Four Strikes Against Physical Mapping of DNA

Paul W. Goldberg
Sandia National Labs
Dept. 1423, P.O. Box 5800
Albuquerque, NM 87185 USA
pwgoldb@cs.sandia.gov

Martin C. Golumbic
Dept. of Math. and Computer Science
Bar-Ilan University
Ramat Gan, Israel
golumbic@vm.biu.ac.il

Haim Kaplan
Department of Computer Science
Princeton University
Princeton, NJ 08544, USA
hkl@cs.princeton.edu

Ron Shamir
Department of Computer Science
Sackler Faculty of Exact Sciences
Tel Aviv University
Tel-Aviv 69978, Israel
shamir@math.tau.ac.il

April 1993, revised December 1993 and January 1995

Abstract

Physical Mapping is a central problem in molecular biology and the human genome project. The problem is to reconstruct the relative position of fragments of DNA along the genome from information on their pairwise overlaps. We show that four simplified models of the problem lead to NP-complete decision problems: Colored unit interval graph completion, the maximum interval (or unit interval) subgraph, the pathwidth of a bipartite graph, and the k-consecutive ones problem for $k \geq 2$. These models have been chosen to reflect various features typical in biological data, including false negative and positive errors, small width of the map and chimericism.

1 Introduction

In order to replicate and study a certain contiguous stretch of the DNA (a chromosome or a part of one), copies of the target DNA are cut using enzymatic or mechanical means into shorter segments, which in turn can be inserted into the DNA molecule of another host organism (cosmid, phage, yeast etc.). The host then preserves, replicates and reproduces the fragment of the target DNA as if it were part of its own genome. In particular, numerous copies of the fragment can be generated using the host's reproduction system, with all copies being identical to the original fragment. This process is called cloning, and the preserved fragments are called *clones*.

In the cloning process, all information on the relative location of the clones along the target genome is lost. On the other hand, since the procedure is applied to many copies of the original genome, fragments may overlap. The problem of reconstructing the relative position of the clones along the original stretch of DNA, based on this redundancy, is called Physical Mapping (PM), and the result is called a physical map. Physical mapping [9, 42] is a central problem in molecular biology. A physical map is an essential part of most sequencing, gene locating and cloning projects. One of the main initial goals set for the Human Genome Project is to obtain a detailed physical map of all human chromosomes [47, 11].

The key to map construction is determining overlap (intersection) between pairs of clones. There are various biological techniques for determining if two clones intersect [12, 38, 44]. All of these techniques involve obtaining some partial information on the contents of a clone, which we call the *fingerprint* of that clone. Intuitively, two clones should intersect if their fingerprints are sufficiently similar. Ideally one would like to compute the intersection probability for each pair of clones from their fingerprints, and then, based on the set of probabilities, determine the most likely placement of all the clones.

In this paper we show that several simplified versions of the physical mapping problem are NP-complete, and therefore unlikely to have exact, efficient algorithms. (For a discussion of complexity and NP-completeness in general see [18].) These problems assume a simplified input data and each tries to model differently uncertainties in the data. For the biologist, these results may provide insight to what makes a mapping problem difficult. For the computer scientist, these results may indicate directions for future research which can be pursued once a problem has been established to be hard (e.g., approximation, probabilistic analysis, parameterized analysis). ¿From our own experience, perhaps the most positive outcome which may come out of such results is using insights on the problem difficulty in order to modify and specialize the model and obtain a tractable problem. This process requires strong interaction between the biologist and the computer scientist in order to find out what is feasible both experimentally and computationally.

To formulate the problems graph theoretically we use the model of interval graph: (For a general introduction and basic terminology in graph theory see, e.g., [21].) G is an *interval graph* if one can assign an interval on the real line to each vertex so that two vertices are adjacent iff their intervals have a nonempty intersection. The set of intervals is called a *realization* of G. A graph is a *unit interval graph* if it is an interval graph which has a realization in which all intervals have the same length. Interval graphs have been studied intensively (cf. [21]) because of their wide applicability to practical problems and to biological problems in particular [5, 48]. In the context of PM, both the interval graph model, which allows arbitrary interval lengths, and the unit interval model are relevant: Some cloning techniques (YACs [8], P1 [46]) generate clones of variable lengths, while in other clone types (cosmid [13], Lambda [12]) clones are of roughly the same length.

The problems we study here are as follows:

A. Detecting false negatives in mapping with equal length clones with several complete digests:

Biological motivation: Suppose a set of clones is obtained by *complete digestion* of the genome by one or more restriction enzymes (cf. [12]). Since the digestion is complete, in such a set no two clones will overlap. Consider a PM project in which the set of clones (the clone *library*) consists of equal length clones (cosmids, lambda, etc.), and it is composed of several subsets of clones, where each subset is obtained by a complete digest with a different set of enzymes. We would like to reconstruct the map from clone overlap data, in the presence of "false negative" errors, i.e., some overlaps which are not detected experimentally. We wish to construct a map which is as close as possible to our input data, i.e., it assumes as few errors as possible.

Mathematical formulation: Assign a vertex for each clone and connect two vertices by an edge whenever the corresponding clones overlap according to the experimental data. Denote the disjoint sets of clones obtained from k complete digests by $S_1, \ldots, S_k$. Recall that a *proper coloring* of a graph $G = (V, E)$ is a function $c : V \rightarrow Z$ (i.e., assigning a "color" $c(v)$ to each vertex v), such that for every $(u, v) \in E$, $c(u) \neq c(v)$ (i.e., adjacent vertices have different colors). Since in each set S_i all clones are disjoint, it forms an independent set in the graph, and all its vertices can be assigned the same color, say i. The biological problem is thus equivalent to augmenting the graph by adding as few edges (corresponding to false negatives) as possible in order to obtain a unit interval graph, without violating the color constraints. A decision version of the problem is thus the following:

COLORED UNIT INTERVAL GRAPH COMPLETION:

Given a graph $G^1 = (V, E^1)$ with a proper coloring c for it, does there exist a unit interval graph $G = (V, E)$ such that $E \supseteq E^1$ and G is still properly colored by c?

Discussion: This problem can be viewed as a restriction of the *unit interval sandwich problem*: Given a graph $G^1 = (V, E^1)$ and $G^2 = (V, E^2)$ such that $E^1 \subseteq E^2$, does there exist a "sandwich graph" $G = (V, E)$ where $E^1 \subseteq E \subseteq E^2$ and G is a unit interval graph? In our case $E^2 = \{(u, v) | c(u) \neq c(v)\}$. Sandwich formulations have been used in the past to model ambiguity in the data of PM, where for each two clones either we are sure that they intersect, or we are sure that they do not, or we have no information. The interval sandwich problem was defined in [24] in the context of temporal reasoning, where it was shown to be NP-complete. Sandwich problems for other graph families were later systematically studied in [22]. NP-completeness proofs were given in [23] for the unit interval sandwich and the colored version with non-unit intervals. Fellows et al. [16] have given another proof for the latter result, and have also shown that the colored interval graph completion is polynomial if the bound on the number of colors is fixed. They also showed that if that bound is viewed as a parameter, the problem is $W[t]$-hard for all t. (For an exposition of the W hierarchy see [15].)

