If the number of equivalence classes were smaller than the number of physical maps, then this approach would allow reduction of computational time for the DDP algorithm.

Schmitt and Waterman, 1991 took the first step in this direction. They introduced an equivalence relation on physical maps. All maps of the same equivalence class are transformed into one another by means of a natural set of *cassette transformations*. Pevzner,1994 generalizes the cassette transformations of Schmitt and Waterman and characterizes equivalence classes under these transformations for pairwise maps. Martin, 1994 further develops this idea to characterize equivalence classes in the case of coincident cuts. However the following problem remains unsolved.

Problem 22 *Given two maps from the same equivalence class, find a shortest series of cassette transformations to transform one map into another.*

Problem 23 *Characterize cassette transformations of multiple maps (3 and more enzymes).*

Computational experiments demonstrate that the number of equivalence classes of physical maps grows with the size of DDP. It raises the following problem:

Problem 24 *Find the expected number of physical maps and equivalence classes of physical maps for a "random" Double Digest problem.*

Pevzner, 1994 proved that the elements of an equivalent class are in one-to-one correspondence with all *alternating Eulerian paths* in a certain bicoloured graph. This result raises the question of whether the size of an equivalent class can be efficiently computed in the terms of the number of alternating Eulerian cycles in bicoloured graphs.

Problem 25 *Compute the number of elements in the equivalence classes of physical maps.*

Another physical mapping technique leads to the *Probed Partial Digestion Problem (PPDP)*. In this method DNA is partially digested with a restriction enzyme thus generating a collection of DNA fragments between any two cutting sites. Afterwards a labeled probe, which attaches to the DNA between two cutting sites, is hybridized to the partially digested DNA,

and the sizes of fragments to which the probe hybridizes are measured. The problem is to reconstruct the positions of the sites from the multiset of measured length.

Formally in the PPDP, a multiset $X \subset [-s,t]$ is partitioned into two subsets $X = A \bigcup B$ with $0 \epsilon A, B$ and $A \subset [-s,0]$, $B \subset [0,t]$; and the experiment provides the multiset $S = \{b - a \ : \ a \epsilon A, b \epsilon B\}$. The problem is to find A and B given S.

Denote by $p(S)$ the number of solutions of PPDP problem for the instance S and define $P(n) = \sup\{p(S) \ : \ |S| = n\}$. Newberg and Naor, 1993 constructed an example demonstrating that $P(n)$ grows faster than $n^{1.73}$. The question about tight lower and upper bounds for $P(n)$ is still open.

Problem 26 *Derive tighter upper and lower bounds for $P(n)$*

4 DNA sequencing

With current laboratory methods, only small DNA fragments of roughly 500 base pairs can be sequenced at a time. Then, from thousands of these fragments (starting at unknown positions), a biologist must to reconstruct the sequence of the whole DNA molecule. The problem is complicated by unavoidable experimental errors in reading small DNA fragments.

The simplest and naive approximation of classical gel-based DNA sequencing (without noise) corresponds to the *shortest superstring problem*: Given a collection of strings $s_1, \ldots, s_m$, find the shortest string s such that each s_i appears as a substring of s. Since the shortest superstring problem is known to be NP-hard (Gallant, 1980), a number of heuristics have been considered. The simplest heuristic is the GREEDY algorithm: Repeatedly merge a pair of strings with maximum overlap until only one string remains. Tarhio and Ukkonen, 1988 and Turner, 1989 defined *compression* of a shortest superstring algorithm as the number of symbols saved by the algorithm compared to simply concatenating all the strings. They also proved that GREEDY achieves at least $\frac{1}{2}$ the compression of an optimal superstring.

Problem 27 *(Teng and Yao,1993) Find an algorithm that achieves better than $\frac{1}{2}$ of optimal compression.*

A performance guarantee with respect to compression does not implies a performance guarantee with respect to length. The first performance guarantee shortest superstring algorithm with respect to the length was proposed by Blum et al., 1991. The Blum et al. 1991 algorithm achieves approximation ratio 3; later in a series of papers Teng and Yao, 1993, Kosaraju et al., 1994, Czumaj et al., 1994, and Armen and Stein, 1994 reduced the performance ratio to 2.75. On the other hand, an example for which GREEDY's approximation ratio is worst than 2 is unknown.

Problem 28 *(Blum et al.,1991) Prove or disprove the conjecture that GREEDY achieves performance guarantee 2.*

Problem 29 *Devise an algorithm with improved performance guarantee for the SSP problem.*

The resolution power of sequencing methods could be improved if classical DNA sequencing is combined with hybridization to additional probes (in the fragments with low accuracy) or sequencing by hybridization. Such experiments provide both *positive* and *negative* examples of substrings. This leads to the following problem.

Let $P = \{s_1, \ldots, s_m\}$ be a set of *positive* strings and $N = \{t_1, \ldots, t_k\}$ be a set of negative strings. We assume that no negative string t_i is a substring of any positive string s_j. A *consistent superstring* for (P, N) is a string s such that each s_i is a substring of s and no t_i is a substring of s.

Jiang and Li, 1993 gave an approximation algorithm for the shortest consistent superstring problem with performance guarantee $\log n$. Later they found a linear approximation for shortest consistent superstring problem for two special cases: (i) When no negative strings contain positive strings as substrings and (ii) when there are only a constant number of negative strings.

Problem 30 *(Jiang and Li, 1993) Does a polynomial time linear approximation algorithm exist for the shortest consistent superstring problem?*

Short fragments read by the conventional sequencing techniques contain errors which lead to complications in the sequence assembly process. Introducing errors leads to the following *shortest k-approximate superstring problem*: Given a set S of strings, find a shortest string w such that each string x in S matches some substring of w with at most k errors. Jiang and Li, 1994 gave a polynomial algorithm for the shortest k-approximate superstring problem with performance guarantee $\log n$.

