Pergamon

0097-8485(94)00033-6

Computers Chem. Vol. 18, No. 3, pp. 207–220, 1994

0097-8485/94 $7.00 + 0.00

THE CHIMERIC MAPPING PROBLEM: ALGORITHMIC STRATEGIES AND PERFORMANCE EVALUATION ON SYNTHETIC GENOMIC DATA*

DAVID GREENBERG and SORIN ISTRAIL

Sandia National Laboratories, Algorithms and Discrete Mathematics Department, Albuquerque, NM 87185-1110, U.S.A.

(Received 21 December 1993; in revised form 18 May 1994)

Abstract—The Human Genome Project requires better software for the creation of physical maps of chromosomes. Current mapping techniques involve breaking large segments of DNA into smaller, more-manageable pieces, gathering information on all the small pieces, and then constructing a map of the original large piece from the information about the small pieces. Unfortunately, in the process of breaking up the DNA some information is lost and noise of various types is introduced; in particular, the order of the pieces is not preserved. Thus, the map maker must solve a combinatorial problem in order to reconstruct the map. Good software is indispensable for quick, accurate reconstruction.

The reconstruction is complicated by various experimental errors. A major source of difficulty—which sems to be inherent to the recombination technology—is the presence of *chimeric* DNA clones. It is fairly common for two disjoint DNA pieces to form a chimera, i.e. a fusion of two pieces which appears as a single piece. Attempts to order chimera will fail unless they are algorithmically divided into their constituent pieces. Despite consensus within the genomic mapping community of the critical importance of correcting chimerism, algorithms for solving the chimeric clone problem have received only passing attention in the literature. Based on a model proposed by Lander (1992a, b) this paper presents the first algorithms for analyzing chimerism.

We construct physical maps in the presence of chimerism by creating optimization functions which have minimizations which correlate with map quality. Despite the fact that these optimization functions are invariably NP-complete our algorithms are guaranteed to produce solutions which are close to the optimum. The practical import of using these algorithms depends on the strength of the correlation of the function to the map quality as well as on the accuracy of the approximations. We employ two fundamentally different optimization functions as a means of avoiding biases likely to decorrelate the solutions from the desired map.

Experiments on simulated data show that both our algorithm which minimizes the number of chimeric fragments in a solution and our algorithm which minimizes the maximum number of fragments per clone in a solution do, in fact, correlate to high quality solutions. Furthermore, tests on simulated data using parameters set to mimic real experiments show that the algorithms have the potential to find high quality solutions with real data. We plan to test our software against real data from the Whitehead Institute and from Los Alamos Genomic Research Center in the near future.

1. INTRODUCTION

Computational support is vital for the creation of robust genomic maps. There is a nearly seven orders of magnitude gap between the size of biological molecules in real organisms and the size of molecules which can be examined in detail. In order to bridge this gap physical maps are constructed which hierarchically divide the genome into fragments of successively smaller sizes. The construction of the physical maps requires algorithms to order the fragments at each stage of the hierarchy.

More specifically, for our purposes a piece of DNA can be considered to be a string over the four character alphabet $\{A, C, G, T\}$. The human genome (i.e. one copy of the DNA in a human cell) is approx. 3×10^9 base pairs long. In the laboratory it is possible to read reliably the sequence of characters from segments containing at most 500 base pairs. Obviously the only viable approach is to break the long segments of DNA into smaller sections and analyze the smaller sections individually. In practice this is done in several stages. At each stage the DNA is broken into a number of pieces called *clones*. Through various biological techniques information is gathered about which clones overlap. A *physical map* is the reconstruction of the order in which the clones appeared in the original segment. The requirement that the reconstructed solution map must match an unknown template, i.e. the *true map*, makes the problem difficult to cast in computational terms. Since the quality of a map is defined only in terms of

*Some topics included in this work have been discussed during the *Third International Workshop on Open Problems of Computational Molecular Biology*, Telluride, Colo., 11–25 July 1993.

its closeness to an *unknown* true map there is no immediate means of comparing potential solution maps. Fortunately, although the true map is not known, some of its properties are. Intuitively, the maps which are close to the true map possess more "structure" than random arrangements. Therefore, one tries to discover natural parameters of the maps that are correlated with the structure of true maps. Once such parameters have been found the problem of finding the true map can be modeled by the combinatorial problem of finding solutions which minimize the deviation of the parameter from the expected value of the true map.

Schmitt and Waterman have summarized the difficulties of this type of approach and propose a potential improvement.

> A molecular biologist wishes to find *the* correct map, the map consistent with the unknown DNA sequence. Therefore a map that is "close" to optimal as measured by some arbitrary objective function might be very far from acceptable to a biologist. Mapping algorithms should produce the smallest possible set of maps that reliably include the biologically correct map (Schmidtt and Waterman, 1991).

Their important observation is that even if *the* optimum of some optimization function could be found it is unlikely to be the true solution. Instead one should look for a set of good solutions according to the optimization function with the hope that one such solution is the true solution and that all such solutions will share many features with the true solution. Even once an optimization function has been chosen several difficulties remain. In practice it is invariably the case that any objective function which correlates with good maps, on arbitrary hybridization matrices, provides us with an NP-complete optimization problem (Garey & Johnson, 1979). This means that finding an efficient algorithm which always finds the optimal solution is unlikely to be possible. Instead we must settle for an algorithm which always finds solutions which are close to optimal and/or one which finds the optimal solution most of the time. Given that our algorithm will at best give us a set of solutions which are close to optimal for our optimization function one might ask what, if anything, we have learned about the true map. *A priori* the answer is "not much". We need one more property for our solutions—*maps which are close to optimal under the optimization function must share many features with the true map.*

All of the preceding is true of the physical mapping problem in general. As a first step we apply it to the specific question of physical mapping via STSs or single-copy landmarks when there are chimeric clones (see Section 2 for definitions). In Section 3 we show how to convert the chimeric clone problem into an optimization function and in Section 4 and 5 define a set of optimization functions based on the *chimeric pattern* of a proposed map. Two simple members of this class of functions are χ, the maximum number of chimeric fragments in a single clone, and σ, the total number of chimeric fragments in all clones. Since chimerism is a result of experimental error (the perfect experiment would yield no chimeric clones) it is reasonable to assume that the map corresponding to the actual order of the experimental fragments along the DNA being mapped has a low amount of chimerism. Thus minimizing functions such as χ and σ (which are monotonic in the amount of chimerism) should yield solutions which are similar to the true map.

In Sections 6 and 7 we present efficient algorithms for χ and σ which are guaranteed to produce close to optimal solutions despite the fact that both optimization problems are NP-complete. In Section 8 and 9 we describe experiments which demonstrate that minimizing either χ or σ does indeed result in good maps. On simulated data designed to mimic current experiments the algorithms correctly reconstructed the true map over 80% of the time and was incorrect by only a small number of transpositions in the remaining cases.

2. THE BIOLOGY BEHIND PHYSICAL MAPPING

Several groups have detailed a program for the creation of physical maps from single-copy landmarks (Lander & Waterman, 1988; Barillot *et al.*, 1991). Details of the biological techniques involved as well as reports on subsequent uses of the procedure can be found in (Brown, 1990; Cohen *et al.*, 1993; Craig *et al.*, 1990; Green *et al.*, 1991; Nelson & Brownstein, 1993; Olson *et al.*, 1989; Torney, 1991). We give here only the brief overview necessary for the definition of terms.