In Section 2 we sketch a simple NP-completeness proof for problem (A). This also provides a new NP-completeness proof for the unit interval sandwich problem, which is much simpler than the proof in [23]. Quite recently, two of us [27] have given a polynomial algorithm for the version of problem (A) in which the number of colors is fixed. On the other hand, they have given a W[1]-hardness proof for the parameterized version of problem (A), where the number of colors is a parameter. While this result implies the NP-completeness of problem (A), readers of the 10-page parametric reduction there may appreciate the simplicity of the one here.

B. Detecting false positive errors:

Biological motivation: In some fingerprinting techniques (e.g., fingerprints based on restriction enzyme fragments [7]) the accuracy of the decision about clone pair intersection is proportional to the amount of overlap. Assuming that false negative errors are caused by insufficient overlap between clones, they will, in many cases, cause the decomposition of the map into more components (contigs) than there are in the "real" map, without destroying the interval structure. Under this assumption (which is made in most statistical analyses, e.g. [35]), the few but crucial false positive errors are those which may destroy the interval property, and must be detected. Problem (B) models this situation where we wish to find a realization with as few false positive errors as possible.

Mathematical formulation: Form the clone intersection graph as in A. The problem then is equivalent to the following:

MAXIMUM INTERVAL SUBGRAPH:
Given a graph $G = (V, E)$ find a subgraph $G' = (V, E')$ where $E' \subseteq E$, G' is interval and $|E'|$ is maximum.

Discussion: In Section 3 we prove that this problem is NP-complete, even under severe restrictions on the graph, both for arbitrary and for unit intervals. Maximum subgraph problems have been studied for many graph properties, see [18, pp.194–199]. The analogous completion problem where one asks for *adding* as few edges as possible to a graph and making it interval is known to be NP-complete [18, problem GT35],[28]. The NP-completeness of the unit interval graph completion problem is implied by the proof of Yannakakis [51] for chordal graph completion, as the graphs generated by that proof are chordal if and only if they are unit interval.

C. Generating minimum-width map in the presence of false negatives:

Biological motivation: We say that a physical map has *width k* if k is the largest number of mutually overlapping clones, in any position in the map. Put differently, the width is the largest number of intervals cut by any vertical line across the map. We have observed that most published physical maps have very small width: Typically, the width in genome mapping experiments ranges between 5 and 15, while the total number of clones may be thousands [32, 43, 10]. The reasons for this phenomenon are the statistical distribution of the clones along the genome, and the need to minimize the experimental effort which grows (up to quadratically) with the number of clones. Since trying to minimize the number of false negatives needed to form a map is known to be hard, (this is the completion problem discussed above) a reasonable alternative goal is to recognize false negatives which will yield a map with minimum width. I.e., the parameter we would like to optimize on now is the width of the resulting map, while we do not directly minimize the number of corrected false negatives. We show here that the decision version of this problem is NP-complete, even under the severe restrictions on the input.

Mathematical formulation: The width of a map as defined above is equivalent to the *clique size* of the corresponding interval graph. (The clique size of a graph is the size of the largest set of vertices each two of which are adjacent). There are several notions of graph width, the one which we shall need here is called pathwidth. A formal definition of pathwidth we be given in section 4. We shall consider the problem when the input graph is assumed to be bipartite:

PATHWIDTH OF A BIPARTITE GRAPH:
Given a bipartite graph G and a constant k, is there an interval supergraph of G whose clique size is at most k?
Equivalently, is the pathwidth of G at most $k - 1$?

Discussion: The Pathwidth problem is NP-complete on arbitrary graphs [29, 2], and even for chordal graphs [25] and (using the equivalence to node search [30]) for planar graphs with vertex degrees at most three [41]. On the other hand, it is solvable in linear time when k is fixed [31, chapter 11]. We shall prove that under the restriction of the problem to bipartite graphs it remains NP-complete.

D. Handling probe-clone errorless data under chimericism:

Biological motivation: In some fingerprinting techniques (cf. [3, 37]) one has a collection of clones, and a set of short genomic inserts (called *anchors* or *probes*). A probe defines a *single location* where a given subset of the clones coincide. For each probe/clone pair, biological techniques can find out whether the clone contains the probe as a subsequence. Then the problem is to construct an ordering in which the probes could occur along the original chromosome, that is consistent with this probe/clone incidence information. This has an efficient solution if we assume that all clones are simple substrings of the chromosome. We show that the problem is potentially much harder in the presence of *chimericism.* Chimericism (see [49]) is the result of concatenating two or more clones from different parts of the genome, producing a "chimeric clone" – one that is no longer a simple substring. Given a collection of clones obtained from a single chromosome, it is not known which are chimeric, but some clone libraries suffer from high rates of chimericism (estimated to be as high as 60% [49]). However, most chimeric clones consist of the concatenation of just two substrings, so we would like to find probe orderings that show the clones as consisting of one or two subsequences.

Mathematical formulation: Consider the probe-clone incidence matrix M, with rows indexed by probes and columns by clones, and 1 in position (i, j) if clone j contains probe i. If each clone is a contiguous interval, the problem is to permute the rows so that the ones in each columns are consecutive. In the presence of chimeric clones, each of which is a concatenation of at most two substrings, the problem corresponds to finding an arrangement of the rows of M such that there are at most two sequences of consecutive ones in each column. More generally, let us say that a $(0, 1)$ matrix has the *k-consecutive ones property* (k-C1P) if there exists a row order such that in each column the occurrences of all ones appear in at most k consecutive blocks. The decision version of our problem is the following:

The k-CONSECUTIVE ONES Problem:

Does a given $(0, 1)$ matrix have the k-consecutive ones property?

Discussion: For $k = 1$ this problem is polynomial [6]. We show in Section 5 that deciding if there exists an order satisfying the property is NP-complete for every $k \geq 2$. This applies in particular to the case where $k = 2$ which is of interest to PM. Note that if instead we have pairwise overlap information between such chimeric clones, then the problem is to find a 2-interval realization of the associated graph (where each vertex corresponds to one or two intervals on the real line). This problem has already been shown to be NP-complete by West and Shmoys [50].