Problem 31 *Does a polynomial-time linear approximation algorithm exists for the shortest k-approximate superstring problem?*

Suppose we are given a set $\mathcal{S}$ of n *random* strings of fixed size l over an A-letter alphabet. If n is large (of the order of A^l), the length of the shortest common superstring $E(SSP(\mathcal{S}))$ for the set $\mathcal{S}$ is of the order n. If n is small, $E(SSP(\mathcal{S}))$ is of the order $n \cdot l$.

Problem 32 *Find $E(SSP(\mathcal{S}))$ as a function of l, n and A.*

In a more complicated version of the same problem n strings from $\mathcal{S}$ have the fixed lengths $l_1, \ldots, l_n$.

Problem 33 *Find $E(SSP(\mathcal{S}))$ as a function of $l_1, \ldots, l_n$, n and A.*

See Alexander, 1994 for recent progress on the last two problems.

In a slightly different formulation which is more relevant to real DNA sequencing, the set $\mathcal{S}$ of n strings of size l is chosen at random from a *given* sequence.

Problem 34 *Find $E(SSP(\mathcal{S}))$ for the case $\mathcal{S}$ is chosen at random from a given sequence of length m.*

Sequencing by Hybridization (SBH) is an alternative approach to DNA sequencing that was proposed simultaneously and independently by Southern,1988, Bains and Smith,1988, Lysov et al,1988 and Drmanac et al.,1989. The simplest SBH technique can be described as follows

- Attach all possible single stranded oligonucleotides of length k (strings of length k over the A, T, G, C alphabet) to the surface of a substrate, each fragment at a distinct location. This set of oligonucleotides is called the *sequencing chip*.
- Apply a solution containing a labeled single stranded target DNA fragment to the sequencing chip.
- Target DNA fragment hybridizes with those oligonucleotides which are complementary to substrings of length k of the target fragment.
- Detect oligonucleotides that hybridize with the target fragment with a nuclear or spectroscopic detector. The *oligonucleotide content* (k-tuple composition) of the target DNA fragment is obtained.
- Apply a combinatorial algorithm to reconstruct the sequence of the target DNA fragment from the k-tuple composition.

In the very first SBH studies, biologists considered the chip $C(8)$ containing all $4^8 = 65536$ 8-tuples ($k = 8$). For $C(k)$, sequencing by hybridization provides information about k-tuple composition of an unknown DNA fragment (the *spectrum* of DNA fragment), and the problem is to reconstruct the sequence of the DNA fragment from these data. In the case of an *ideal* SBH experiment in which the *exact* count of the number of occurrences of each k-tuple in a target fragment is known, SBH reconstruction with $C(k)$ chips can be reduced to the Eulerian path problem (Pevzner,1989). For *real* SBH experiments, the unavoidable errors in the spectra make reconstruction more complicated.

Problem 35 *Devise an algorithm for SBH sequence reconstruction from data with errors.*

Even in the case of an ideal SBH experiment, the chip $C(8)$ is insufficient for sequencing long DNA fragments. In particular, Pevzner,1989 demonstrated that in this case (no hybridization errors), one can unambiguously reconstruct a 200 nucleotide long sequence only in 94 out of 100 cases. Even for a sequencing chip $C(10)$ containing 4^{10} 10-tuples (such a chip could be fabricated using the photolitographic technique of Affymetrix, Fodor et al.,1991), one can reliably reconstruct a DNA fragment only about 600 nucleotide long. This discouraging result indicates that additional joint efforts by biologists and computer scientists are required to make SBH practical and raises a problem of improving resolution power of sequencing chips.

A *sequencing chip* is a set of words $C = \{c_1, \ldots c_m\}$ over an alphabet $\mathcal{A}$ where m is the *memory* of the chip. Given a string W over alphabet $\mathcal{A}$ and a word c, we write $c \dashv W$ if c occurs in W. A *spectrum* C_W of W with respect to sequencing chip C is the subset of chip C: $C_W = \{c \in C : c \dashv W\}$.

A string W of length n (n-string) is called *unambiguous* with respect to the chip C, if for every other n-string W', $C_W \neq C_{W'}$. Let U be the number of unambiguous n-strings (with respect to a sequencing chip C). Define *resolution power* $r(C, n)$ of a sequencing chip C as the fraction $\frac{U}{|\mathcal{A}|^n}$ of unambiguous strings among all n-strings.

Problem 36 *Given m and n, find a sequencing chip C of memory m maximizing the resolution power $r(C, n)$.*

The attempts to improve the resolution power of sequencing chips lead to the idea of *pools* of oligonucleotides at every position of sequencing chips. In this approach a pool of oligonucleotides can be attached

to the same position of the chip without significant complication of the oligonucleotide synthesis process. A hybridization signal at a particular position of the chip indicates that *at least* one of the oligonucleotides in the pool is complementary to the target DNA fragment. Pevzner et al.,1991 demonstrated that pooling allows one to reduce the memory of the chip by a factor of 5-15 times without significantly decreasing the resolving power.

We define a *probe* to be a set (pool) of words (oligonucleotides). For example, WWS is a probe consisting of 8 trinucleotides:

$$AAG, AAC, ATG, ATC, TAG, TAC, TTA, TTC,$$

where W designates the *weak* nucleotides A or T, while S designates the *strong* nucleotides G or C. RYR is a probe consisting of 8 oligonucleotides:

$$ATA, ATG, ACA, ACG, GTA, GTC, GCA, GCG,$$

where R designates the *purines* A or G, while Y designates the *pyrimidines* T or C.