The necessity of creating physical maps is a result of the large gap between the size of a piece of DNA which can be transcribed (about 500 bases) and the size of a chromosome (order 10^8 bases). The general technique for bridging this gap is to break copies of a chromosome up into many pieces called *clones*. The production of the clones does not preserve information about their relative position on the chromosome. Instead, various techniques are used to identify which clones overlap.

We will primarily be concerned with the use of single-copy landmarks and in particular STSs. An SCL represents a unique point somewhere in the genome. By applying the SCL as a *probe* to a clone it can be determined if the clone covers the point. Two clones which hybridize to the same probe are then known to overlap. An STS is a particular type of SCL which has the advantage of being defined by its sequence and not just by a biological sample.

A map is created by finding an ordering of the probes and their incident clones which conforms to the overlap information provided by the probes. Although the order of the clones is the ultimate goal of the physical mapping process we will define a map by the order of the probes since such an order is not

complicated by chimerism. An ordering of non-chimeric clones is then easily constructed from the order of the probes.

2.1. *Experimental errors—chimerism*

Ideally we would start with a target piece of DNA, break it up into clones, construct a set of probes, and determine exactly which probes stuck to each clone. We would also expect that the clones cover the entire original piece, that each probe has a unique occurrence in the target DNA peice, and that each overlap of clones is witnessed by at least one probe. Unfortunately molecular biology, like all experimental sciences, does not produce perfect data. The determination of which probes stick to which clones will yield both false positives and false negatives. Some sections of the target DNA will tend to shatter into tiny pieces which are lost while other sections will contain no probe sequences. Thus we can only hope to come close to reconstructing a complete, unambiguous map.

One important type of experimental error, *chimerism*, results from the cloning process itself. Clones, as the name suggests, are created by inserting fragments of DNA into vectors which are then replicated as clones. The process of inserting the fragments into vectors sometimes results in two or more fragments being contained in one clone. These clones are referred to as *chimera* since they contain unrelated pieces.

Multiple-insert chimera complicate the physical mapping process since a chimeric clone cannot be mapped to a single contiguous section on the genome. Deletions can also produce clones which do not contain a single contiguous section of DNA. Sometimes the vector will remove sections of the insert fragment during replication. Unlike the multiple-insert chimera the resulting clone does not contain completely unrelated pieces but the discontinunity of the pieces creates similar problems for the mapping process.

In this paper we will treat any clone containing two or more non-contiguous pieces of DNA as a chimeric clone.

Despite the many methods developed for chimerism detection the percentage of chimerism in genomic libraries continues to be high. When the Yeast Artificial Chromosome, YAC, technology was developed in 1987, the inventors predicted that the technology would suffer from some chimerism, with an estimate of about 10% chimerism in the clones of a YAC library. In the clone libraries that were used in the creation of the first high-resolution maps (Vollrath *et al.*, 1992; Chumakov *et al.*, 1992; Foote *et al.*, 1992) the chimerism was discovered, however, to occur at much higher percentages—reaching 40% in the chromosome 21 map and about 59% in the Y chromosome map. Also, the frequency of chimerism has been estimated at 40–60% for the two most widely used human YAC libraries (Green *et al.*, 1991; Nelson & Brownstein, 1993). The recent advance from YACs to megaYACs seems to confirm the expectation that using larger YACs means more chimerism is introduced in the library.

Although we will concentrate on resolving the problems due to chimeric clones there are several other types of errors common in physical mapping. We expect that the techniques described here will also be applicable to resolving other errors such as false positive and negative hybridizations and non-uniqueness of probes.

3. CONVERTING TO A COMBINATORIAL PROBLEM

We have seen that in the process of making a physical map that the biologist starts with a target piece of DNA (call it $\mathscr{D}$) and then creates clones of fragments of $\mathscr{D}$. Experiments are then done to determine which probes stick to which clones. The result is the hybridization data for $\mathscr{D}$. Note that the clones and probes are typically chosen so that each probe hybridizes to a small number of clones. The fact that only a very small percentage of possible clone-probe hybridizations are positive is critical to the understanding of physical mapping. In this section we give terminology and definitions necessary to convert the chimeric mapping problem into a combinatorial problem. We start by defining a hybridization matrix which encodes all the information about the biological experiments to be used by our algorithms. The sparseness parameter allows us to concentrate only on the biologically relevant matrices—those in which probes hybridize to only a few clones.

Definition 1. *We denote the set of clones as* $C = \{C_1, \ldots, C_m\}$, *the set of probes as* $P = \{P_1, \ldots, P_n\}$.

The clone/probe hybridization matrix is an $m \times n$, *0–1 matrix,* A, *such* $A[i, j] = 1$ *exactly when probe* P_j *hybridizes to clone* C_i.

Let s_j *be the number of entries of column* j *equal to 1. Define the spareness of* A $s(A) = \max\{s_j \mid 1 \leqslant j \leqslant n\}$.

Since we are concentrating on the question of chimerism the entries of A are defined to be 0 or 1; that is, we are assuming that the hybridization experiments are unambiguous and fault-free. Furthermore, the use of STS probes corresponds to the assumption that the probes in P each occur exactly once along $\mathscr{D}$, but not necessarily in the order 1 through n. The construction of a physical map thus corresponds to retrieving the correct order of the probes.

Definition 2. *Let* $PO(A)$ *be the set of all possible probe orders, i.e. the set of permutations of* $\{1, 2, \ldots, n\}$.

For any $\pi \in PO(A)$ *the map induced by* π *(denoted* A^{π}*) is the matrix obtained by permuting the columns of* A *according to* π.

A map, with its explicit ordering of the probes, results in an implicit arrangement of the clones. In the permuted matrix, A^{π} the 1's in each row of the matrix then tell where the map assigns the putative

fragments of that clone. In order to quantify how much clone splitting was required for a proposed map we make the following definitions.

Let $\alpha = (r_1, r_2, \ldots, r_n)$ be a vector with 0–1 entries. The sub-vector $r_l, r_{l+1}, \ldots, r_{l+l}, r_{l+l+1}$ is called a *block of consecutive ones* iff $r_{l-1} = 0$, $r_l = \cdots = r_{l+l} = 1$, and $r_{l+l+1} = 0$ (let $r_0 = r_{n+1} = 0$). For example, the vector $\alpha_1 = (1, 1, 1, 0, 1, 0, 0, 0, 1, 1)$ has three blocks of consecutive ones while the vector $\alpha_2 = (0, 0, 0, 1, 1, 1, 1, 0, 0, 0)$ has one block of consecutive ones. If α_1 and α_2 were the rows of a map A^π corresponding to clones C_1 and C_2 then in this map C_1 is a chimeric clone while C_2 is non-chimeric. (That is clone C_1 appears as three fragments which hybridized three, one, and two probes respectively and C_2 appears as a single fragment which hybridized four probes.) In a map matrix a row corresponding to a non-chimeric clone will have exactly one block of consecutive ones, while a row corresponding to a chimeric clone will have at least two blocks of consecutive ones.

Definition 3. *For a map A^π define its chimeric pattern as the vector $c(A^\pi) = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, where α_i is the number of blocks of consecutive ones occurring in row i, $1 \leqslant i \leqslant m$. A map, A^π, has the consecutive ones property (C1P for short) if it has chimeric pattern $= \{1, 1, \ldots, 1\}$. In this case, π is called a C1P-probe order for A and the matrix A^π is said to be in the C1P form.*

4. LIMITATIONS OF THE COMBINATORIAL APPROACH

We can now state the physical mapping problem as combinatoric one—given a probe-clone incidence matrix A find the probe ordering π_0 corresponding to the ordering on $\mathscr{D}$. Unfortunately this problem is clearly under-defined. Without knowing anything about $\mathscr{D}$ and A *any* probe ordering could be the correct order; each ordering corresponds to a chimeric pattern which conforms to the probe-clone incidences.

