Algorithms for the Extraction of Synteny Blocks from Comparative Maps

Vicky Choi[1], Chunfang Zheng[2], Qian Zhu[2], and David Sankoff[2]

[1] Department of Computer Science, Virginia Tech, Blacksburg, VA 24061
vchoi@cs.vt.edu
[2] Departments of Biology, Biochemistry, and Mathematics and Statistics, University of Ottawa, Ottawa, Canada K1N 6N5
{czhen033,qzhu012,sankoff}@uottawa.ca

Abstract. In comparing genomic maps, we try to distinguish mapping errors and incorrectly resolved paralogies from genuine rearrangements of the genomes. This can be formulated as a Maximum Weight Independent Set (MWIS) search, where vertices are potential strips of markers syntenic on both genomes, and edges join conflicting strips, in order to extract the subset of compatible strips that accounts for the largest proportion of the data. This technique is computationally hard. We introduce biologically meaningful constraints on the strips, reducing the number of vertices for the MWIS analysis and provoking a decomposition of the graph into more tractable components. New improvements to existing MWIS algorithms greatly improve running time, especially when the strip conflicts define an interval graph structure. A validation of solutions through genome rearrangement analysis enables us to identify the most realistic solution. We apply this to the comparison of the rice and sorghum genomes.

1 Introduction

Comparing two genomic maps containing orthologous sets of markers induces a decomposition of the genomes into synteny blocks, segments of chromosomes containing orthologous markers in the same or reverse order in the two genomes. The blocks may be differently grouped into chromosomes, and differently ordered and oriented, in the two genomes being compared.

In the course of genomic evolution, as more and more rearrangements intervene since the common ancestor, the synteny blocks in common between the two genomes become more fragmented, i.e., shorter, and eventually contain only one marker, or none.

The construction of the synteny blocks based on traditional comparative maps is different in both spirit and technique from the analogous problem based on genome sequences, and is very vulnerable to errors and ambiguities in the position of the markers on a map, depending on the specific mapping technology. Another kind of problem involves ambiguous homology, leading to the risk of matching up inappropriate pairs of markers as orthologs in the two genomes. These problems tend to artifactually increase the number of synteny blocks induced by the comparison, disrupting true synteny blocks by artifactual blocks containing only one or two markers.

Thus, when many rearrangements have intervened since the common ancestor, or where the sampling density of markers on the chromosome is sparse, it may be unclear

R. Giancarlo and S. Hannenhalli (Eds.): WABI 2007, LNBI 4645, pp. 277–288, 2007.

whether any particular one of the increasing number of short synteny blocks is due to error or to rearrangement. These considerations suggest the principle that inferences that depend on the position of a single marker should not be given as much weight as inferences that are supported by more markers. We would thus like to construct a set of synteny blocks that are conflict-free, contain as much of the data as possible, and are credible from a genome rearrangement viewpoint.

In [9], we proposed the following strategy: first, construct a set of *pre-strips*, which are certain short common subsequences of one chromosome from each genome; second, extract from this set a subset of mutually compatible (non-intersecting) containing a maximum number of markers; third, add to this subset any markers that do not increase the rearrangement distance [7] between the genomes; fourth, assemble the synteny blocks from the markers in the solution.

This approach encountered a bottleneck at the second step, formulated in terms of a solution for the NP-hard maximum weight clique (MWC) problem in a graph representing pre-strip compatibilities. It was not feasible to run the whole data set using available algorithms. Thus we devised biologically-motivated constraints to reduce the data set and were then able to run moderate size instances.