A (generalized) *sequencing chip* is defined as a set of probes $C = (p_1, \ldots, p_m)$. We write $p \dashv W$ if a probe p contains a word $c \dashv W$. Each DNA sequence, W defines a subset of chip C consisting of the probes occurring in W (*spectrum* of W in C):

$$C_W = \{p \epsilon C : p \dashv W\}.$$

A n-string is called *unambiguous* with respect to the chip C if there are no other n-strings with the same spectrum in C. The resolution power of the chip, $r(C, n)$ is the fraction of unambiguous n-strings. We call chip k-bounded if all oligonucleotides in the chip have length at most k. In the following problem we require the chip to be k-bounded to avoid complications in oligonucleotide synthesis and detection of hybridization signal.

Problem 37 *Given m, n and k, find a k-bounded pooling sequencing chip C with memory m maximizing the resolution power $r(C, n)$.*

Pevzner and Lipshutz,1994 introduce new chip designs for SBH and demonstrate their advantages over $C(k)$. In particular they devised the *binary chip* $C_{bin}(k)$ as the chip with memory $2 \cdot 2^k \cdot 4$ composed of all probes of two kinds:

$$\underbrace{\{W, S\}, \{W, S\}, \ldots \{W, S\}}_{k}, \{A, T, G, C\}$$

and

$$\underbrace{\{R, Y\}, \{R, Y\}, \ldots \{R, Y\}}_{k}, \{A, T, G, C\}.$$

Each probe is a mixture of 2^k oligonucleotides of length $k + 1$.

Problem 38 *Do binary chips provide asymptotically the best resolution power among all $(k + 1)$-bounded sequencing chips?*

Although binary chips provide better resolution power than classical $C(k)$ chips, an efficient algorithm for reconstruction of a DNA fragment from its spectrum over a binary chip is still unknown.

Problem 39 *Devise an efficient algorithm for the SBH sequence reconstruction by binary chips.*

Even large-memory sequencing chips do not guarantee unambiguous reconstruction of a long DNA fragment. To resolve ambiguities in SBH sequence reconstruction a number of additional biochemical experiments were proposed. For example, for the spectrum

$$\{ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT\},$$

two reconstructions are possible ($ATGCGTGGCA$ and $ATGGCGTGCA$). We don't know which 3-tuple (TGC or TGG) follows ATG in the original sequence. Therefore, we can not distinguish between a correct reconstruction and the incorrect reconstruction. If we could conduct an additional biochemical experiment with 4-nucleotide ATGC, we would immediately find the correct reconstruction.

Khrapko et al., 1989 proposed the idea of using additional biochemical experiment called continuous stacking hybridization to resolve branchings in the reconstruction process. *Continuous stacking hybridization* assumes an additional hybridization of short oligonucleotides which continuously extend duplexes formed by the target DNA fragment and the probes from the sequencing chip. In this approach, additional hybridization with a short m-tuple on the chip $C(k)$ provides information about some $(k + m)$-tuples in the sequence. Questions regarding the computational complexity and the resolving power of this approach need further consideration.

Problem 40 *Given a spectrum S which does not provide an unambiguous SBH reconstruction, find the minimum number of continuous stacking hybridization experiments to unambiguously reconstruct the target fragment.*

Problem 41 *Reconstruct the sequence of DNA fragment from a given spectrum and the results of additional continuous stacking hybridizations.*

In light-directed synthesis of high-density sequencing chips (Fodor et al., 1991), synthesis errors result from the unintended illumination of chip regions that should remain dark. Most synthesis errors occur at the borders of illuminated regions, where light diffraction, internal reflection, and scattering produce the most unintended illumination. As a result, at the border of illuminated regions the outcome of the light-directed synthesis is uncertain. This uncertainty gives rise to probes with unknown sequences and unknown lengths, that may hybridize to a target DNA strand, thus complicating interpretation of the experimental data. Methods are being sought to minimize the lengths of these borders, so that the level of uncertainty is reduced. A combinatorial synthesis strategy based on two-dimensional Gray codes is optimal for minimizing the border length for $C(k)$ sequencing chips (Feldman and Pevzner, 1994). However, for an *arbitrary* chip, the problem of a chip design minimizing the overall border lengths of photolithographic masks remains unsolved.

Problem 42 *Find an arrangement of oligonucleotides on a chip minimizing the overall border lengths of the masks for photolithographic chip design.*

5 DNA sequence comparison

Biologists looking for similarities between different sequences try to find sequence *alignments*. Dynamic programming is a usual tool for finding alignments in computational molecular biology (Smith and Waterman, 1981, Miller and Myers, 1988, Galil and Giancarlo, 1989). One of the central difficulties in DNA and protein sequence comparison is the choice of alignment penalty parameters. There is a considerable disagreement about how to weight matches, mismatches, insertions/deletions and gaps in sequence alignment. Ideally, we would like to compute optimal alignments for the infinite set of all possible parameter values. The goal of *parametric sequence comparison* is to partition the parameter space into regions such that in each region one alignment is optimal throughout and such the regions are maximal for this property. Fitch and Smith, 1983 were able to find all regions in decomposition of the parameter space for two hemoglobin genes where the penalty for a gap of length k is $\alpha + \beta k$ and α and β vary. The first algorithms for parametric sequence alignment were suggested by Gusfield et al., 1992 and Waterman et al., 1992.