4.1. Using a priori knowledge: C1P

Suppose, however, that we knew that A corresponded to a chimera-free experiment. That is, suppose that we knew that each clone in C corresponded to a single fragment of $\mathscr{D}$. Then we would know that A^{π_0} must have the C1P form. Now we can look for those probe orders which correspond to C1P matrices. The order π_0 is guaranteed to be among these. If we can find these orders and if there are not too many of them we have solved the physical map problem in the non-chimeric case.

Booth & Lueker (1976) developed an algorithm which creates a compact representation of *all* permutations of the columns which result in a C1P matrix. There can be exponentially many such permutations so a compact representation is crucial. Their representation has an additional feature important for physical maps: it highlights portions of the permutation which are critical for the matrix to be C1P. For example, if probes 20, 55, 1, 5 and 112 always occur consecutively in all C1P orders then it is simple to retrieve this fact from the representation. Since the true order, π_0, is C1P, any property which is true of all C1P orderings is true of π_0. Thus even when the C1P algorithm fails to find a small set of orderings it can still provide powerful information about the true ordering.

Booth and Lueker's algorithm does not solve the physical mapping problem for two reasons. The first is that, as we have seen, real incidence matrices will contain chimeric clones and therefore will not be C1P. The second reason is that the incidence matrix often does not contain enough information to uniquely determine the true probe ordering.

4.2. Lack of information in the matrix

Even with the strong information that the matrix is C1P it is almost never the case that π_0 is the only C1P arrangement of the matrix. The problem is that the matrix A may not contain enough information to uniquely determine π_0. Suppose that two probes are recorded in A as hybridizing to exactly the same set of clones. Consequently, their particular order within any C1P probe order is not determined by the information reported by A. Although π_0 specifies precisely the order of these two probes, this information is not enforced by A. It is beyond the possibilities of any mapping algorithm to retrieve this information. As, in general, the information content of the hybridization matrix is less than the information content of the true map, the well-defined goal of the mapping should be to compute a probe order π that is equivalent to π_0 when restricted to the information contained by A. In the C1P case Booth and Lueker's algorithm does exactly this.

When A does not contain enough information to uniquely determine π_0 there is some potential recourse. The experiment used to create A can be augmented by including more clones. The fact that more clones create more *coverage* and therefore better maps is well known. In the non-chimeric case the reason for desiring more coverage is simply to ensure that A has enough information to narrow the possible orderings down to one or at most a few possibilities. The map maker can decide when more time should be spent making a larger A and when the time can be better used to perform specific experiments designed to disambiguate among the orderings consistent with A. Since additional types of errors besides chimerism are likely to limit the information content of A the latter approach may often be more effective.

4.3. Approximating a map

In order to create maps from data containing chimeric clones it would be nice to have some sort of *a priori* knowledge similar to knowing that the matrix is C1P in the non-chimeric case.

Before suggesting such properties let's look at what made C1P a good property in the non-chimeric case.

Ideally only the matrix permutation corresponding to the true order would have the C1P form. Even in the non-chimeric case this was not always possible because the matrix A does not always contain full information. Instead the restriction to C1P matrices yielded a set a maps which contained the true order and in some sense had small size. In the case of C1P a small set of maps did not mean a small cardinality—the number of C1P arrangements of a matrix can be exponential in the size of the matrix. Instead small meant that all the maps in the set were close together in the sense that they shared information about the true order. Using Booth and Lueker's algorithm it is in fact possible to retrieve all the information about the true order contained in A. Furthermore the algorithm runs very quickly.

We can summarize the advantages of knowing that the true matrix is C1P by the following three properties.

1. There exists a set of probe orderings, $Maps(A)$, (in this case the C1P orderings) such that each ordering in $Maps(A)$ contains all the ordering information available from A.
2. A fast algorithm exists that will produce a probe ordering π from $Maps(A)$.
3. The true probe ordering π_0 is $Maps(A)$.

Since the chimeric case is more complicated than the non-chimeric case we will have to relax these conditions. We will choose properties which are representative of typical, experimental, incidence matrices. However, it is possible that a given experiment yields an incidence matrix which is quite atypical. Thus rather than demanding that the true probe ordering is in $Maps(A)$ we must instead ask that a good approximation of the true ordering is in $Maps(A)$. Similarly it is unlikely that we can retrieve all the information in A. Instead we ask that each member of $Maps(A)$ represent a large fraction of the information in A. Lastly, it will turn out that our properties will correspond to NP-complete optimization functions. Thus if $Maps(A)$ is defined as the orders which have optimal value for the optimization function it will not be possible to find efficient algorithms. Instead we look for algorithms which find solutions which are close to optimal and are efficient.

The three basic properties of $Maps(A)$ become:

1. There exists a set of probe orderings, $Maps(A)$, such that every ordering in $Maps(A)$ contains a *large fraction* of the ordering information available from A.
2. A fast *approximation* algorithm exists that will produce a probe ordering π from $Maps(A)$.
3. Good *approximations* of the true probe ordering π_0 are in $Maps(A)$.

Although the notions of approximation and large fraction are not well defined the intent should be clear. We wish to come as close as possible to the C1P case. An assessment of whether we come close enough will in the end depend on how well the algorithms work on real data. Since we have not yet been able to run our algorithms on real data we substitute some intuitive rationales and the results of runs on simulated data.

5. CHIMERIC OPTIMIZATION FUNCTIONS

The presence of chimeric clones excludes assuming that A^{π_0} is C1P. However, the chimerism introduced by experiment is not arbitrary. We know that the percentage of chimeric clones is likely to be at most 50% and that the number of chimeric fragments in a clone will rarely be greater than two. Therefore we need to look for properties similar to C1P which capture this knowledge of about correct orderings.

Rather than choose a single property we define a class of optimization functions. Matrices with optimum value of these functions have in some sense low chimerism. We then look in detail at two members of this class. One reason for choosing a class of functions rather than a single one is that any single function is unlikely to correlate perfectly with the true ordering. By having several functions we hope to avoid missing available information about the true ordering and to give greater confidence to information we report.

The purpose of having an optimization function is to help us find probe orderings which are similar to the true ordering. Since actual experimental incidence matrices will have restricted chimeric patterns† we want the optimization function to favor probe orderings with small numbers of fragments per clone and small numbers of chimeric clones. Thus, for an experiment with n probes and m clones, we choose a function, f, which maps the chimeric pattern (an n-vectors over N, where N is the set of non-negative integers) to an optimization value (typical also in N).

The following three restrictions on f tie the value $f(c(\pi))$ to the chimerism in π. The first enforces that f correlate exactly in the C1P case; the optimum value for f is a non-chimeric ordering. The second requirement ensures that when some chimerism is inherent in the matrix that f will never favor a solution which has more chimerism. The last property ensures that the effectiveness of f grows with the amount of information in the incidence matrix.

1. Optimizing f should yield a C1P ordering if a C1P ordering exists.
2. The function f should be monotone in the amount of chimerism. That is, for two chimeric patterns α_1 and α_2, $\alpha_1 \leqslant \alpha_2 \mapsto f(\alpha_1) \leqslant f(\alpha_2)$. (One vector is $\leqslant$ another vector if each component of the first is $\leqslant$ to the corresponding component of the second.)
3. Adding more information to the matrix (e.g. more clones) should not increase the number of optimal solutions to f.