Other generalizations of the C1P have been shown to be NP-complete, in the context of consecutive retrieval in databases. Kou [34] proves the NP-completeness of finding a permutation of the rows that minimizes the *total number* of sets of consecutive ones in a matrix. The result here mirrors that of [50] for the related problem of recognizing k-interval graphs. However, the kind of information contained in an interval graph about the corresponding sets of intervals is different from the information contained in an incidence matrix, and it does not appear possible to prove this result by a simple application of [50] (or vice versa.)

There have been several statistical and algorithmic studies of PM. The papers [35, 4, 3] study probabilistic models for the distribution of the number and size of connected components in the map, as a function of experimental parameters, and its implications for mapping project design. Alizadeh et al. [1] have recently investigated the model of [35] for the special case of hybridization fingerprints with equal-length clones. Using heuristics for a combinatorial version of the problem, they obtained encouraging results with simulated data. They also prove NP-hardness of two variants of the problem. Goldstein and Waterman [20] show that restriction mapping is NP-complete. These related hardness results are not equivalent to those presented here.

2 Colored Unit Interval Graph Completion

Given a set of intervals on the real line, one can define a partial order on the intervals by $a \prec b$ if and only if the interval a is completely to the left of interval b. We use this *interval order* to give a reduction from BETWEENNESS [45]: Given a set of elements $S = \{a_1, \ldots, a_n\}$, and a set $T = \{T_1, \ldots, T_m\}$ of ordered triplets of elements from S, where $T_i = (a_{i_1}, a_{i_2}, a_{i_3})$ $i = 1, \ldots, m$, does there exist a one-to-one function $f : S \to \{1, 2, \ldots, n\}$ such that either $f(a_{i_1}) < f(a_{i_2}) < f(a_{i_3})$ or $f(a_{i_1}) > f(a_{i_2}) > f(a_{i_3})$ for $i = 1, \ldots, m$?

Theorem 2.1 *The colored unit interval graph sandwich problem is NPC.*

Proof. The problem is in NP since recognition of unit interval graphs can be done in linear time [6, 33, 14]. We provide a reduction from Betweenness: Given a set of triplets $\{T_1, \ldots, T_k\}$, of elements from the ground set $S = \{s_1, \ldots, s_n\}$, suppose w.l.o.g. that n is odd, say, $n = 2m - 1 > 1$. Construct a graph $G = (V, E)$ where the vertex set is $V = U \cup W_1 \cup \cdots \cup W_k$. $U = \{u_i | s_i \in S\}$ i.e., each $u_i \in U$ corresponds to an element $s_i \in S$. The vertex set W_i corresponding to triplet T_i consists of $2n$ vertices: $W_i = \{v_i^1, \ldots, v_i^{2n}\}$. Define for each triplet $T_i = (x, y, z)$ the edges (x, v_i^1), (v_i^1, v_i^2),(v_i^2, v_i^3), $\ldots$, (v_i^n, y),(y, v_i^{n+1}),(v_i^{n+1}, v_i^{n+2}), $\ldots$, (v_i^{2n-1}, v_i^{2n}),(v_i^{2n}, z). In other words, W_i together with the vertices x, y, z form a chain of $2n+3$ vertices, with x, y in the ends and z in the middle. The union of the k chains is the edge set E. Finally, color $c(u) = 0$ for each $u \in U$, $c(v_i^t) = c(v_i^{n+1-t})$ $=c(v_i^{n+t}) = c(v_i^{2n+1-t}) = (i-1)m + t$ for $i = 1, \ldots, k$ and $t = 1, \ldots, m$. Note that the coloring is proper.

Suppose there is a linear order on S solving the Betweenness problem. Then one can place n disjoint unit open intervals, one for each vertex in U, on the line segment $[0, n]$ in the same order. Since the middle interval for each triplet appears between the other two, it is easy to verify that one can now add unit intervals for the W-vertices in each chain so that no like-colored intervals are overlapping and the mandatory overlaps are respected (see figure 1)

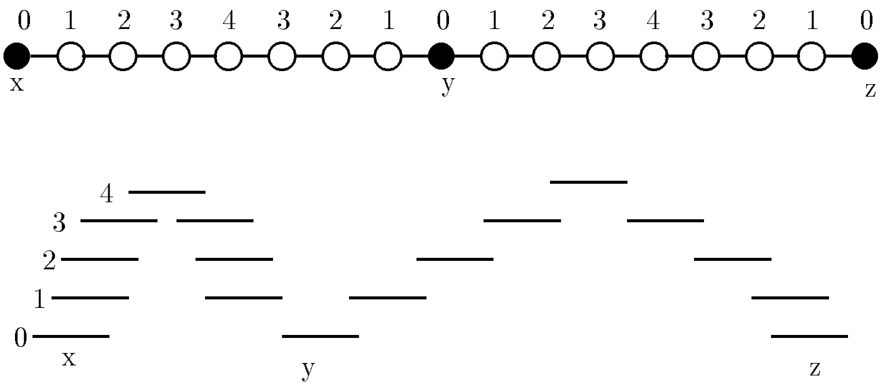


Figure 1: Top: A chain corresponding to the the triplet (x, y, z), for the case where $m = 4$. Numbers denote colors. Bottom: A unit interval representation of the chain. Numbers denote the color of all intervals on that level.

Conversely, suppose that there is a solution to the colored sandwich problem. Since all vertices in U have the same color, their intervals must be pairwise disjoint in every realization, and thus form a linear order along the line. The coloring of the chains guarantees that in each chain, the intervals to the left and to the right of the middle vertex in it do not cross. Thus, the interval corresponding to the middle element in each triplet must be between the intervals of the other two elements. Hence, any interval realization of the sandwich graph induces a linear order on the set S which satisfies the betweenness condition. ∎

3 Maximum Interval Subgraph Problems

The graph $G' = (V', E')$ is a *subgraph* of the graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. A graph is *cubic* if every vertex has exactly three edges incident on it. A graph is *chordal* if it does not contain an induced cycle of length greater than three. Three vertices in G form an *asteroidal triplet* in G if they are pairwise nonadjacent, and any two of them are connected by a path which does not pass through the neighborhood of the third. In the proof below, we shall use the fact that an interval graph cannot contain an asteroidal triplet and must be chordal [36]. We refer to the decision version of the problem where one asks for an interval subgraph with at least k edges.