An alignment is characterized by the number of *matches* v, *mismatches* w and insertions/deletions (*indels*) u. (See Waterman, 1989 for a review). In the simplest sequence alignment problem, the mismatches are scored by the alphabet substitution function

$$s(x,y) = \begin{cases} 1, & x = y \\ -\mu, & x \neq y \end{cases}$$

and the indels are scored by $-\delta$. The score of alignment is defined to be

$$v = v - \mu w - \delta u.$$

A given choice of parameter values μ and δ defines an optimal alignment (not necessarily unique). Any alignment that is optimal for fixed values (μ, δ) is optimal in a certain region in the (μ, δ) space. Hence, the 2-parameter space is decomposed into *optimal regions* such that in every region one alignment is optimal throughout, and the regions are maximal for this property. Gusfield et al.,1992 proved that the number of convex polygons in the decomposition for *global* alignment is bounded by $O(n^{2/3})$. This estimate looks rather pessimistic and no example of sequences with $O(n^{2/3})$ regions in the parameter space decomposition is known.

Problem 43 *Improve the bound on the number of regions in the parameter space decomposition for global alignment.*

Problem 44 *Derive bounds for the expected number of regions in the parameter space decomposition for global alignment of two random sequences of length n.*

Parameter space decompositions for *local* alignment usually contain more regions compared to parameter space decompositions for global alignment. Arratia and Waterman, 1994 described a partition of parameter space for local alignment into *linear* and *logarithmic* areas where the score of optimal alignments grow linearly and logarithmically, correspondingly. Vingron and Waterman, 1994 further studied the links between the parametric sequence alignment and the Arratia-Waterman phase transition between linear and logarithmic areas. In this connection it is interesting to study the parameter decomposition of the logarithmic area.

Problem 45 *Derive bounds for the expected number of regions in space decomposition in the logarithmic area for local alignment of two random sequences of length n.*

Vingron and Waterman, 1994 also observed some correlations between the *combinatorial* structure of the tessellations of the parameter space and the *statistical* phase transition curve in the parameter space (Waterman et al.,1987, Arratia and Waterman, 1994). A theoretical substantiation of this correlation remained an open problem.

Problem 46 *Characterize the phase transition curve in terms of the combinatorial structure of the tessellations of the parameter space.*

The Gusfield et al.,1992 algorithm for global parametric sequence alignment of two sequences of length n runs in $O(kn^2)$ time where k is the number of regions. Hence, the amortized cost for finding a single region is the time to find a *single* optimal alignment within that region. For local parametric sequence alignment Gusfield et al.,1992 finds all regions in $O(n^3 \log n)$ per region which is costly.

Problem 47 *Devise a fast algorithm for space decomposition in local sequence alignment.*

The above problems are related to parametric sequence alignment for the simplest scoring schema (no gap penalties). For more general scoring schemas, no efficient algorithm for parametric sequence alignment is known.

Problem 48 *Develop a fast algorithm for space decomposition in local sequence alignment with a more general scoring schema.*

Similar to sequence alignment there is considerable disagreement among biophysicists about the *energy* parameters for RNA folding. To analyze energy parameters and to study their accuracy, it would be useful to study parametric RNA folding. For example, comparison of regions corresponding to cloverleafs for tRNA parameter space decomposition would provide an estimate of accuracy of currently used RNA energy parameters.

Problem 49 *Develop an algorithm for parametric RNA folding.*

Longest common subsequence is the simplest variant of sequence alignment problem when insertions/deletions/mismatches penalties are zero ($\mu = \delta = 0$). Given two words $V = v[1], \ldots, v[n]$, $W = w[1], \ldots, w[n]$ we define a *common subsequence* of V and W as sequences of indices $1 \leq i_1 < \ldots < i_k \leq n$ and $1 \leq j_1 < \ldots < j_k \leq n$ such that

$$v[i_t] = w[j_t] \text{ for } 1 \leq t \leq k$$

We define k to be the length of the common subsequence, and we denote $l(V, W)$ the length of a *longest common subsequence* (LCS) of V and W. Let $\mathcal{V}$ and $\mathcal{W}$ be two sets of n-letter words over the same alphabet. Given $\mathcal{V}$, $\mathcal{W}$ and a probability measure p on $\mathcal{V} \times \mathcal{W}$ we are interested in the *average length* of LCS which is

$$l(n) = \sum_{V \in \mathcal{V}, W \in \mathcal{W}} l(V, W) \cdot p(V, W).$$

Two cases of LCS average length are of particular interest.

Longest increasing subsequence in random permutations $l_{per}(n)$: $\mathcal{V}$ contains only the word $(1, \ldots, n)$, $\mathcal{W}$ contains all permutations of length n and $p(V, W) = \frac{1}{n!}$.

The problem of finding $l_{per}(n)$ was raised by Ulam, 1961. Hammersley, 1972 proved that

$$\lim_{n \to \infty} \frac{l_{per}(n)}{\sqrt{n}} = c_{per}$$

where c_{per} is a constant. Even before the convergence has been proved, Baer and Brock, 1968 had conjectured that $c_{per} = 2$ on the basis of extensive computations. Logan and Shepp, 1977 and Vershik and Kerov, 1977 proved that $c_{per} \geq 2$. Later Vershik and Kerov proved that $c_{per} = 2$ (see Vershik and Kerov, 1985) and completed the solution of the Ulam problem.

Longest common subsequence $l_k(n)$ over k-letter alphabet: Both $\mathcal{V}$ and $\mathcal{W}$ contain all k^n n-letter words over the k-letter alphabet.

Chvatal and Sankoff, 1975 showed that

$$\lim_{n \to \infty} \frac{l_k(n)}{n} = c_k,$$

where c_k is a constant and gave lower and upper bounds for c_k. Later the bounds were improved by Deken, 1979, Chvatal and Sankoff, 1983, Deken, 1983 and Dancik and Patterson, 1994.