† See Section 3 for definition.

5.1. The functions σ and χ

Although there are many functions which fit these requirements, two functions seemed especially natural to us and to biologists with whom we have talked. We denote them as σ = the total number of DNA inserts, and χ = the maximum number of DNA inserts per clone. It is possible that a more complicated function such as a linear combination of σ and χ will have even better correlation to true probe orders but these two functions have the advantage of being simple and of having provably good approximation algorithms. Formally we define them as follows:

Definition 4. *Let A be an $m \times n$ hybridization matrix, π a probe order in $PO(A)$, A^π the corresponding map, and $c(A^\pi) = (\alpha_1, \ldots, \alpha_m)$ the chimeric pattern of A^π. Define:*

1. $\sigma: N^n \to N$, *given by* $\sigma(\alpha_1, \ldots, \alpha_m) = \sum_{i=1}^{m} \alpha_i$, *and*
2. $\chi: N^n \to N$, *given by* $\chi(\alpha_1, \ldots, \alpha_m) = max\{\alpha_1, \ldots, \alpha_m\}$

The σ-value (χ-value) of the map A^π is given by $\sigma(c(A^\pi))$ (respectively, $\chi(c(A^\pi))$).

Having chosen an optimization function the conversion of the biological data to an optimization problem is straighforward. We can formally define *The chimeric f-Optimization Problem* as follows:

Given: an $m \times n$ incidence matrix A and a monotone function $f: N^n \to N$.
Find: a probe ordering $\pi \in PO(A)$ such that the f-value of the map A^π is minimal.

A probe ordering that is a solution of the f-optimization problem is called an *f-solution*. Both the σ-optimization and the χ-optimization problem are NP-complete, i.e. exact solutions are apparently computationally intractable (Goldberg, 1992; Kou, 1977). Although NP-completeness means that is beyond the power of present computing techniques to find an exact σ-solution or χ-solution in a reasonable amount of time it does not mean that σ and χ cannot be used for real genomic data. Firstly, NP-completeness results typically refer to arbitrary input matrices. However, the genomic data has specific properties, such as the sparseness of the hybridization matrices, which may reduce the problem to a class of instances for which efficient algorithms exist. Secondly, once a problem is proved to be NP-complete, the algorithm design focus changes towards designing approximation algorithms. Successful techniques used in the design of approximation algorithms for other NP-complete problems turn out to provide powerful tools for dealing with the new NP-complete problems.

5.2. Vector-TSP optimization problems

In order to solve the σ and χ optimization problems we convert them into instances of what we call the *vector-TSP* (vTSP for short). The vTSP is an extension of the classic Traveling Salesman Problem (Garey & Johnson, 1979) to graphs on which the edges have vector weights. The components of the vectors represent different types of cost for traversing the edge. For example one component might be distance, another time, and a third dollar cost. Every instance of a f-optimization problem has a corresponding vTSP instance such that the two instances have an *identical* set of solutions. The vTSP approach creates a framework for a unified analysis of f-optimizations which can allow us to bootstrap on work which has been done on the classic TSP.

In Greenberg *et al.* (1994) we discuss in detail the theoretical import of the vTSP but for this paper we only define it and show it in relation to f-optimizations.

Definition 5. *An instance of the vector-TSP is an n vertex, vector-labeled, complete, undirected graph, $G = (V, E, cost_v)$ and a function $f: N^n \to N$ (where $cost_v$ is a function from edges to n-vectors over $\{0, 1\}$).*

The sparseness *of a vTSP graph, $s(G) = max_{e \in E}$ (the number of ones in the vector $cost_v(e)$). The* vector-cost *of a tour in G is the component-wise sum of the cost of the edges in the tour. The* f-cost *of a tour is f applied to the vector-cost.*

The vTSP f-optimization problem *takes as input a vTSP instance $I = (G, f)$ and returns a tour in G of minimal f-cost.*

The connection between optimizing σ and the standard TSP has appeared repeatedly in the literature (Alizadeh, 1993; Kou, 1977). In theorem 1 we generalize the relation to form a correspondence between any f-optimization of an incidence matrix and a vTSP problem. We are currently working on ways to extend the known approximation techniques for TSP to the more general vTSP.

Theorem 1. *For every instance of a chimeric optimization problem there exists a vTSP instance such that there is a 1–1 correspondance between tours in the vTSP and probe orderings in the chimeric optimization problem and*

(1) an optimal solution to one problem is also an optimal solution to the other.
(2) the sparseness of the vTSP graph is at most twice the sparseness of the incidence matrix.

The correspondences between incidence matrices and vTSP graphs and between probe orderings and tours is as follows and the remainder of the proof of the theorem is straight-forward.

Given an $m \times n$ incidence matrix A let G_A be the n vertex, vector-labeled complete undirected graph with cost function, $cost_v(e = (i, j)) = A[i, *] \otimes A[j, *]$. ($A[i, *] \otimes A[j, *]$ is the number of rows in which column i and column j of A differ in value.)

The matrix A and the graph G_A satisfy the following properties. For every probe order $\pi \in PO(A)$ there exists a tour τ_π in G_A such that the chimeric pattern of the map A^π and the vector-cost of τ_π coincide. i.e. $c(A^\pi) = cost_v(\tau_\pi)$. Conversely, for every

tour τ in G_A, there exists a probe order $\pi_\tau \in PO(A)$ such that $cost_t(\tau) = c(A^{\pi_\tau})$.

6. CONSTRUCTING MAPS BY APPROXIMATING χ

We are now ready to describe the first of our mapping algorithms. This algorithm is designed to minimize, χ, the maximum number of fragments into which any clone is divided by the map. Minimizing χ meets our objective of matching the known structure of true maps since it has been observed that almost all chimeric clones in a clone library have two inserts (Lander, 1992a). Furthermore, in Section 9 we show that, on simulated data, mimimizing χ does in fact produce maps which are close to the true map. Minimizing χ also meets our criteria of finding C1P maps when they exist, of being monotone, and of improving when given more information.

Although finding the exact minimum for χ is difficult we describe in this section an algorithm which is guaranteed to find a solution with close to minimal χ. Randomization within the algorithm can produce a variety of solutions, each of which is guaranteed to be close to optimal. Thus a user of our algorithms can compare several solutions in order to look for consensus elements. Of course, the solutions found by minimizing χ can also be compared with the solutions found by minimizing σ (see Section 7).

6.1. An approximation algorithm for χ

Our algorithm for minimizing χ proceeds in several steps. First, the matrix A is converted into a vector-labeled graph, G_A, as in the discussion of Theorem 1 on p. 212. Next, a spanning tree T is constructed for G_A that has a close to optimal χ-value for spanning trees of G_A. Lastly, a tour τ of G_A is obtained from a depth-first search of T. This tour τ gives the probe order π.

The algorithm for finding close to optimal spanning trees is called the cycle-basis algorithm. A high-level pseudo-code implementation of the algorithm is shown in Fig. 1, more details can be found in (Greenberg *et al.*, 1994). The idea of the algorithm is to start with *any* spanning tree (not necessarily one close to optimal), and to make successive local improvements to the tree until no more local improvements can be made. The local improvements are made by choosing an edge of G_A not in the tree and checking whether it can be swapped with an edge of the tree so as to reduce the vector-cost of the tree.