Theorem 3.1 *The Maximum Interval Subgraph problem is NP-complete.*

Proof. The problem is in NP since one can recognize an interval graph in linear time [6, 26]. We give a reduction from the Hamiltonian Path problem restricted to planar cubic graphs [19]. Given a cubic graph $G = (V, E)$ with $|V| = n$, we create a new graph $G' = (V', E')$ by "blowing up each vertex into a triangle, and maintaining the cubic property of the graph". Precisely, the vertex set is $V' = \{v_1^i, v_2^i, v_3^i \mid v^i \in V\}$. We call v_1^i, v_2^i, v_3^i the *representatives* of v^i. The edge set consists of two types of edges $E' = E^{new} \cup E^{old}$, as follows:
(1) $E^{new} = \{(v_1^i, v_2^i), (v_2^i, v_3^i), (v_3^i, v_1^i) \mid v^i \in V\}$. Hence, the three representatives of each vertex $v^i \in V$ form a triangle in G'.
(2) For each original edge $(v^i, v^j) \in E$ define an edge $(v_k^i, v_l^j) \in E^{old}$, choosing the indices k, l in such a way that all E^{old}-edges are nonadjacent. In other words, the edge connects representatives of its original endpoints, and for every $v \in V$, each of the three original edges incident on v in G is incident on a different representative of v. Call the edges in the E^{new}-triangles *new* and the E^{old}-edges *old*. Finally, set $k = 4n - 1$.
The reduction is clearly polynomial. Note that G' is cubic and has $3n$ new edges and $\frac{3n}{2}$ old edges. Moreover, G' is planar since it does not contain a K_5 or a $K_{3,3}$. (Equivalently, it is easy to see that starting from a planar representation of G, the choice in (2) can be made so that no edges cross.) We claim that G has a hamiltonian path iff G' has an interval subgraph with at least $4n - 1$ edges.

Suppose G contains a hamiltonian path P. Delete all old edges in G' which do not correspond to edges in P. Denote the resulting subgraph by $\tilde{G}$ (see figure 2(a)). $\tilde{G}$ is an interval graph, as can be verified by its interval representation as depicted in figure 2(b). Moreover, $\tilde{G}$ contains exactly $4n - 1$ edges: $3n$ new edges and $n - 1$ old edges.

Conversely, suppose $\tilde{G} = (V, \tilde{E})$ is an interval subgraph of G' with $|\tilde{E}| \geq 4n - 1$. First, we prove that $\tilde{G}$ contains all the new edges of G' and exactly $n - 1$ old edges:

For every vertex v in G, let S_v be the subgraph of G' induced by the three representatives of v together with their three neighbors. S_v is an asteroidal triplet, and deleting *any* edge in S_v cancels the asteroidality of the triplet. There are n such triplets. Since $\tilde{G}$ is an interval graph,

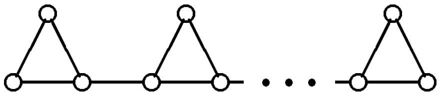
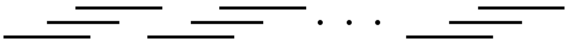
(a)  (b) 

Figure 2: (a) The interval subgraph $\tilde{G}$. (b) A unit interval realization of $\tilde{G}$.

it cannot contain an asteroidal triplet [36], so out of each such S_v, at least one edge must be missing in $\tilde{G}$.

Each S_v contains three new and three old edges. Suppose two new edges are missing from some S_x. Since the total number of edges removed from G' to form $\tilde{G}$ does not exceed $\frac{n}{2}+1$, it turns out that at most $\frac{n}{2}-1$ additional edges must be removed from G' in order to cancel the remaining $n-1$ asteroidal triplets $\{S_v \mid v \neq x\}$. Since each (new or old) edge is contained in at most two S_v-s we obtain a contradiction. Hence, out of the edge set of each S_v, at most one new edge can be missing in $\tilde{G}$. In particular, the representatives of each vertex induce a connected subgraph in $\tilde{G}$.

Let H be the graph obtained from $\tilde{G}$ by contracting all new edges. By what we have just proved, the number of vertices in H is exactly n, one for each original vertex in G. Moreover, H is acyclic, since the existence of a cycle in H would imply the existence of a chordless cycle at least twice as long in $\tilde{G}$, contradicting the fact that $\tilde{G}$ must be chordal, as an interval graph. It follows that H contains at most $n-1$ edges. Thus, $\tilde{G}$ contains at most $n-1$ old edges. Since the total number of edges in $\tilde{G}$ is at least $4n-1$, $\tilde{G}$ must contain *all* the $3n$ new edges in G' and *exactly* $n-1$ old edges.

Since H is acyclic with $n-1$ edges and n vertices it must be connected. Suppose H contains a vertex v with degree three. Since we have just proved that $\tilde{G}$ contains all the new edges, this implies that the asteroidal triplet S_v from G' exists also in $\tilde{G}$, contradicting the fact that $\tilde{G}$ is an interval graph. Hence, H defines a hamiltonian path in G. ■

The same reduction in the theorem above applies also to the maximum *unit* interval subgraph problem: Simply observe that the graph $\tilde{G}$ is actually a unit interval graph. In fact, a unit interval realization of it is drawn in figure 2(b).

Corollary 3.2 *The Maximum Unit Interval Subgraph problem is NP-complete.* ■

Remark 3.3 *The reductions above imply the stronger results that the Maximum Interval (and Unit Interval) Subgraph problems are NP-complete even when the input is restricted to planar cubic graphs.*

4 Pathwidth of bipartite graphs

A *path decomposition* of a given graph $G = (V, E)$, is a sequence of subsets of V, $(X) = (X_1, \dots, X_l)$ such that

(1) $V = \cup_i X_i$

(2) For each edge $(u, v) \in E$, there exists some $i \in \{1, \dots, l\}$ so that both u and v belong to X_i.

(3) For each $v \in V$ there exist some $s(v), e(v) \in \{1, \dots, l\}$ so that $s(v) \leq e(v)$, and $v \in X_j$ if and only if $j \in \{s(v), s(v) + 1, \dots, e(v)\}$.

The *width* of a path decomposition (X) is defined by $pw_X(G) = \max\{|X_i| \mid i = 1, \dots, l\} - 1$. The *pathwidth* of G, denoted $pw(G)$, is the minimum value of $pw_X(G)$ over all path decompositions, i.e., $pw(G) = \min\{pw_X(G) \mid (X)$ is a path decomposition of $G\}$. The *PATHWIDTH* problem is to decide for a given graph G and a given integer k if $pw(G) \leq k$.

The notions of pathwidth and interval graphs are related by the following well-known observation:

Lemma 4.1 *(cf. [39]) For every graph G, the pathwidth of G is one less than the least clique size of any interval supergraph of G.*

Hence, computing the pathwidth is equivalent to finding an interval supergraph with minimum clique size.