Problem 50 *Give tighter bounds for average length of LCS.*

Computing c_k seems to be a harder problem than computing c_{per} since the *Young tableaux* technique used by Logan and Shepp, 1977 and Vershik and Kerov, 1977 does not work for longest common subsequences. In the middle of 80's two conjectures about c_k were stated:

Problem 51 *(Sankoff-Mainville hypothesis, [SM83])* $\lim_{k \to \infty}(c_k \cdot \sqrt{k}) = 2$.

Problem 52 *(Arratia-Steele hypothesis, [S86])* $c_k = \frac{2}{1+\sqrt{k}}$.

Chvatal and Sankoff, 1975 were also interested in computing the variance of $l_k(n)$, $Var_k(n)$. They conjectured

Problem 53 $Var_k(n) = o(n^{2/3})$.

Steele, 1986 proved that $Var_k(n) \leq n$, it is still the sharpest known result.

Recently Chang and Lampe, 1992 suggested an analog of the Sankoff-Mainville hypothesis for the case of the *minimum edit distance* problem. Edit distance $d(V, W)$ between the words V and W is the minimum number of substitutions, deletions and insertions needed to transform V into W ($\mu = \delta = 1$). Chang and Lampe, 1992 conjectured that for random n-letter words over a k-letter alphabet

Problem 54

$$\lim_{n\to\infty} \frac{E[d(V,W)]}{n} = 1 - \frac{1}{\sqrt{k}} + o(\frac{1}{\sqrt{k}})$$

Below we assume that the strings $\mathbf{X} = X_1X_2\cdots X_n$ and $\mathbf{Y} = Y_1Y_2\cdots Y_n$ have *iid* letters. Let $S_n = S(\mathbf{X}, \mathbf{Y})$ be the *global* alignment score of the strings $\mathbf{X}$ and $\mathbf{Y}$ of length n (Waterman, 1989). The *local* alignment score is

$$H_n = \max\{S(I, J) : I(J) \text{ is a substring of } \mathbf{X}(\mathbf{Y})\}.$$

Sometimes parameter dependence is shown by $S_n = S_n(\mu, \delta)$ and $H_n = H_n(\mu, \delta)$.

Note that when $\mu = \delta = 0$, $S_n(0,0) = H_n(0,0) = l_k(n)$. Arratia and Waterman,1994 show that a " generalized" c_k

$$c = c(\mu, \delta) = \lim_{n\to\infty} \frac{S_n}{n}$$

exists with probability 1. Moreover $\{c = 0\} = \{(\mu, \delta) : c(\mu, \delta) = 0\}$ is a continuous curve separating $[0, \infty]^2$ into two components $\{c < 0\} = \{(\mu, \delta) : c(\mu, \delta) < 0\}$ and $\{c > 0\} = \{(\mu, \delta) : c(\mu, \delta) > 0\}$.

If $(\mu, \delta) \in \{c > 0\}$, Arratia and Waterman,1994 show that $\lim_{n\to\infty} \frac{H_n(\mu,\delta)}{n} = c(\mu, \delta)$. If $(\mu, \delta) \in \{c < 0\}$, they introduced a constant $b = b(\mu, \delta)$ such that

$$\lim_{n\to\infty} \mathbf{P}\{(1-\epsilon)b < \frac{H_n(\mu,\delta)}{\log(n)} < (2+\epsilon)b\} = 1.$$

Problem 55 *Show that for* $(\mu, \delta) \in \{c < 0\}$, $\lim_{n\to\infty} \frac{H_n(\mu,\delta)}{\log(n)} = 2b(\mu, \delta)$ *in probability.*

The problem of explicitly computing $c(\mu, \delta)$ for $(\mu, \delta) \in \{c > 0\}$ seems to be very difficult. In particular, computing $c(0, 0)$ would solve the Steele-Arratia conjecture. The corresponding problem for $\{c < 0\}$ is not quite so hard and special cases can be computed, $b(\infty, \infty) = 2$, for example.

Problem 56 *Compute* $b(\mu, \delta)$ *for* $(\mu, \delta) \in \{c < 0\}$.

An inverse problem is to find all (μ, δ) such that $c(\mu, \delta)$ is a constant.

Problem 57 *Characterize the curve* $c(\mu, \delta) =$ *constant, in particular* $c(\mu, \delta) = 0$.

6 Acknowledgments

Thanks to our many colleagues (too many in number to list here!), without whose papers, conversations and generous assistance this paper could not have been written.

References

[AW87] Aigner M, West D.B., Sorting by insertion of leading element, *Journal of Combinatorial Theory (Series A)* , **45** (1987) 306-309.

[A94] Alexander K. Shortest common superstrings for strings of random letters. *Proc. of 5th Ann. Symp. on Combinatorial Pattern Matching, Lecture Notes in Computer Science*, **807** (1994) 164-172

[AY88] Allison L., Yee,C.N. Restriction site mapping is in separation theory. *CABIOS*, **4** (1988) 97-101.

[AS94] Armen C., Stein C. A 2.75 approximation algorithm for the shortest superstring problem. *Proc. of 35th IEEE Symp. on Foundations of Computer Science*, (1994) (to appear)

[AW94] Arratia R., Waterman M.S. A phase transition for the score in matching random sequences allowing deletions. *Ann. Appl. Prob.*, **4** (1994) 200-225.