The key to efficient implementations with guaranteed close to optimal results is the determination of whether a new tree is an improvement over an old tree. As with the tours in Theorem 1 costs can be assigned to vector-labeled trees by taking a component-wise summation. Our algorithm considers only the $\log m$ (recall that m is the number of vector components) largest components. One tree is considered better than another if a sorted list of its $\log m$ largest components is lexigraphically smaller than the corresponding list for the other tree. For example if trees T_1, T_2, and T_3 have largest components (listed in sorted order) equal to (5, 3, 3, 2), (6, 1, 1, 1) and (4, 4, 4, 4) then T_1 is an improvement over T_2 but not over T_3.

It is shown in Greenberg *et al.* (1994) that the above algorithm results in a tree with close to optimal value regardless of the initial choice of tree or the order in which edges are chosen for creating potential improvements.

Theorem 2. *For an n-vertex graph G vector-labeled with m-dimensional vectors of sparseness $s(G)$, the cycle-basis algorithm finds a spanning tree T such that $\chi(T) = O(s(G)OPT + \log m)$, where OPT is the optimal χ-value of a spanning tree of G.*

The conversion of the spanning tree found by the cycle-basis-algorithm into a tour make use of a generalization of a folklore algorithm called "twice-around-the-tree" (Garey & Johnson, 1979). This algorithm has been shown to allow, for scalar-labeled graphs which obey the triangle inequality (it is never cheaper to travel via an intermediate node), a tour to be created from a minimum spanning tree which has at most twice the cost of the optimal tour. Our generalization is given below.

Lemma 1. *Given a vector labeled graph G, a function f, and a spanning tree T of G. If (1) the f-cost of T is less than c times the optimal f-cost of any spanning tree of G, (2) f is monotone and sub-additive, and (3) the triangle inequality is obeyed on G restricted to each component of the vector then the tour τ created by twice-around-the-tree applied to T has f-cost at most $2c$ times the optimal f-cost of any tour of G.*

Theorem 2 and Lemma 1 together imply that using the cycle-basis algorithm to find a tree and converting

```
Let T be any spanning tree of G = (V, E);
until T cannot be improved
    D = E − T;
    until T is improved or checked all edges D
        Let e' ∈ D ; D = D − {e}';
        Let C = cycle(T ∪ {e'}); e ∈ C − {e'};
        Let T' = (T ∪ {e'}) − {e};
        if T' has lower vector cost than T then T = T';
```

Fig. 1

the tree to a tour using twice-around-the-tree yields a provably good approximation to the optimum for χ. (It is trivial to show the χ is both monotone and sub-additive.)

Theorem 3. *For a n-vertex graph G vector-labeled with m-dimensional vectors of sparseness s(G), the algorithm for mimimizing χ described above finds a tour τ such that $\chi(\tau) = O(s(G) * OPT + \log m)$, where OPT is the optimal χ-value of a tour of G.*

7. CONSTRUCTING MAPS BY APPROXIMATING σ

Just as minimizing χ matches the observation that clones from real experiments rarely consist of more than two fragments, the minimization of σ, the total number of fragments, matches the observation that chimerism is an error introduced on data which is inherently C1P and thus should be relatively low.

As noted previously, the use of σ as an optimization function for physical mapping is not new. We do not believe, however, that it has previously been applied experimentally to the chimeric clone problem.

One interesting property of σ is that, since it is defined to be the sum of the elements of the vector cost of a tour and the vector cost is the sum of components along the tour, it is possible to reduce the vTSP for σ to the standard TSP. This allows us to apply the many ideas that have been used for solving TSP in the literature. In particular we have the following approximation guarantee based on a lemma of Christophedes (Garey & Johnson, 1979) which is an extension of the twice-around-the-tree lemma of the previous section.

Theorem 4. *There is a polynomial time approximation algorithm for the σ-optimization problem that is guaranteed to produce a tour of cost no larger than $\frac{3}{2}$* (cost of optimal tour).*

Many researchers (Bentley & Saxe, 1980; Johnson & Papadimitriou, 1985; Phillips, 1989) have looked at practical ways of computing the optimal TSP tour. As a first trial we chose the simplest greedy algorithm (see Fig. 2 for pseudo-code). To our surprise it performed very well on our simulated data and we have not yet felt it necessary to use more complex approaches. The simplicity of the greedy algorithm has the advantage that it runs so quickly that we could run 100s of trials of simulated data for problem sizes which reflect the expected sizes of real data. This allowed us to attain a confidence in the results which would not have been possible with a slower algorithm. As in the algorithm for minimizing χ there are random choices in the greedy algorithm which can be varied to give a set of solutions, all of which have close to optimal value.

8. EXPERIMENTAL DESIGN

Our hypothesis is that probe orders which have low values of σ or of χ will be close to the true order. In order to test this hypothesis we need experimental data in which we know the true ordering. Real experimental data is the ultimate test for any algorithm. However, in order to understand the ability of our algorithms to correct chimeric errors simulated data is indispensable. Simulated data, unlike real data, can be created so that the amount of chimerism is controlled. We can therefore look at cases in which we know the true order and we vary the amount of chimerism present.

In order to produce simulated experimental data we created our own simple generator. The generator takes as input the number of STS probes (n), the number of clones (m), the fraction of the clones which are chimeric (p), and the range of clone sizes (l to u). The intent is to mimic an experiment in which n STS are used and m clones are created. Of the m clones, $(1-p)m$ clones consist of a single fragment of DNA while pm clones consist of two fragments. Each clone hybridizes between l and u probes.

These parameters allow us to vary the size of the experiment and the quality of the clones. We wanted to know whether the techniques would work equally well for coarse experiments in which just a few probes were used and for finer grained experiments in which many probes were used. The *coverage* of an experiment (the average number of clones per probe) is known to affect the quality of maps. Very low coverages tend to leave gaps in the map while high coverages make the experiment more costly. By vary-

```
Let G be a TSP instance graph with nprobes vertices.
Let tour be a length nprobes vector initially all -1.

/* Start with vertex 0 */
current = 0;
/* Keep adding nearest vertex to endpoint which is not already in the tour to the tour */
For (i = 0; i < nprobes - 1; i++)
    nearest = vertex such that G[current,nearest] is minimal AND nearest not already in tour.
    tour[current] = nearest; /* add nearest to the tour */
    current = nearest; /* and continue tour from nearest */
End
```

Fig. 2

ing the number of clones in the simulated experiment we can determine what ranges of coverage allow good reconstructions.

Since our focus is on chimerism we, of course, also wanted to vary the amount of chimerism. Of particular interest was the case of no chimerism (which should yield high quality results) and the case of 40–50% chimerism which is the expected level for real data.

As the experiments progressed we realized that there was another important parameter, the size of the clones. The larger the number of probes to which a clone hybridizes the more information it provides to the algorithm. In general we assumed that the clones hybridized, on average, about 10% of the probes. In retrospect this may have been a little high and we plan to run additional tests with small clone sizes in the future.

8.1. The simulator

In our simulator we aimed for simplicity so as to have as good as possible an understanding of our input data. Simplicity necessitated some arbitrary decisions. In this section we describe what decisions we made and why we made them.