In this paper, our main contributions are: first, based on a key combinatorial observation, the establishment of constraints on the set of pre-strips that are necessary to a solution, thus reducing the amount of data that must be input to MWIS without losing optimality (Section 3), and second, the design of a new algorithm for the maximum weight independent set (MWIS) problem[1], specifically motivated by the nature of pre-strip data (Section 4.1). Finally, taking advantage of the source of the incompatibilities in the chromosome-based data, we propose a natural decomposition of the graph which allows us to solve relatively large instances of the problem extremely efficiently – 1 to 2 seconds on a Pentium IV computer for instances that took days or that proved infeasible with the previous techniques. As a prerequisite to this material, in Section 2, we review the definition of strips and pre-strips, as well as a polynomial-time algorithm for generating all pre-strips. And after the theoretical development, we discuss the question of restoring additional markers to the solution in Section 6 and analyze the rice and sorghum comparative map in Section 7.

2 Problem and Terminology: Strips, Pre-strips, Pure Strips

Let n be the number of markers in common in two genomes with χ_1 and χ_2 chromosomes. In one genome, number all these markers on any one of the chromosomes from left to right in increasing order starting with marker 1. Continue the numbering sequence on a second chromosome and so on, until finishing with the n-th marker on the χ_1-st chromosome. Then each marker in the second genome receives the same label as its supposed ortholog in the first genome.

We recall the definition of *strips, pre-strips* and *pure strips* in [9]. Consider any $l \geq 2$ consecutive contiguous markers on a chromosome in one genome. If the same l markers are consecutive on a chromosome in the other genome, with the same (or

[1] Equivalent to the MWC formulation in the complementary graph in [9].

reverse) order and with each marker having the same (or opposite) orientation[2] in both genomes, they constitute a *forward strip* (*reverse strip*) of *length* l. Note that many or most of the markers in a comparative map may not be in any strip. The synteny blocks in the decomposition of the two genomes we are looking for are all strips, but many of these blocks will not be visible in the original data since they are disrupted by erroneously mapped markers and mistaken orthologs, so we have to construct them by discarding the markers disrupting their contiguity property.

ORIGINAL		**REDUCED**	
Genome 1	**Genome 2**	**Genome 1**	**Genome 2**
abcdef	lbcdpz	abcd	lbcdz
lmnoprq	-x-q-o-m	lmoq	-q-o-m
wxyz	we-fry	wyz	wy
	na		a
Pre-strips	**Pure strip**	**Strips**	**Singletons not**
bcd,bc,cd,	bcd	bcd,moq,wy	**in pre-strips**
moq,mo,oq,			**but compatible**
wy,lp			a,l,z
Common subsequences		**Discarded as noise**	
not pre-strips		e,f,n,p,r,x	
bd,mq			

Fig. 1. Strips and pre-strips. "-" indicates different orientation markers in two genomes.

Maximal Strip Recovery (MSR) problem: *Given two genomes as described above, discard some subset of the markers, leaving only markers in disjoint strips* $S_1, \cdots, S_r$ *of lengths* $w_1, \cdots, w_r$*, respectively, in the genomes thus reduced, such that* $\sum_{i=1}^{r} w_i$ *is maximized.* The MSR problem corresponds to our previously stated goal of constructing a set of compatible strips containing as much of the data as possible.

We will search for *pre-strips* in the two genomes, relying on the subsequent analyses to eliminate the disrupting markers and thus reveal the "underlying" strips. This is illustrated in Fig. 1.A pre-strip P is a common subsequence, or a reverse common subsequence, of the markers on the two chromosomes, such that there is no other marker of appropriate orientation on both chromosomes that is between two successive markers in P. For example, if AB is a pre-strip, then there does not exist C such that ACB is a pre-strip. For all reverse pre-strips, this is indicated by minus signs on the markers involved in the second genome only. =Notice that a pre-strip satisfies the same definition as a strip, except that the markers need not be contiguous. A pre-strip that is a strip in the original genome data, and is not contained in another strip, is called a *pure strip*. Remark: *Strips* are defined relative to the current state of the two genomes, either before, during or after reducing their size, but *pre-strips* and *pure strips* are defined in terms of the original genome data only.

In [9] it is shown that every pre-strip P has a unique representation as a string of p's and 1's, where a p represents a pure strip and a 1, called a *singleton*, represents a marker not in a pure strip. Moreover,

[2] Reading direction, DNA strand.