Theorem 4.2 *Computing the pathwidth of bipartite graphs is NP-complete.*

Proof. Reduction from PATHWIDTH on arbitrary graphs: Given a graph $G = (V, E)$ where $|V| = n$ and a parameter k as the input of PATHWIDTH, we define a new bipartite graph as follows: Replace each original edge by $n + 1$ parallel edges and add a vertex on each such edge. Formally, the new bipartite graph is $G' = (V, U, E')$, where $U = \{e_i \mid e = (u, v) \in E,\ i = 1, \dots, n + 1\}$, and $E' = \{(u, e_i), (v, e_i) \mid e = (u, v) \in E,\ i = 1, \dots, n + 1\}$. The parameter in the new problem is set to $k + 1$. The reduction is clearly polynomial.

Suppose $(X_1, \dots, X_r)$ is a width k path decomposition of G. Fix an edge $e = (u, v) \in E$. There must exist at least one original set X_i such that $u, v \in X_i$. Pick one such set and define a new decomposition

$$(X_1, \dots, X_{i-1}, X_i, X_i \cup e_1, X_i \cup e_2, \dots, X_i \cup e_{n+1}, X_{i+1}, \dots, X_r)$$

Repeat the process with each edge $e \in E$. Note that we allow only original sets to be "expanded" in this way, possibly expanding the same set several times for several edges. It is easy to verify that the final result is a path decomposition for G' with width at most $k + 1$.

Conversely, let $(Y') = (Y'_1, \ldots, Y'_t)$ be a path decomposition of G' with width $k+1$. Without loss of generality no Y'_i induces an independent set, since if there is one it can simply be deleted from the decomposition. Hence, every Y'_i contains at least one vertex from U. Let $Y_i = Y'_i \cap V$. $|Y'_i| \leq k-1$ for each i. We shall show that $(Y) = (Y_1, \ldots, Y_t)$ is a path decomposition for G. Clearly, for each $v \in V$ the set $\{i | v \in Y_i\}$ is contiguous in the decomposition since the property holds for the original decomposition (Y').

Let $e = (x, y) \in E$ and suppose x, y do not appear together in any set in (Y). Without loss of generality, suppose $j = \max\{i | x \in Y_i\} < l = \min\{i | y \in Y_i\}$. Since in G' each of the vertices in the set $S = \{e_i | i = 1, \ldots, n+1\}$ is incident on both x and y, it follows that $S \subseteq Y'_i$ for $j \leq i \leq l$. Since $n+1 > k+1$ this is a contradiction. Hence, for every edge $e = (x, y) \in E$, x, y appear together in some set of the decomposition (Y') and therefore also in the decomposition (Y). ■

It is now natural to ask the same problem with unit intervals: Here we can obtain an even stronger result, using the following recent theorem from [27]: Computing the bandwidth is equivalent to finding a unit interval supergraph with minimum clique size. Hence, the NP-completeness of BANDWIDTH on trees [17, 40], immediately implies that given a graph G, finding a unit interval supergraph of G whose clique size is minimum is NP-complete, even if G is a binary tree, or a caterpillar with hair-length three.

5 k-Consecutive Ones Matrices

Recall that the *k-consecutive-ones property* (abbreviated to k-C1P) of a (0,1)-matrix M is the property that the rows of M can be permuted such that within each column, there are at most k sequences of consecutive ones. We show in this section that testing the k-C1P is NP-complete for all $k \geq 2$.

Using the terminology of [49], we refer to a maximal sequence of consecutive ones in a column as a *contig*. Then we say that a matrix is *k-consecutive* if there are at most k contigs in each column. (Thus the k-C1P is the property that some permutation of the rows makes a matrix k-consecutive.) We refer to a permutation of the rows of M which makes M k-consecutive as a *k-consecutive arrangement* of M.

We use the following notation: A column in a (0,1)-matrix is represented by a list of the positions at which a 1 occurs. Thus $[1, 2, 5]$ represents a column of zeroes and ones in which only the first, second and fifth entries are ones.

Lemma 5.1 *For any $k \geq 2$, any (0,1)-matrix M and any two rows, a set of columns can be appended to M so as to constrain those two rows to be adjacent (in either order) in any*

k-consecutive arrangement of M.

Proof. Consider the column $[1, 2, \ldots, k, k+1]$. If this column occurs in a matrix M then it requires at least two of the first $k+1$ rows to be adjacent in some order. Now suppose that M contains the columns

$$\{[1, 2, \ldots, k, l] \ : \ l = k+1, k+2, \ldots, 3k+1\}$$

Then if in fact no two of the first k rows are adjacent in some k-consecutive arrangement of M, then in that arrangement, each of the subsequent $2k+1$ rows must be adjacent to one of the first k rows. However, this is impossible since there are at most $2k$ positions adjacent to these rows. So we conclude that the above set of columns must imply that two of the first k rows must be adjacent. (Note that no further constraints are implied by these columns on the positions of the subsequent $2k+1$ rows in k-consecutive permutations of M.)

We can now repeat this entire construction in the same way but so as to constrain two of rows $1, \ldots, k-1$ and row l to be adjacent, for $l = k+1, \ldots, 3k-1$. The resulting set of columns imply that two of rows $1, \ldots, k-1$ must be adjacent. Continuing in this fashion, we can eventually force rows 1 and 2 in particular to be adjacent. ■

As a consequence of this result, we have that for any k, there is in fact a matrix which does not have the k-C1P. We can construct such a matrix by forcing the first $2k+1$ rows to be adjacent and in order (row i is adjacent to row $i+1$ for $i = 1, \ldots, 2k$), and then including the column $[1, 3, 5, 7, \ldots, 2k+1]$. Note that the construction uses only $3k+1$ rows, but $(\Theta(k))!$ columns, which is a weak upper bound. It is relatively easy to prove, using the probabilistic method, the existence of matrices whose size is polynomial in k which do not have the k-C1P. The following proof, of some independent interest, shows this fact by *constructing* such a matrix.

Let P be the $(k+1)^2 \times (k+1)^2$ permutation matrix

$$\begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{pmatrix}$$

Let M be the $(k+1)^3 \times (k+1)^3$ matrix

$$\begin{pmatrix} P^{a_{1,1}} & \ldots & P^{a_{1,(k+1)}} \\ \vdots & \ddots & \vdots \\ P^{a_{(k+1),1}} & \ldots & P^{a_{(k+1),(k+1)}} \end{pmatrix}$$

M has exactly $k+1$ ones per row and $k+1$ ones per column. We next show that there exists an appropriate choice of values for the powers $a_{i,j}$ such that M does not have the k-C1P.