The first decision was how to model the creation of clones. Since the incidence matrix only contains information about clone/probe hybridization and not information about clone length we chose to model a non-chimeric clone as an interval of the sequence 1 to n. In other words a clone is defined solely by which probes it hybridizes. Specifically, a size (s) was chosen for each clone uniformly from l to u and then a starting point (x) was chosen uniformly from 1 to $(n-s+1)$. The clone was defined to hybridize probes x through $x+s-1$. Since the clone's start points and sizes were chosen uniformly this had the effect of modeling an experiment in which the probes were uniformly placed along the DNA. Certainly this is not the case in real experiments but it gave us a starting point. We hope to use more sophisticated probe models in the future.

The second decision was how to model chimerism. We chose to fix the number of chimeric clones and to divide the size of the chimeric clone among two pieces. Specifically, a size (s) was chosen uniformly from l to u as in the non-chimeric case. Then a piece size s_1 was chosen uniformly from 1 to $s-1$ and s_2 was set to $s-s_1$. A gap size (g) was then chosen uniformly from 1 to $n-s$ and a start point (x) from 1 to $n-s-g+1$. The chimeric clone was defined to hybridize probes x through $x+s_1-1$ and probes $x+s_1+g$ through $x+s_1+g+s_2-1$. This had the effect of forcing all chimeric clones to have two fragments and to have the same *total* size distribution as the non-chimeric clones. Again the reality is more complicated but this seemed a good first approximation. In particular we did not want chimeric clones to be larger than non-chimeric clones because then chimeric clones would contain more information than non-chimeric clones.

After the generator has created the required number of chimeric and non-chimeric clones it has all the information necessary to produce a probe/clone incidence matrix. The order of the probes is randomly scrambled and the incidence matrix is produced for the random order. This randomly ordered matrix is passed to the physical mapping routines while the inverse of the scrambling permutation (which represents the true map) is passed to the routines which check the quality of the maps produced.

8.2. Measures of map quality

Since the true map is known for our simulated data we were able to compare the maps produced by our algorithms to the correct answer. It is not, however, obvious how to make such a comparison. If the algorithm returns the exact true map then all is well. If instead the algorithm returns some other map how do we measure how close it is to correct.

We believe that innovative measures of map closeness will be important to the future development of mapping software. In particular we will eventually want a map to include not only an ordering of the probes but some confidence measure over parts of the ordering. Developing good representations of a map plus confidence will require understanding how one map is close to others. However, for the current paper we use a simple measure of map quality—adjacent pair consensus. That is, the number of adjacent probe pairs in the true map which are identified as adjacent in the algorithm's map.

So that the best quality map always has the same value we actually use the number of pairs *not* found as our cost measure. Thus a perfect solution, one matching the true map, has cost 0. Formally we define:

Definition 6. *The cost of the map π produced by algorithm $\mathcal{A}$ on a matrix whose true map is π_0 is the number of pairs $\{i,j\}$ such that $\pi_0(i)$ is adjacent to $\pi_0(j)$ but $\pi(i)$ is not adjacent to $\pi(j)$.*

9. EXPERIMENTAL RESULTS

In Figs 3 and 4 we summarize all our simulated experiments. The values for cases using 20 probes represent the average of 100 trials, those for 40 and 50 probes are averages of 50 trials, and those for 100 and 200 trials are averages of 10 trials. For each case the first four columns give the number of probes, the number of clones, the number of chimeric clones, and the range of clone sizes (in number of probes hybridized).

The next three columns represent the output of the greedy algorithm for reducing σ described in Section 7. They show the average σ and χ of the solutions as well as the average cost. The value of σ for the true map is (# of clones + # of chimeric clones).

probes	clones	chimeric clones	clone size	greedy sigma	greedy chi	greedy cost	chimin sigma	chimin chi	chimin cost	total cost	random cost
20	20	0	2-4	22.13	1.81	5.16	23.47	1.96	6.23	3.65	17.03
20	20	5	2-4	26.81	2.28	7.17	28.91	2.5	7.89	5.26	17.16
20	20	10	2-4	31.02	2.61	8.34	32.91	2.86	9.24	6.29	17.02
20	20	15	2-4	34.77	2.91	9.92	36.19	3.04	10.35	7.76	16.81
20	40	0	2-4	43.53	1.74	2.35	45.3	1.76	2.73	0.95	17.00
20	40	10	2-4	54.61	2.47	4.14	58.72	2.8	5.61	2.65	16.98
20	40	20	2-4	65.03	2.93	5.79	67.87	3.06	6.64	3.83	16.89
20	40	30	2-4	72.41	3.09	7	77.83	3.22	8.75	5.23	16.87
20	60	0	2-4	63.09	1.53	1.29	66.62	1.71	1.96	0.52	16.96
20	60	15	2-4	80.87	2.56	2.89	86.58	2.88	4.56	1.68	17.06
20	60	30	2-4	97.14	2.9	4.39	104.9	3.09	6.33	2.77	16.97
20	60	45	2-4	111.8	3.13	5.77	119.6	3.32	7.64	3.79	17.15
40	40	0	4-8	41.8	1.48	5.84	52.98	2.18	9.18	3.76	37
40	40	10	4-8	54.94	2.48	7.06	64.58	3.12	9.7	4.38	37.44
40	40	20	4-8	67.76	3.06	8.74	76.52	3.36	11.52	5.48	37.32
40	40	30	4-8	77.4	3.14	9.38	85.98	3.58	11.28	5.52	37.20
40	80	0	4-8	80.3	1.14	1.1	86.96	1.46	1.8	0.46	36.84
40	80	20	4-8	104.4	2.48	2.16	113.1	2.84	3.48	0.98	37.18
40	80	40	4-8	127.4	2.84	2.88	137.1	3.08	4.2	1.26	37.34
40	80	60	4-8	150.3	3.08	4.1	161.4	3.4	5.52	2.02	37.24
40	120	0	4-8	120.3	1.14	0.44	122.3	1.1	0.38	0.14	36.68
40	120	30	4-8	151.8	2.2	0.84	159.8	2.48	1.46	0.3	37.08
40	120	60	4-8	184.9	2.48	1.44	198.3	2.88	2.36	0.6	37.04
40	120	90	4-8	222.1	2.92	2.28	234.6	3.18	3.54	0.94	37.2
60	60	0	3-9	62.96	1.76	9.52	83.1	2.52	15.2	5.78	56.76
60	60	15	3-9	83.8	2.82	11.56	99.04	3.26	15.36	7.2	56.94
60	60	30	3-9	100.5	3.1	13.66	117.2	3.68	18.38	9.36	57.18
60	60	45	3-9	117.7	3.46	14.48	133.7	4.1	19.94	9.82	57.2
60	120	0	3-9	121.9	1.24	2.36	135.6	1.9	3.8	1.1	57.06
60	120	30	3-9	157.4	2.66	3.96	172.9	2.96	5.72	1.8	57.04
60	120	60	3-9	190.9	3	3.7	213.8	3.48	7.36	1.66	56.98
60	120	90	3-9	225.5	3.24	5.94	248.1	3.78	9.24	3.32	57.3
60	180	0	3-9	180.5	1.18	0.8	190.5	1.42	1.34	0.32	57.08
60	180	45	3-9	227.8	2.26	0.82	241.6	2.78	2.16	0.32	57.06
60	180	90	3-9	282.3	2.76	2.2	302.1	3.32	4.34	0.84	57.04
60	180	135	3-9	330.8	2.94	3.04	360.7	3.64	6.16	1.28	57.48

Fig. 3

Although the greedy algorithm does not always find an optimal solution, being slightly non-optimal apparently leads to only a small cost in quality of solution. The value of χ is 1 for the non-chimeric cases and 2 for all other cases. Although the goal of greedy is not to minimize χ it still comes close to the minimum in all cases.