Proposition 1. *Any pre-strip can be uniquely represented by a sequence of terms of form* $p, 11, 1p, p1, 111$ *and* $1p1$.

Proposition 2. *All possible strips that can be formed by the deletion of markers from two genomes, and that can be part of a solution to the MSR problem, are pre-strips of these genomes.*

Consequently, it suffices to consider only pre-strips of the forms mentioned in Proposition 1. All such pre-strips can be calculated by an algorithm requiring $O(n^4)$ time in the worst case. In practice, the running time is far less. In the following, we show that we can further reduce the set of pre-strips to be considered and define the conflict graph.

3 Data Reduction and the Conflict Graph

We say two pre-strips P and Q are *in conflict* if they share at least one marker or if one pre-strip, say P, contains a marker between two successive markers, in either genome, in the other pre-strip, Q. Otherwise P and Q are compatible.

Let k-pure-strip denote a pure strip of length k. Then we have (proof omitted):

Lemma 1. *All* k*-pure-strips with* $k \geq 4$ *are part of a solution to the MSR problem.*

Corollary 1. *Pre-strips that are in conflict with a* k*-pure-strip for* $k \geq 4$ *are not included in the solution to the MSR problem.*

In fact, we can eliminate further pre-strips (proof omitted) :

Corollary 2. *Pre-strips of the form* $1p$, $p1$, 111, $1p1$ *that are in conflict with a* k*-pure-strip for* $k \geq 3$*, are not included in the solution to the MSR problem.*

By these two reductions (Corollaries 1 and 2), we can generate pre-strips more efficiently, namely generate them "on the fly" with the k-pure-strips acting as "terminators". The corollaries also imply that we need not treat any marker in k-pure-strips for $k \geq 3$ as a singleton in a reverse pre-strip. At most one marker in each 2-pure-strip need be considered as a singleton in a reverse pre-strip.

We define the *conflict* graph $G = (V, E)$, where V consists of all pre-strips after reduction, and E consists of all pairs of conflicting pre-strips. The conflict graph is the complement of the compatibility graph defined in [9], but it has an important interval graph-related property (cf. Section 5 below.)

Graph theory terminology and notation. Let $G = (V, E)$ be a simple undirected graph. For $v \in V$, the set of neighbors of v in G is denoted by $\mathsf{nbr}(v) = \{u \in V : uv \in E\}$. For $S \subset V$, $G[S] = (S, E_S)$ is called a (vertex) *induced* subgraph of G, where $E_S = \{uv \in E : u, v \in S\}$. The complement of $G = (V, E)$ is denoted by $\bar{G} = (V, \bar{E})$, where $\bar{E} = \{uv : u \neq v \in V, uv \notin E\}$.

For $S \subseteq V$, S is called an *independent set* if for $u, v \in S$, $uv \notin E$; S is called a *clique* if $u, v \in S$, $uv \in E$; S is a *vertex cover* of G if for $uv \in E$, either u or v is in S.

A *linear ordering* of $G = (V, E)$ with $|V| = n$ is a bijection $\phi : V \longrightarrow [n] = \{1, \ldots, n\}$. When ϕ is understood, we denote $\phi^{-1}(i)$ by v_i and write $V_i =$

$\{v_1, v_2, \ldots, v_i\}$, for $i = 1, \ldots, n$. For $1 \leq i \leq n$, we define the right neighbors of v_i to be $\mathsf{rnbr}(v_i) = \{v_j \in V : v_i v_j \in E, j < i\}$.

We consider the vertex-weighted graph, where the weight of vertices is given by a function $\mathsf{w} : V \longrightarrow Z^+$. For $S \subseteq V$, the weight of S, $\mathsf{w}(S) = \sum_{v \in S} \mathsf{w}(v)$.

MWIS problem: *Given a vertex-weighted graph $G = (V, E)$ with the weight function $\mathsf{w} : V \longrightarrow Z^+$, find an independent set S of G such that $\mathsf{w}(S)$ is maximized. We denote the optimum independent set by* $\mathrm{mis}(G)$.