[BB68] Baer R., Brock P. (1968) Natural sorting over permutation spaces. *Math. Comp.*, **22** (1968) 385-410

[BP93] Bafna V., Pevzner P.A. Genome rearrangements and sorting by reversals. *Proc. of 34th IEEE Symp. on Foundations of Computer Science*, (1993) 148-157 (to appear in *SIAM J. on Computing*)

[BP95a] Bafna V., Pevzner P.A. Sorting by transpositions. *Proc. of 6th ACM-SIAM Symp. on Discrete Algorithms* (1995a) (to appear)

[BP95b] Bafna V., Pevzner P.A. Sorting by reversals: genome rearrangements in plant organelles and evolutionary history of X chromosome. *Mol. Biol. and Evol.*, **12** (1995b) (to appear)

[BS88] Bains W., Smith G.C. A novel method for DNA sequence determination. *J.Theor. Biol.*, **135** (1988) 303-307

[BJL91] Blum A., Jiang T., Li M., Tromp J., Yannakakis M. Linear approximation of shortest superstrings. *Proc. of 23rd ACM Ann. Symp. on Theory of Computing*, (1991) 328-336

[C83] Carson H. L. Chromosomal sequences and interisland colonization in Hawaiian Drosophila. *Genetics.* **103** (1983) 465–482.

[CL92] Chang W.I., Lampe J. Theoretical and empirical comparisons of approximate string matching algorithms. *Proc. of 3th Ann. Symp. on Combinatorial Pattern Matching, Lecture Notes in Computer Science*, **644** (1992) 172-181

[CS75] Chvatal V., Sankoff, D. Longest common subsequences of two random sequences. *Journal in Appl. Prob.*, **12** (1975) 306-315

[CS83] Chvatal V., Sankoff, D. An upper-bound techniques for lengths of common subsequences. In Sankoff, D., Kruskal, J.B., (eds.), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, (1983) 353-357

[C93] Copeland N. G., N. A. Jenkins, D. J. Gilbert, J. T. Eppig, L. J. Maltals, J. C. Miller, W. F. Dietrich, A. Weaver, S. E. Lincoln, R. G. Steen, L. D. Steen, J. H. Nadeau, E. S. Lander. A genetic linkage map of the mouse: Current applications and future prospects. *Science.* **262** (1993) 57-65.

[CGP94] Czumaj A., Gasieniec L., Piotrow M., Rytter W. Sequential and parallel approximation of shortest superstrings. *Proc of the 4th Scandinavian Workshop on Algorithms Theory SWAT'94*, (1994) 95-106

[DP94] Dancik V., Paterson M., Upper bounds for for the expected length of a longest common subsequence of two binary sequences. *Proc. of 11th Ann. Symp on Theoretical Aspects of Computer Science*, (1994) 669-678

[D79] Deken J. Some limit results for longest common subsequences. *Discrete Mathematics*, **26** (1979) 17-31

[D83] Deken J. Probabilistic behavior of longest common subsequence length. In Sankoff, D., Kruskal, J.B., (eds.), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, (1983) 359-362.

[DLB89] Drmanac R., Labat L., Brukner I., Crkvenjakov R. Sequencing of megabase plus DNA by hybridization. *Genomics*, **4** (1989) 114-128

[EG81] Even S., Goldreich, O., The minimum-length generator sequence problem is NP-hard, *Journal of Algorithms*, **2** (1981) 311-313

[FKW93] Farach M., Kannan S., Warnow T. A robust model for finding optimal evolutionary trees *Proc. of 25th Ann. ACM Symp. on the Theory of Computing*, (1993) 137-145

[FP94] Feldman W., Pevzner P.A. Gray code masks for DNA Sequencing by Hybridization *Genomics*, **23** (1994) 233-235

[FS83] Fitch W., Smith T. Optimal sequence alignments. *Proc. Natl. Acad. Sci. USA* **80** (1983) 1382-1386.

[FSR83] Fitch W.M., Smith,T.F., Ralph,W.W. Mapping the order of DNA restriction fragments, *Gene*, **22** (1983) 19-29.

[FRP91] Fodor S.P.A., Read J.L., Pirrung M.S., Stryer L., Lu A.T., Solas D. Light-directed spatially addressable parallel chemical synthesis. *Science*, **251** (1991) 767-773

[FH89] Fauron C., Havlik C. The maize mitochondrial genome of the normal type and the cytoplasmic male sterile type T have very different organization. *Current Genetics.* **15** (1989) 149-154.

[F85] Franklin N. Conservation of genome form but not sequence in the transcription antitermination determinants of bacteriophages $\lambda, \phi 21$ and $P22$. *J. of Mol. Evol.*, **181** (1985) 75-84

[GG89] Galil Z., Giancarlo R. Speeding up dynamic programming with applications to molecular biology. *Theor. Comput. Sci.*, **64** (1989) 107-118

[GMS80] Gallant J., Maier D., Storer J. On finding minimal length superstrings. *J. of Comp. and Syst. Sci.*, **20** (1980) 50-58

[GP79] Gates W.H., Papadimitriou C.H., Bounds for sorting by prefix reversals, *Discrete Mathematics*, **27** (1979) 47-57.

[GW87] Goldstein L., Waterman M.S. Mapping DNA by stochastic relaxation. *Advances in Appl. Math.*, **8** (1987) 194-207.

[GM90] Grigorjev A. V., Mironov, A. A. Mapping DNA by stochastic relaxation: a new approach to fragment sizes *CABIOS*, **6** (1990) 107-111

[GBN92] Gusfield D., K. Balasubramanian, D. Naor Parametric optimization of sequence alignment. *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, (1992) 432-439

(full version appeared in *Algorithmica*, **12** (1994) 312-326)

[GT78] Gyori E., Turan, E., Stack of pancakes, *Studia Sci. Math. Hungar.*, **13** (1978) 133-137.