We want to choose the powers $a_{i,j}$ such that there is no 2×2 submatrix containing all ones. Such a submatrix would have to have its four entries from four distinct $P^{a_{i,j}}$ submatrices with powers of the form $a_{\alpha,\gamma}, a_{\alpha,\delta}, a_{\beta,\gamma}, a_{\beta,\delta}$ (where we may assume $\alpha < \beta$, $\gamma < \delta$). We claim that these powers must satisfy the equation $a_{\beta,\delta} = a_{\alpha,\delta} + a_{\beta,\gamma} - a_{\alpha,\gamma}$.

To prove this claim, note first that the one in row x of $P^{a_{i,j}}$ is in column $(x - a_{i,j})$ of $P^{a_{i,j}}$, and the one in column y of $P^{a_{i,j}}$ is in row $(y + a_{i,j})$ of $P^{a_{i,j}}$. Now suppose that the above matrices have a 2×2 submatrix containing all ones. Let x be the row number in $P^{a_{\alpha,\gamma}}$ where an entry of this submatrix occurs. The one in row x of $P^{a_{\alpha,\gamma}}$ is in column $(x - a_{\alpha,\gamma})$ of $P^{a_{\alpha,\gamma}}$, and corresponds to the one in row x of $P^{a_{\alpha,\delta}}$, which must be in column $(x - a_{\alpha,\delta})$ of $P^{a_{\alpha,\delta}}$. The corresponding one in column $(x - a_{\alpha,\gamma})$ of $P^{a_{\beta,\gamma}}$ must be in row $(x - a_{\alpha,\gamma} + a_{\beta,\gamma})$ of $P^{a_{\beta,\gamma}}$. The corresponding one in row $(x - a_{\alpha,\gamma} + a_{\beta,\gamma})$ of $P^{a_{\beta,\delta}}$ must be in column $(x - a_{\alpha,\gamma} + a_{\beta,\gamma} - a_{\beta,\delta})$ of $P^{a_{\beta,\delta}}$. This must be the same column as the one in $P^{a_{\alpha,\delta}}$ (ie. column $(x - a_{\alpha,\delta})$), hence the equation follows.

We may choose the $a_{i,j}$ to avoid this situation: there are $(k+1)^2$ possible values for each of the $a_{i,j}$, and the assignment of a value to a different one of the $a_{i,j}$ will eliminate at most one possibility for each of the others. Since this set has only $(k+1)^2$ members, we may assign values to them all without introducing the submatrix that we wish to avoid.

Any permutation of the rows involves only $(k+1)^3 - 1$ adjacencies of rows, and so for appropriate choices of the $a_{i,j}$ described above, makes at most $(k+1)^3 - 1$ pairs of ones in columns adjacent. Hence there will be at least one column with none of its ones made adjacent.

We next show the main result of this section, that it is NP-complete to test a matrix for the 2-C1P. We prove the result in the standard way, using a reduction from 3SAT, the problem of finding a satisfying assignment to a boolean formula in 3CNF (cf. [18]).

Theorem 5.2 *Testing for the 2-C1P is NP-complete.*

Proof. Reduction from 3SAT. We show how to transform a formula Φ in 3CNF into a (0,1)-matrix M_Φ in polynomial time, in such a way that Φ is satisfiable if and only if M_Φ has the 2-C1P.

Let Φ be a 3CNF formula over n variables $\{v_1, \ldots, v_n\}$, and with m clauses $\{C_1, \ldots, C_m\}$. We construct the associated matrix M_Φ (having $2n + 5m$ rows and $6m + 10n - 5$ columns) as follows.

Let $r_1, \ldots, r_{2n+5m}$ be the rows of M_Φ (in order from top to bottom). Associate with variable

v_i the rows r_{2i-1} and r_{2i}. v_i is to represent the statement (about a permutation of the rows) "r_{2i-1} is above r_{2i}". So Φ represents a statement about a permutation of the rows of M_Φ. We construct M_Φ to ensure that any 2-consecutive arrangement of M_Φ corresponds to a satisfying assignment of Φ, and that if Φ is satisfiable then M_Φ has the 2-C1P.

We introduce a set of columns which have the following effects:

- In any 2-consecutive arrangement of M_Φ, r_{2i-1} must be adjacent to r_{2i}, for $i = 1, \ldots, n$.
- In any 2-consecutive arrangement of M_Φ, the pair $\{r_{2i-1}, r_{2i}\}$ must be adjacent to the pair $\{r_{2i+1}, r_{2i+2}\}$, for $i = 1, \ldots, n-1$.

So the position of r_{2i-1} immediately above or below r_{2i} encodes an assignment of truth or falsity to the variable v_i, and the pairs of rows encoding the variables are consecutive and in order. But first it is necessary to show how these constraints may be imposed.

Include the five columns $[1,2,3], [1,2,4], [1,2,5], [1,2,6], [1,2,7]$. This is the construction of Lemma 5.1 for $k = 2$, and these columns force rows r_1 and r_2 to be consecutive in any 2-consecutive arrangement of M_Φ.

Hence we may enforce the first of the above constraints using $5n$ columns, applying the above construction for each pair r_{2i-1}, r_{2i}.

Refer to a set of rows which must be adjacent in some order as a *block*. Let block b_i be the pair $\{r_{2i-1}, r_{2i}\}$, for $i = 1, \ldots, n$. So block b_i represents variable v_i.

We may constrain two blocks to be adjacent in much the same way as for individual rows. For example, to make b_1 and b_2 adjacent, include the columns $[1,2,3,4,5], [1,2,3,4,6], [1,2,3,4,7]$, $[1,2,3,4,8], [1,2,3,4,9]$. Applying this construction repeatedly and shifting the ones two rows lower at each repetition, we may use $5(n-1)$ columns to make block b_i adjacent to block b_{i+1}, for $i = 1, \ldots, n-1$, in any permutation that makes M_Φ 2-consecutive. This ensures that the second constraint holds.

Having used two rows for each variable, we now use an additional block of five rows for each clause in Φ. Assume $n \geq 3$ in what follows, and consider the column

$$[2n-4, 2n-1, 2n, 2n+1, 2n+2, 2n+3, 2n+4, 2n+5].$$

Any 2-consecutive arrangement of M_Φ must now leave the five rows $r_{2n+1}, \ldots, r_{2n+5}$ adjacent in some order, and adjacent to block b_n. (This is because r_{2n-4} is not allowed to be adjacent to the other rows containing a 1 in this column.) No other constraints are placed on 2-consecutive permutations. We may then force the next five rows $r_{2n+6}, \ldots, r_{2n+10}$ to be adjacent in some order and for this block to be next to the block $r_{2n+1}, \ldots, r_{2n+5}$. This is accomplished with the column

$$[2n-4, 2n+1, 2n+2, 2n+3, \ldots, 2n+10].$$

Since m is the number of clauses in Φ, we need an additional $m-2$ blocks of five rows, which are obtained using columns of the general form

$$\{[2n-4, 2n+5i-9, 2n+5i-8, 2n+5i-7, \ldots, 2n+5i] \ : \ i=3,\ldots,m\}$$

Let B_j be the block $\{r_{2n+5j+1}, \ldots, r_{2n+5j+5}\}$, $j=0,\ldots,m-1$. B_j corresponds to clause C_j. The idea now is that when any literal in that clause is set to false, this should lead to some restriction on the order in which the five rows in B_j may appear in a 2-consecutive arrangement of M_Φ. In what follows, the "top" position or positions of B_j mean the positions of rows in B_j closest to the blocks $b_1, \ldots, b_n$ representing the values of the boolean variables.