3.1 Reformulation as Maximum Weight Independent Sets (MWIS)

By propositions from [9], the MSR problem (Section 2) is just the maximum weight independent set (MWIS) problem on the conflict graph G where the weight of each vertex is the number of markers in the corresponding pre-strip.

Proposition 3. *Given any set C of pairwise compatible pre-strips. Consider the reduced genomes produced by deleting all markers that are in none of the pre-strips in C. In these reduced genomes all of the markers in each pre-strip in C appear as strips. The number of markers in each strip is the same as in the corresponding pre-strip.*

Proposition 4. *The solution* $\mathrm{mis}(G)$ *of the MWIS problem on G induces a reduction of the original genomes so that they are composed completely of disjoint strips and so that the total strip score is maximized.*

It is well-known that the MWIS problem, equivalent to the Maximum Weight Clique (MWC) problem and the Minimum Weight Vertex Cover (MWVC) problem, is NP-hard. Exact algorithms and heuristics have been developed for these problems. The most recent MWC algorithm is due to Kumlander [3], itself a minor improvement of Ostergard's [5,6] algorithm.

4 Maximum Weight Independent Sets (MWIS)

In the following, we will first describe a linear-time algorithm for MWIS problem on interval graphs. We then describe improvements on one of the best exact algorithms – Ostergard's algorithm[5,6] – for MWIS problem on general graphs. Our improvement consists of (1) better upper and lower bounds (for pruning the search tree); (2) the ordering of the vertices. In particular, we give a characteristic of a good ordering, partially answering an open problem in [5,6].

Suppose G is linearly ordered, $V = \{v_1, \ldots, v_n\}$. As in [5,6], we consider the induced subgraph incrementally, $G[V_1]$, $G[V_2], \ldots, G[V_n]$. Recall that $\mathrm{mis}(G)$ is an maximum-weight independent set of G. Define $s_i = \mathsf{w}(\mathrm{mis}(G[V_i]))$, for $i = 1, \ldots, n$. Thus, we have $s_1 = \mathsf{w}(v_1)$ and $s_n = \mathsf{w}(\mathrm{mis}(G))$, the weight of the maximum independent set sought.

It is easy to see that $s_{i-1} \leq s_i \leq s_{i-1} + \mathsf{w}(v_i)$. If $v_i \notin \mathrm{mis}(G[V_i])$, then $s_i = s_{i-1}$. If $v_i \in \mathrm{mis}(G[V_i])$, then by definition, $s_i = \mathsf{w}(v_i) + \mathsf{w}(\mathrm{mis}(G[V_{i-1} \setminus \mathsf{rnbr}(v_i)]))$. Denote this quantity by s_i^1. Hence, to compute s_i incrementally, we compute s_i^1 and compare it with s_{i-1}, and set s_i to be the larger of the two.

In the following, we first recall the definition of interval graphs. Then we will describe a linear-time algorithm for MWIS problems on interval graphs, which motivated our improved algorithm for MWIS problems on general graphs.

4.1 Linear-Time Algorithm for MWIS on Interval Graphs

Definition 1. *A graph $G = (V, E)$ is an interval graph if and only if it admits an interval graph realization: there exists a set of intervals such that there is a one-one correspondence between each vertex and each interval and there is an edge between two vertices if and only if their corresponding intervals overlap.*

Theorem 1. *A graph $G = (V, E)$ is an interval graph if and only if there is a linear ordering of G such that the right neighbors of each vertex are consecutive: i.e., there is a ordering $V = \{v_1, v_2, \ldots, v_n\}$ such that for $i > j > k$, if $v_i v_k \in E$, then $v_i v_j \in E$.*