[H72] Hammersley J.M. (1972) A few seedlings of research. *Proc Sixth Berkeley Symp. Math. Statist. Probability*, **1** (1972) 345-394,

[HCK94] Hannenhalli S., C. Chappey, E. Koonin, P. Pevzner. Scenarios for genome rearrangements: Herpesvirus evolution as a test case. *Proc. of 3rd Intl. Conference on Bioinformatics and Complex Genome Analysis* (1994) (to appear)

[HBB92] Hoffmann R. J., J. L. Boore, W. M. Brown (1992) A novel mitochondrial genome organization for the blue mussel, *Mytilus edulis. Genetics* **131** (1992) 397–412.

[HP94] Hoot S. B., J. D. Palmer. Structural rearrangements including parallel inversions within the chloroplast genome of anemone and related genera. *J. of Mol. Evol.*, **38** (1994) 274-281.

[HS93] Heidari M.H., Sudborough I.H. On sorting by prefix reversals and the diameter of pancake networks. (1993) (*manuscript*)

[HAY90] Ho S.T.S., Allison, L. Yee, C.N. Restriction site mapping for three or more enzymes. *CABIOS*, **6** (1990) 195-204.

[J85] Jerrum M. R., The complexity of finding minimum-length generator sequences, *Theoretical Computer Science*, **36** (1985) 265-289.

[JL93] Jiang T., Li M., Approximating shortest superstrings with constraints, *Proc. 3rd Workshop on Algorithms and Data Structures*, (1993) 385-396 (to appear in *Theoretical Computer Science*).

[JL94] Jiang T., Li M., DNA sequencing and string learning, *Mathematical System Theory*, (1994) (to appear)

[K93] Karp R.M. Mapping the Genome: some combinatorial problems arising in molecular biology. *Proc. of 25th Ann. ACM Symp. on the Theory of Computing*, (1993) 278-285

[KS93] Kececioglu J., Sankoff D. Exact and approximation algorithms for the inversion distance between two chromosomes. *Proc. of 4th Ann. Symp. on Combinatorial Pattern Matching, Lecture Notes in Computer Science.* **684**, (1993) 87-105 (to appear in *Algorithmica* as Exact and approximation algorithms for sorting by reversals, with application to genome rearrangements).

[KS94] Kececioglu J., Sankoff D. Efficient bounds for oriented chromosome inversion distance. *Proc. of 5th Ann. Symp. on Combinatorial Pattern Matching, Lecture Notes in Computer Science*, **807** (1994) 307-325.

[KR95] Kececioglu J., Ravi R. Of mice and men: evolutionary distances between genomes under translocations. *Proc. of 6th ACM-SIAM Symp. on Discrete Algorithms* (1995) (to appear)

[KLK89] Khrapko K.R.,Lysov Yu.P.,Khorlin A.A.,Shik V.V.,Florent'ev V.L.,Mirzabekov A.D. An oligonucleotide approach to DNA sequencing. *FEBS Letters*, **256** (1989) 118-122

[KDP93] Knox E. B., Downie S.R., Palmer J.D., Chloroplast genome rearrangements and evolution of giant lobelias from herbaceous ancestors, *Mol. Biol. Evol.*, **10** (1993) 414-430

[K93] Koonin E. V., Dolja, V. V., Evolution and Taxonomy of Positive-strand RNA viruses: Implications of comparative analysis of amino acid sequences, *Crit. Rev. Biochem. Molec. Biol.*, **28** (1993) 375-430

[KPS94] Kosaraju R., Park J., Stein C. Long tours and short superstrings. *Proc. of 35th IEEE Symp. on Foundations of Computer Science*, (1994) (to appear)

[K88] Krawczak M. Algorithms for the restriction site mapping of DNA molecules. *Proc. Natl. Acad. Sci. USA*, **85** (1988) 7298-7301.

[LS77] Logan B.F., Shepp L.A. (1977) A variational problem for random Young tableaux. *Adv. in Math.*, **26** (1977) 206-222

[L88] Lyon M. F. X-Chromosome Inactivation and the Location and Expression of X-linked Genes. *Am. J. Hum. Genet.*, **42** (1988) 8-16.

[LFK88] Lysov Yu.P., Florent'ev V.L., Khorlin A.A., Khrapko., Shik V.V., Mirzabekov A.D. DNA Sequencing by hybridization with oligonucleotides. A novel method. *Dokl. Acad. Sci USSR*, **303** (1988) 1508-1511

[M94] Martin D.R. Equivalence classes for the Double-Digest problem with coincident cut sites. *J. Comp. Biol.*, **1** (1994) 241-253

[MM88] Miller W., Myers E. Sequence comparison with concave weighting functions, *Bull.Math.Biol.*, **50** (1988) 97-120

[NT84] Nadeau J. H., Taylor B.A. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proc. Natl. Acad. Sci. USA.* **81** (1984) 814–818.

[NN93] Newberg L., Naor D. A lower bound on the number of solutions to the Probed Partial Digest Problem. *Adv. in Appl. Math.*, **13** (1993) 172-183

[PH88] Palmer J. D., Herbon L.A. (1988) Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *J. of Mol. Evol.*, **27** (1988) 87-97

[P82] Pearson W. Automatic construction of restriction site maps. *Nucleic Acids Res.*, **10** (1982) 217-227.