The three columns following those for greedy represent the output of the algorithm to minimize χ described in Section 6. Although the values of χ and σ are not, in general, as good as those for the greedy σ minimization algorithm the χ minimization algorithm is still important. In many cases adjacency pairs missed by the greedy algorithm are found by the χ minimization algorithm. Thus we have added the column for total cost which counts only adjacent pairs not found by either algorithm. In addition using two different algorithms yield greater confidence in the adjacencies which occur in both algorithms. The ability to create a consensus map from several different algorithms is likely to be an important next step in the creation of mapping software.

probes	clones	chimeric clones	clone size	greedy sigma	greedy chi	greedy cost	chimin sigma	chimin chi	chimin cost	total cost	random cost
100	300	120	5-15	423.3	2.2	0.7	427.1	2.3	0.6	0.3	97.2
200	700	280	10-20	980	2	0	980	2	0	0	197.3
200	700	280	5-10	986.1	2.2	0.9	1032	3.2	4	0.5	197.3

Fig. 4

From a standpoint of algorithm development the trend information in Fig. 3 and in the graphs of Figs 5, 6 and 7 are especially noteworthy. The quality of the solution does not correlate with the number of probes but does correlate with the number of clones (i.e. coverage) and percentage of chimeric clones. Regardless of the number of probes and amount of chimerism, Fig. 7 shows that the solutions get better as the coverage increases. Similarly, regardless of the number of probes and amount of coverage, Fig. 6 shows that the solutions get worse as the chimerism increases. Thus the algorithms do capture important features of the biology.

As a double check that the generator was not responsible for these trends we also calculated the cost of a random permutation. (See the column labeled random cost in Figs 3 and 4.) The cost of the random permutation was always high and did not correlate with any of the three other parameters (probes, coverage and chimerism).

The values in the table of Fig. 4 represent a few cases which correspond to realistic parameters for an actual mapping experiment. They are based on the data presented in (Vollrath *et al.*, 1992; Chumakov *et al.*, 1992; Foote *et al.*, 1992). The number of probes, clones, and chimeric clones is relatively easy to extract from such data but the size of the clones is harder to predict. Clearly smaller sized clones would yield less information.

We are not ready to say that our algorithms always produce perfect maps in the presence of chimeric clones but we do find the results encouraging. In fact, in the probably most realistic case (the last row of Fig. 4 with 200 probes, a coverage of 3.5, chimerism of 40% and clones which covered between 2.5% and 5% of the probes) the greedy algorithm recovered the correct order in 8 out of 10 trials. On one of the two trials in which it did not recover the exact correct solution it missed on only two adjacencies. This is the equivalent of one piece of DNA being flipped in the middle of the order. In the other trial the greedy algorithm mised seven adjacencies. However, five of these adjacencies were indentified by the χ minimization algorithm.

10. CONCLUSIONS

The algorithmic strategies that we propose concentrate on the chimerism error in insolation from other

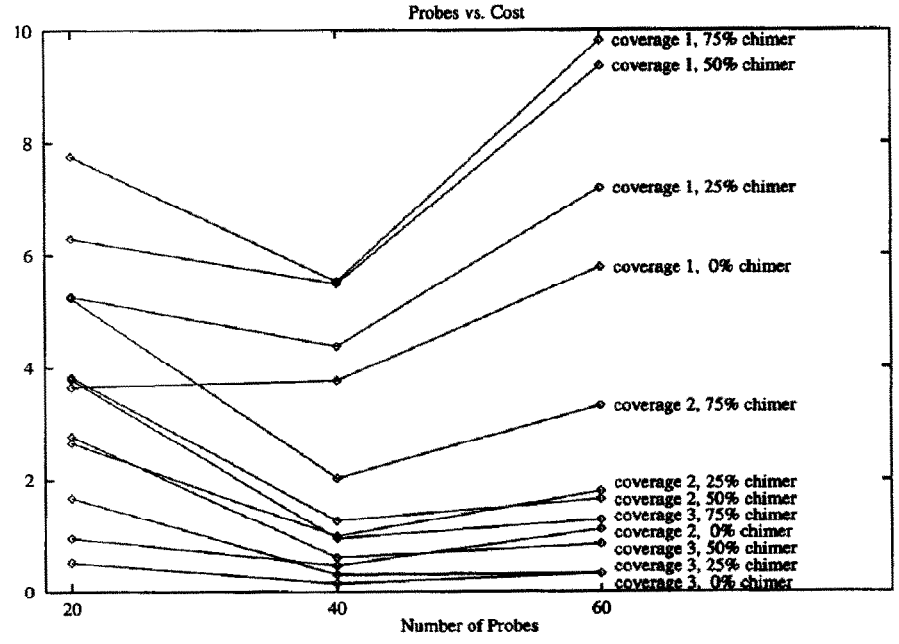

Fig. 5

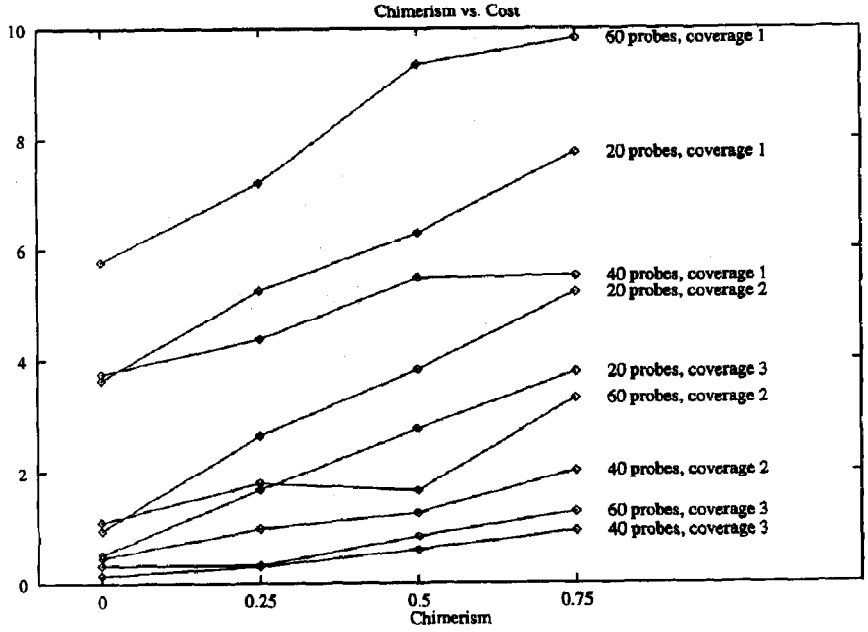

Fig. 6

errors that occur in mapping. We noted the difficulty of even defining the quality of a proposed physical map in the presence of errors. The goal of merely finding an ordering of the probes which is consistant with each clone being a contiguous piece of target DNA is no longer good enough. Since errors are probabilistic any probe ordering has some likelihood of being the true order.