The ordering of G is called an *I-ordering* [1]. An I-ordering can be obtained in linear time (e.g., by the 5-SWEEP LBFS algorithm [2]). Note that for an I-ordering, $\mathsf{rnbr}(v_i) = \{v_{i-1}, v_{i-2}, \ldots, v_{i-t}\}$ for some $t \geq 0$ and $V_{i-1} \setminus \mathsf{rnbr}(v_i) = \{v_1, v_2, \ldots, v_{i-t-1}\}$. Thus we have

$$s_i = \max \begin{cases} s_{i-1} & \text{if } v_i \notin \mathrm{mis}(G[V_i]) \\ \mathsf{w}(v_i) + s_{i-t-1} & \text{if } v_i \in \mathrm{mis}(G[V_i]) \end{cases}$$

Thus one can easily get the linear time $O(|V| + |E|)$ time algorithm for the MWIS problem on interval graphs (compared with $O(|V|^2)$ time in [4]).

4.2 Improved Algorithm for MWIS Problem on the General Graph

Our algorithm improves upon the algorithm by Ostergard [5,6], which is a branch-and-bound algorithm. Branch-and-bound algorithms build a search tree, which associates to each node a current partial solution set, and the remaining working set. Critical to the basic branch-and-bound algorithm for MWC/MWIS problem are a good ordering of the vertices, a good lower bound on the size of maximum independent set of the graph, and a good upper bound on the size of independent set of the induced subgraph on the working set. For example, colouring the vertices (the graph and its complement) can be used to obtain both upper and lower bounds. (Indeed, Kumlander's minor improvement over Ostergard's algorithm is on the efficient computing upper bounds on the working set based on a greedy colouring of the entire graph.) Ostergard resolves the tight lower bound problem by incrementally computing the MWIS of the graph. This method actually gives a best possible bound that, once attained, terminates the search. The motivation for the incremental method, however, is to get a tighter upper bound, namely s_i, where i is the maximum of the remaining working set.

The key to our algorithm is the observation that with no extra work, we can get a better upper bound by dividing the working set into two parts: a disruption list and a consecutive prefix. Recall that if $v_i \in \mathrm{mis}(G[V_i])$, then $s_i = \mathsf{w}(v_i) + \mathsf{w}(\mathrm{mis}(G[V_{i-1} \setminus \mathsf{rnbr}(v_i)]))$. In general, we have $V_{i-1} \setminus \mathsf{rnbr}(v_i) = \{d_{i_1}, \ldots, d_{i_s}\} \cup V_{i_t}$, as shown in Figure 2. We call $D_i = \{d_{i_1}, \ldots, d_{i_s}\}$ the *disruption list* of v_i, and V_{i_t} the consecutive prefix. (If G is an interval graph, we can order the vertices such that $D_i = \emptyset$.)

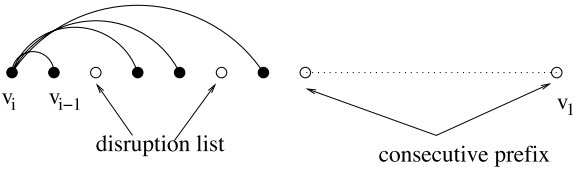

Fig. 2. v_i is adjacent to the black vertices

Thus, a tighter upper bound can be obtained by the upper bound of the disruption list and the exact solution of consecutive prefix. And we only need to branch on the disruption list, in contrast to the entire working set in [5,6]. Further, by partitioning the working set into two parts, we can get a good lower bound of the working set (which Ostergard's algorithm does not have), namely a lower bound by the exact solution of the consecutive prefix.

5 Union of Interval Graphs; Decomposition of the Conflict Graph

5.1 Union of Interval Graphs

Recall that in the conflict graph $G = (V, E)$, each vertex $v \in V$ corresponds to each pre-strip $P(v)$. For $u, v \in V$, $uv \in E$ if and only if $P(v)$ and $P(u)$ are in conflict. Recall that a pre-strip corresponds to two copies of a subsequence of markers, one copy from a chromosome in each genome. We say two pre-strips $P(u)$ and $P(v)$ conflict in genome 1 (resp. 2) if their copies in genome 1 (resp. 2) conflict. By definition, $P(u)$ and $P(v)$ conflict if they conflict in genome 1 or genome 2 or both. For $i = 1, 2$, let $E_i = \{uv : P(u)$ and $P(v)$ conflict in genome $i, u, v \in V\}$. Then we have $E = E_1 \cup E_2$.