Suppose that C_j contains the literal v_α. Consider the column

$$[2\alpha, 2\alpha+1, \ldots, 2n+5j+1].$$

If the rows in block b_α are switched (setting v_α to false) then to make M_Φ 2-consecutive, row $r_{2n+5j+1}$ must be the top row in B_j. (Any other arrangement of B_j gives this column three contigs.) If C_j contains $\neg v_\alpha$ then this constraint is made conditional on v_α being true by switching the 1 and 0 in b_α, giving column $[2\alpha-1, 2\alpha+1, 2\alpha+2, \ldots, 2n+5j+1]$.

Suppose v_β is another literal in C_j and consider the column

$$[2\beta, 2\beta+1, \ldots, 2n+5j+4].$$

If v_β is false, this column forces $r_{2n+5j+5}$ to be the bottom row of B_j. (If $r_{2n+5j+5}$ is placed anywhere else in B_j, then the zero it contains breaks up the contig above it.)

Suppose that v_γ is the third literal in C_j. Introduce the following two columns:

$$[2\gamma, 2\gamma+1, \ldots, 2n+5j+2],$$

$$[2\gamma, 2\gamma+1, \ldots, 2n+5j+3].$$

If v_γ is false then $r_{2n+5j+1}$ and $r_{2n+5j+2}$ must be in the top two positions in B_j. Also $r_{2n+5j+1}$, $r_{2n+5j+2}$, and $r_{2n+5j+3}$ must be in the top three positions. Consequently $r_{2n+5j+3}$ must be in the middle position. If v_γ is true, we have the following milder constraints: We may not let $r_{2n+5j+1}$ and $r_{2n+5j+2}$ be non-adjacent with neither at the top of B_j (this gives rise to three contigs in the first of these two columns.) Furthermore we may not let $r_{2n+5j+1}$, $r_{2n+5j+2}$, and $r_{2n+5j+3}$ occupy the top, middle and bottom positions, or the second, third/fourth, and bottom positions, since these arrangements give rise to three contigs in the second of the above two columns.

Include the column $[2n+5j+1, 2n+5j+3, 2n+5j+5]$. If all three literals in C_j are false, then the three ones in this column (which occur in B_j) must be separated by zeroes if the rows of M_Φ have been permuted such that the rest of the columns described earlier are 2-consecutive. It remains to show that if any literal in C_j is true, then some arrangement of the rows in B_j leaves this latest column with only two contigs while satisfying the other constraints put on their order.

If v_α is true then rows $r_{2n+5j+1}, \ldots, r_{2n+5j+5}$ may be arranged in the order:

$$r_{2n+5j+2}, r_{2n+5j+1}, r_{2n+5j+3}, r_{2n+5j+4}, r_{2n+5j+5}.$$

If v_β is true then they may be arranged in the order:

$$r_{2n+5j+1}, r_{2n+5j+2}, r_{2n+5j+3}, r_{2n+5j+5}, r_{2n+5j+4}.$$

If v_γ is true then they may be arranged in the order:

$$r_{2n+5j+1}, r_{2n+5j+3}, r_{2n+5j+2}, r_{2n+5j+4}, r_{2n+5j+5}.$$

These sequences (which are from top to bottom) work even if both the other two literals are false.

If we encode all m clauses in this way, using block B_i for clause C_i, this requires $5m$ columns, and to make M_Φ 2-consecutive corresponds to satisfying their conjunction. Hence we have constructed a matrix M_Φ which has the 2-C1P if and only if Φ is satisfiable. ■

Having shown that testing the 2-C1P is NP-complete, we can use the lemma to generalize this result to the k-C1P, for every fixed $k > 2$. This is done by attaching an extra $2k-1$ rows to the top of M_Φ which are constrained to be in order, and extend each column of M_Φ with $2k-1$ alternating zeroes and ones. This adds $k-2$ to the number of contigs in each column.

In reality, sparse matrices are likely to be of more interest to biologists, since typically a clone will only contain a small number of the probes, and there is also only limited coverage of the whole sequence by the clones. Hence it is appropriate to ask how hard it is to test for the k-C1P given a limit l on the number of ones per row and per column. The above reduction can no longer be used, and it remains an open problem whether the 2-C1P is NP-hard for some value of l. Note that it can be shown by similar constructions to the above, that testing for the k-C1P (for any $k \geq 3$) is NP-hard for matrices with a limited number of ones per column (i. e. short clones).

Acknowledgments

We thank Mike Waterman and an anonymous referee for their comments which improved the readability of this manuscript. Paul Goldberg would like to thank Sorin Istrail for bringing to his attention computational problems related to physical mapping, and for many stimulating discussions. His work was carried out at Sandia National Laboratories supported by the U.S. Department of Energy under contract DE-AC04-76DP00789. Ron Shamir's work has been supported in part by a grant from the Israel Ministry of Science and the Arts.

References

[1] F. Alizadeh, R. M. Karp, L. W. Newberg, and D. K. Weisser. Physical mapping of chromosomes: A combinatorial problem in molecular biology. In *Proc. fourth annual ACM-SIAM Symp. on Discrete Algorithms (SODA 93)*, pages 371–381. ACM Press, 1993.

[2] S. Arnborg, D. J. Corneil, and A. Proskurowski. Complexity of finding embedding in a k-tree. *SIAM J. Alg. Disc. Meth.*, 8:227–284, 1987.

[3] R. Arratia, E. S. Lander, S. Tavaré, and M. S. Waterman. Genomic mapping by anchoring random clones: A mathematical analysis. *Genomics*, 11:806–827, 1991.

[4] E. Barillot, J. Dausset, and D. Cohen. Theoretical analysis of a physical mapping strategy using random single-copy landmarks. *PNAS*, 88:3917–3921, 1991.

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

[6] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms. *J. Comput. Sys. Sci.*, 13:335–379, 1976.

[7] E. Branscomb et al. Optimizing restriction fragment fingerprinting methods for ordering large genomic libraries. *Genomics*, 8:351–366, 1990.