The search for a way to make the problem well-defined led us to look for properties of most correct maps. Lander's observation that chimeric maps would have limited overall chimerism and rarely have clones containing more than two fragments opened the way for the investigation of optimization functions which correlate with the structure inherent in correct probe orderings. We defined a general class of

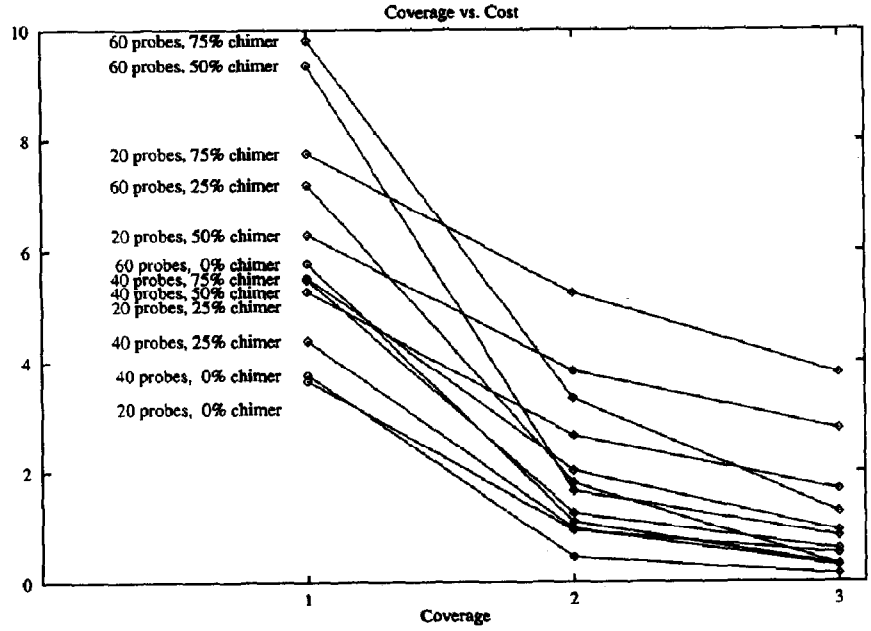

Fig. 7

functions which should have such correlations and designed algorithms for two functions, σ minimization and χ minimization. We then programmed the two algorithms in the C programmming language to run on Sparc workstations. We generated simulated experimental data which allowed us to look at trends in map quality as compared to experimental parameters. Both algorithms showed excellent correlation with experimental input—they yielded higher quality results when there was less chimerism and more clone coverage. When the algorithms were applied to data designed to mimic real experiments they continued to perform well. We intend to continue to refine the input data to make it more closely reflective of real data as well as to use real data as input in the near future.

10.1. Directions for future and related work

Our work on analyzing chimerism is only just begun. We realize that successful analysis will require the incorporation of additional information derived from other experiments besides STS hybridizations.

We also intend to extend our algorithmic approach to other mapping errors besides chimerism. The key will be to understand their corresponding combinatorial nature. Eventually we will need to address more than one error type at once. It is our hope that general techniques for designing approximation algorithms for optimization problems will help in dealing with such cases. Minimizing simultaneously several objective functions would provide us with a potential framework to model multiple error occurrence. A good candidate for a class of objective functions for which general approximation techniques would be helpful is given by the class $\mathcal{MS}$ of monotone and subadditive functions. Two such functions are σ and χ. We were unable to include much discussion of the theoretical computer science aspects of these problems in this paper but think they deserve careful consideration. The NP-completeness of the optimizational problems formulated for arbitrary—as opposed to genomic-like—clone/probe incidence matrices turned out to be a source of directions for research. Indeed, the apparent computational intractability established in (Goldberg, 1992; Kou, 1977) refers to problems that are more general than the ones that occur in actual mapping. However, it is difficult to discover restrictive conditions that, on one hand will model real data, and on the other hand will turn out to inspire feasible algorithmic avenues.

The theory of approximation algorithms for NP-complete problems includes tools for the design of approximate solutions to some optimization problems. We have extended tools developed for the TSP and added new tools in our effort to create algorithms with guaranteed performance. Another way in which the theory of NP-completeness was helpful was through the actual NP-completeness proofs. Understanding which parameters of the matrices are essential for the proofs led us to restrictions that turned out to be consistent with real data. Moreover, the restricted version of the problems have fast approximation algorithms for their solution.

Acknowledgments—It is a great pleasure to thank Eric Lander for suggesting this research topic to us, for many inspiring discussions, contributions and support along the way. We would like to acknowledge the important contributions to this paper derived from many conversations with our Sandia colleagues: Ernie Brickell, Leslie Goldberg, Paul Goldberg, Bruce Hendrickson, James Park, Cindy Phillips and Michael Sipser. Special thanks are due to Ernie Brickell for his contributions in the formative stages of this line of research. We would also like to thank Jim Orlin, David Torney and Mike Waterman for useful comments and suggestions.

Supported in part by the U.S. Department of Energy under contract DE-AC04-76DP00789.

REFERENCES

Alizadeh F., Karp R., Newberg L. & Weiser D. K. (1993) Physical mapping of chromosomes: a combinatorial problem in molecular biology. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 371–381.

Barillot E., Dausset J. & Cohen D. (1991) *Proc. Natl. Acad. Sci. U.S.A.* **88,** 3917.

Bentley J. & Saxe J. (1980) An analysis of two heuristics for the euclidean travelling salesman problem. In *Proc. 18th Allerton Conference on Communications, Control and Computing*, pp. 41–49.

Booth K. & Lueker G. (1976) *J. Comput. Syst. Sci.* **13,** 333.

Brown T. A. (1990) *Gene cloning*, 2nd edn. Chapman & Hall, London, 1990.

Chumakov I. *et al.* (1992) *Nature* **359,** 380.

Cohen D., Chumakov I. & Weissenbach J. (1993) *Nature* **366,** 698.

Craig A. G., Nizetic D., Hoheisel J. D., Zehetner G. & Lehrah H. (1990) *Nucleic Acid Res.* **18,** 2653.

Foote S., Vollrath D., Hilton A. & Page D. C. (1992) *Science* **258,** 60.

Garey M. & Johnson D. (1979) *Computers and Intractability*. Freeman and Co. San Francisco.

Goldberg P. (1992) The generalized consecutive ones property is NP-complete. Technical report, Sandia National Laboratories, Dec. 1992. also included in *Four Strikes against Physical Mapping of DNA* by Goldberg, Golumbic, Kaplan and Shamir, Tel Aviv University Tech Report 287.

Green E., Reithman H., Dutchik J. & Olson M. V. (1991) *Genomics* **11,** 658.

Greenberg D., Istrail S. & Atkins J. (1994) The chimeric clones problem. In preparation.

Johnson D. S. & Papadimitriou C. (1985) Performance guarantees for heuristics. In *The Traveling Salesman Problem* (Edited by Lawler G. *et al.*), pp. 145–180.

Kou L. T. (1977) *SIAM J. Comput.* **6,** 67.

Lander E. S. (1992a) Personal communication.

Lander E. S. (1992b) Lectures at the *2nd DIMACS/SIMS Workshop on Mathematical Sciences in Genomic Analysis*, At DIMACS.

Lander E. S. & Waterman M. S. (1988) *Genomics* **2,** 231.

Nelson D. & Brownstein B. H. (1993) *YAC Libraries: A user's Guide*. W. H. Freeman and Co., San Francisco.

Olson M., Hood L., Cantor C. & Botstein D. (1989) *Science* **245,** 1454.

Phillips C. (1989) Theoretical and experimental analysis of parallel combinatorial algorithms. PhD thesis, MIT, 1989. Also as technical report MIT/LCS/TR-462.

Schmidtt W. & Waterman M. (1991) *Adv. Appl. Math.* **12**, 412.
Torney D. (1991) *J. Mol. Biol.* **217**, 259.
Vollrath D., Foote S., Hilton A., Brown L., Beer-Romero P., Bogan J. & Page D. C. (1992) *Science* **258**, 52.