Further, according to the observation before Section 4.1, a good ordering will have $|D_i|$ as small as possible; a possible objective function for an ordering is $\sum_i |D_i|$. If G is an interval graph, we can find an ordering such that $\sum_i |D_i| = 0$. The 5-sweep LBFS algorithm gives a good ordering if the graph is "close" to an interval graph.

Since the chromosomes are linear, when considering only one genome, each pre-strip corresponds to an interval of a line (chromosome) and two pre-strips are conflict if and only if their corresponding intervals overlap. Therefore, $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are interval graphs. However, $G = (V, E) = (V, E_1 \cup E_2)$ is not necessarily interval, e.g., a four-cycle can be formed $v_1v_2, v_2v_3, v_3v_4 \in E_1$ and $v_1v_4 \in E_2$. In fact, the graph G is in general not an interval graph.

In our experience [1], if the graph is only locally distorted, the ordering so produced by 5-SWEEP LBFS algorithm will also be distorted locally. Namely, the ordering only failed for local forbidden subgraph region, i.e. the vertices before and after the forbidden subgraph satisfy the right neighborhood consecutive property. In other words, if our graph is only locally distorted, then the ordering produced by 5-SWEEP LBFS algorithm will be a good ordering for our MWIS algorithm. Motivated by this observation and together with the ideas in [9], we propose the following natural decomposition of our conflict graph.

5.2 Natural Decomposition

Our idea here is to find a small subset of vertices, called *separators*, such that the removal of these vertices results in a set of computationally tractable connected components (here locally distorted interval components), and the solution to this set of connected components has good properties, either being a good approximation to the optimal solution of the original problem or having biologically desirable properties. The separator vertices ideally correspond to errors in the original genomic maps.

Recall that a pre-strip consists of a common subsequence of markers in two genomes. Two markers in a pre-strip may well be located far from each other in a genome. Note that the larger the gap a pre-strip has, the more pre-strips it can conflict with, because conflicts occur when a marker from certain other pre-strips fall in the gap.

Thus, a computationally and biologically well-motivated way to approximate the MWIS solution will be to remove those pre-strips with the largest gaps. That is, choose those large-gap pre-strips as separators. One would then expect that the non-interval components will only be locally distorted due to the gap constraints. Indeed, in the example to be discussed in Section 7, if we remove all pre-strips with gap > 4, then the graph is decomposed into 36 components, with all but one components being interval. For the only non-interval component, there is one I-critical vertex [1], that is, the component becomes interval when the vertex is removed.

As the gap increases, the number of connected components decreases and the total vertices in non-interval connected components increases. Neverthless, even when we retain pre-strips with rather larger gap size, these components are only locally non-interval, and our algorithm based on the LBFS ordering is still very efficient (within one to two seconds). See Table 1 for the statistics.

Another way to choose separators is to exclude pre-strips containing only two markers separated by a gap of any non-zero size. Biologically speaking, such a strip is the weakest kind of evidence for a synteny block other than singletons (markers in no pre-strips, which are never even considered in the MWIS input).

6 Restoration of Markers

The MWIS solution is incompatible with any pre-strip not in it, but it is not necessarily incompatible with all parts of such a pre-strip. For example, it is possible that some pre-strip of form $p1$ is not in the solution, but the singleton element in this pre-strip does not intervene between any two successive markers of a pre-strip in the solution, and may thus be considered compatible. In addition, singleton markers, in no pre-strip, which play a role neither in the input nor the output of the MWIS, could similarly be compatible with the solution.