[P89] Pevzner P.A. l-tuple DNA sequencing: a computer analysis. *J.Biom.Struct. and Dyn.*, **7** (1989) 63-73

[P90] Pevzner P.A. DNA physical mapping. In M.D.Frank-Kamenetzky (ed.) *Computer analysis of genetic texts*, (1990) 154-188 (in Russian)

[P92] Pevzner P.A. DNA physical mapping, flows in networks and minimum cycles mean in graphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **8** (1992) 99-112

[P94] Pevzner P.A. DNA physical mapping and alternating Eulerian cycles in colored graphs. *Algorithmica*. (1994) (to appear)

[PM87] Pevzner P.A., Mironov A.A. An efficient method for physical mapping of DNA molecules. *Molek.Biol.*, **21** (1987) 788-796

[PLK91] Pevzner P.A., Lysov Yu.P., Khrapko K.R., Belyavsky A.V., Florentiev V.L., Mirzabekov A.D. Optimal chips for megabase DNA sequencing. *J.Biomol.Struct. Dyn.*, **9** (1991) 399-410

[PL94] Pevzner P.A., Lipshutz R. Towards DNA Sequencing Chips *19th International Conference on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science*, **841** (1994) 143-158

[RJ92] Raubenson L.A., Jansen R.K., Chloroplast DNA evidence on the ancient evolutionary split in vascular land plants, *Science*, **255** (1992) 1697-1699

[RS82] Rosenblat J., Seymour P.D. The structure of homometric sets. *SIAM J. Alg. Discr. Meth.*, **3** (1982) 343-350

[SCA90] Sankoff D., Cedergren R, Abel Y. Genomic divergence through gene rearrangement. In R. F. Doolittle (ed.) Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences. *Methods in Enzymology*, **183** (1990) 428-438

[S92] Sankoff D., Leduc G., Antoine N., Paquin B., Lang B.F., Cedergren, R., Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome, *Proc. Natl. Acad. Sci. USA*, **89** (1992) 6575-6579

[SM83] Sankoff D., Mainville S. Common subsequences and monotone subsequences. In Sankoff, D., Kruskal, J.B., (eds.), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, (1983) 363-365.

[SW91] Schmitt W., Waterman M. Multiple solutions of DNA restriction mapping problem. *Adv. in Appl. Math.*, **12** (1991) 412-427

[S90] Skiena S.S., Smith W.D., Lemke P. Reconstructing sets from interpoint distances. *Proc. of 6th Annual Symposium on Computational Geometry, Berkeley, CA*, (1990) 332-339

[SW81] Smith T.F., Waterman M.S. Identification of common molecular subsequences. *J. Mol. Biol.*, **147** (1981) 195-197

[S88] Southern E. *United Kingdom Patent Application GB8810400.* (1988)

[S86] Steele J.M. An Efron-Stein inequality for nonsymmetric statistics. *Ann. of Statistics*, **14** (1986) 753-758

[TU88] Tarhio J., Ukkonen E. A Greedy approximation algorithm for constructing shortest common superstrings. *Theoretical Computer Science*, **57** (1988) 131-145

[TY93] Teng S.H., Yao F. Approximating shortest superstrings. *Proc. of 34th IEEE Symp. on Foundations of Computer Science*, (1993) 158-165

[T89] Turner J. Approximation algorithms for the shortest common superstring problem. *Information and Computation*, **83** (1989) 1-20

[U61] Ulam S. Monte-Carlo calculations in problems of mathematical physics. In *Modern mathematics for the engineer*, McGraw-Hill, New York, (1961) 261-281

[VW94] Vingron M., Waterman M.S. Sequence alignment and penalty choices: review of concepts, case studies and implications. *J. Mol. Biol.*, **235** (1994) 1-12.

[VK77] Vershik A.M., Kerov S.V. Asymptotics of the Plancherel measure of the symmetric group and the limiting form of Young tableaux. *Soviet Math. Dokl.*, **18** (1977) 527-531

[VK85] Vershik A.M., Kerov S.V. Asymptotics of the maximal and typical dimension of irreducible representations of symmetric groups. *Functional. Anal. Appl.*, **19** (1985) 21-31

[W89] Waterman M.S. (ed.). *Mathematical Methods for DNA Sequences*, (1989) CRC Press, Inc., Baco Raton, Florida.

[W95] Waterman M. S. *Introduction to Computational Biology: Maps, Sequences and Genomes* Chapman Hall, 1995

[WG86] Waterman M.S., Griggs,J.R. Interval graphs and maps of DNA. *Bull. Math. Biol.*, **48** (1986) 189-195.

[WGA87] Waterman M.S., Gordon, L., Arratia, R. Phase transitions in sequence matches and nucleic acid structure. *Proc. Natl. Acad. Sci. USA*, **84** (1987) 1239-1243.

[WEL92] Waterman M. S., Eggert M. , Lander E. Parametric sequence comparisons. *Proc. Natl. Acad. Sci. USA*, **89** (1992) 6090-6093.

[WEH82] Watterson G.A., Ewens, W.J., Hall, T.E. and Morgan, A. The chromosome inversion problem, *Journal of Theoretical Biology*, **99** (1982) 1-7.

[We92] Weiss M. *Data Structures and Algorithm Analysis.* (1992) The Benjamin/Cummings Publishing Company.

[WPF89] Whiting J., Pliley,M., Farmer, J., Jeffery, D., In situ hybridization analysis of chromosomal homologies in Drosophila melanogaster and Drosophila virilis, *Genetics*, **122** (1989) 99-109

[Y91] Yap R.H.C. Restriction site mapping in CLP($\mathcal{R}$). *Proceedings of 8th International Conference on Logic Programming* (1991) 521-534.

[Z94] Zhang Z. An exponential example for a Partial Digest Mapping algorithm. *J. Comp. Biol.*, **1** (1994) 235-240