[8] D. T. Burke, G. F. Carle, and M. V. Olson. Cloning of large segments of exogenous DNA into yeast by means of artificial chromosome vectors. *Science*, 236:806–812, 1987.

[9] A. V. Carrano. Establishing the order of human chromosome-specific DNA fragments. In A. D. Woodhead and B. J. Barnhart, editors, *Biotechnology and the Human Genome*, pages 37–50. Plenum Press, 1988.

[10] D. Cohen, A. Chumakov, and J. Weissenbach. A first-generation physical map of the human genome. *Nature*, 366:698–701, 1993.

[11] F. Collins and D. Galas. A new five-year plan of the U.S. Human Genome Project. *Science*, 262:43–46, 1 October 1993.

[12] A. Coulson, J. Sulston, S. Brenner, and J. Karn. Toward a physical map of the genome of the nematode *caenorhabditis elegans*. *PNAS*, 83:7821–7825, 1986.

[13] Alister G. Craig et al. Ordering of cosmid clones covering the herpes simplex virus type I (HSV-I) genome: A test case for fingerprinting by hybridization. *NAR*, 18:2653–2660, 1990.

[14] X. Deng, P. Hell, and J. Huang. Linear time representation algorithms for proper circular arc graphs and proper interval graphs. Technical report, School of Computing Science, Simon Fraser University, 1993.

[15] R. G. Downey and M. R. Fellows. Fixed-parameter intractability. In *Proc. Structures 92*, pages 36–49, 1992.

[16] M. R. Fellows, M. T. Hallet, and H. T. Wareham. DNA physical mapping: Three ways difficult. In *Proc. First European Symp. on Algorithms (ESA '93)*, pages 157–168. Springer, 1993. LNCS 726.

[17] M. R. Garey, R. L. Graham, D. S. Johnson, and D. E. Knuth. Complexity results for bandwidth minimization. *SIAM J. Appl. Math.*, 34(3):477–495, 1978.

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

[19] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM J. on Computing*, 5:704–714, 1976.

[20] L. Goldstein and M. S. Waterman. Mapping DNA by stochastic relaxation. *Advances in Applied Mathematics*, 8:194–207, 1987.

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

[22] M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. Technical Report 270-92, Computer Science Dept., Tel Aviv University, 1992. To appear in *Journal of Algorithms*.

[23] M. C. Golumbic, H. Kaplan, and R. Shamir. On the complexity of DNA physical mapping. *Advances in Applied Mathematics*, 15:251–261, 1994.

[24] M. C. Golumbic and R. Shamir. Complexity and algorithms for reasoning about time: A graph-theoretic approach. *J. ACM*, 40:1108–1133, 1993.

[25] J. Gustedt. On the pathwidth of chordal graphs. Technical report, Fachbereich Mathematik, Technische Universität Berlin, 1992. To appear in *Discrete Math.*

[26] W.-L. Hsu. A simple test for interval graphs. Technical report, Inst. of Information Science, Academica Sinica, Taipei, Taiwan, 1992.

[27] H. Kaplan and R. Shamir. Pathwidth, bandwidth and completion problems to proper interval graphs with small cliques. Technical report, CS Department, Tel Aviv University, November 1993. to appear in *SIAM Journal on Computing*.

[28] T. Kashiwabara and T. Fujisawa. An NP-complete problem on interval graphs. In *IEEE Symp. of Circuits and Systems*, pages 82–83. IEEE, 1979.

[29] T. Kashiwabara and T. Fujisawa. NP-completeness of the problem of finding a minimum-clique-number interval graph containing a given graph as a subgraph. In *IEEE Symp. of Circuits and Systems*, pages 657–660. IEEE, 1979.

[30] L. M. Kirousis and C. H. Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47:205–218, 1986.

[31] T. Kloks. *Treewidth.* PhD thesis, Dept. of Computer Science, Utrecht University, 1993.

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

[33] N. Korte and R. H. Möhring. An incremental linear time algorithm for recognizing interval graphs. *SIAM J. Comput.*, 18:68–81, 1989.

[34] L.T. Kou. Polynomial computable consecutive information retrieval problems. *SIAM J. Comput.*, 6:67–75, 1977.

[35] E. S. Lander and M. S. Waterman. Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics*, 2:231–239, 1988.

[36] C. G. Lekkerkerker and J. Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundam. Math.*, 51:45–64, 1962.

[37] R. D. Little et al. Yeast artificial chromosomes spanning 8 megabases and 10-15 centimorgans of human cytogenetics band Xq26. *PNAS*, 89:177–181, 1992.

[38] F. Michiels, A. G. Craig, G. Zehetner, G. P. Smith, and H. Lehrach. Molecular approaches to genome analysis: a strategy for the construction of ordered overlapping clone libraries. *CABIOS*, 3(3):203–210, 1987.

[39] R. H. Möhring. Graph problems related to gate matrix layout and PLA folding. In G. Tinhofer et al., editors, *Computational Graph Theory, Computing Supplement 7*, pages 17–51. Springer, Vienna, 1990.

[40] B. Monien. The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete. *SIAM J. Algeb. Discr. Meth.*, 7:505–512, 1986.

[41] B. Monien and I. H. Sudborough. Min cut is NP-complete for edge weighted trees. *Theoretical Computer Science*, 58:209–229, 1988.

[42] R. Nagaraja. Current approaches to long-range physical mapping of the human genome. In R. Anand, editor, *Techniques for the Analysis of Complex Genomes*, pages 1–18. Academic Press, London, 1992.

[43] M. V. Olson et al. Random-clone strategy for genomic restriction mapping in yeast. *Proc. Nat. Acad. Sci. USA*, 83:7826–7830, 1986.

[44] M. V. Olson, L. Hood, C. Cantor, and D. Botstein. A common language for physical mapping of the human genome. *Science*, 234:1434–1435, 1985.

[45] J. Opatrny. Total ordering problems. *SIAM J. Computing*, 8(1):111–114, 1979.

[46] N. Sternberg. Bacteriophage P1 cloning system for the isolation, amplification and recovery of DNA fragments as large as 100 kilobase pairs. *PNAS*, 87:103–107, 1990.

[47] U. S. Congress. Mapping our genes–the genome projects, how big, how fast? Technical Report OTA-BA-373, Office of Technology Assessment, Washington, D.C., 1988.

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

[49] J.D. Watson, M. Gilman, J. Witkowski, and M. Zoller. *Recombinant DNA*. W.H. Freeman, New York, 2nd edition, 1992.

[50] D. B. West and D. B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discrete Applied Math.*, 8:295–305, 1984.

[51] M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2, 1981.