Since there is no way of identifying, in real data, exactly which markers excluded from the MWIS solution are valid evidence of evolutionary relatedness or divergence of the two genomes, and which are simply erroneous, we have recourse to genome rearrangement analysis. First we use the strips output by the MWIS to calculate the genomic distance between the two genomes [7]. If we were to add a new marker at random (“noise”) to both genomes, this would generally increase the distance by 1 or 2, even if it were compatible with all the strips in the solution. Thus, if we add a

marker from among those not in the MWIS, and this does not increase the distance, this means that when one genome is optimally transformed into the other, the new marker falls naturally into place with no extra effort and is fully consistent with the inferred evolutionary history of all the markers in the solution.

7 A Comparison of the Rice and Sorghum Genomes

We compare maps of the rice and sorghum genomes. The construction of the data set, based on resources in [8] is described in [9].

In this comparison, the database reports 567 correspondences between the two genomes, involving $n_1 = 481$ rice markers and $n_2 = 567$ sorghum markers. The number of distinct markers in common was $n = 481$. A total of 69 of these were present in two or more copies in the sorghum data, with a maximum gene family size of 6. The inclusion of paralogous genes is shown in [9] to create no problems for the biological interpretation of the analysis, to require only slight modifications of the definitions in Section 2 and to affect the computation simply by increasing the number of pre-strips.

Our algorithm for generating pre-strips produced 1853 pre-strips to enter as vertices into the MWIS routine, which exceeded the capability of our algorithm and, indeed, state-of-the art MWC programs that were tried on it.

The results of the analysis on the data reduced by the techniques of Sections 3 and 5 are shown in Table 1. The first thing to note is that even after all possible compatible markers, consistent with the output rearrangement distance, are restored, only 292-324 are present, meaning that 157-189 were discarded, of the maximum possible represented by $n_1 = 481$. This illustrates the importance of analyzing the marker data to remove errors and conflicts.

Another observation is the slight increase in the number of markers in the output, as the gap size criterion is relaxed from gap < 2 to gap < 9, despite the great increase in the number of pre-strips. Thus the extra pre-strips proved to be largely redundant.

Table 1. Pre-strip inclusion criteria and solution characteristics. Strips out, total markers (including restored markers) and distance are averages over ten solutions. Comps/non-int refers to the numbers of components in the MWIS and vertices not in interval set components.

	11's included						11's excluded					
gap $<$	pre-strps	reduced	comps/non-int	strps out.	total mark.	distance	pre-strps	reduced	comps/non-int	strps out.	total mark.	distance
9	894	739	12/542	127	324	72	616	565	20/275	100	306	52
8	836	700	15/505	126	322	72	577	533	24/203	100	306	52
7	771	654	20/385	124	321	67	529	492	26/149	98	306	53
6	709	608	23/296	125	321	69	484	451	26/129	96	302	51
5	646	567	28/170	126	323	70	449	424	29/88	98	304	51
4	550	503	44/80	126	322	70	382	368	40/46	97	303	51
3	432	410	53/77	124	320	69	302	293	46/45	96	303	52
2	259	255	79/0	115	318	67	183	180	68/0	91	292	51

Finally we note the great drop in the genomic distance (typically 18 out of 70) as the "11's excluded" constraint is added. True, this comes at the cost of losing about 18 markers from the output, but the fact that the distance saved is about equal to the number of markers lost suggests that these markers, coming largely from isolated "11" pre-strips (i.e., that could not be incorporated in $p11$ or $11p$ pre-strips), do not carry authentic evolutionary information, by the same arguments about noise as in Section 6.

Our MWIS program for each instance in Table 1 (actually for "11" excluded, gap was up to 15, data not shown) took less than two seconds in a Pentium IV 3.0GHz computer with 2G memory running under Fedora 2 linux OS. (Our previous program can only run on data with gap < 3 for "11" included, and gap < 4 for "11" excluded.)

In Fig. 3, we show the result of applying our method to the sorghum and rice comparative maps, with gap < 15, excluding "11" pre-strips. The confusing pattern of alternating markers from many rice chromosomes on each sorghum chromosome is replaced by a more credible set of long strips.

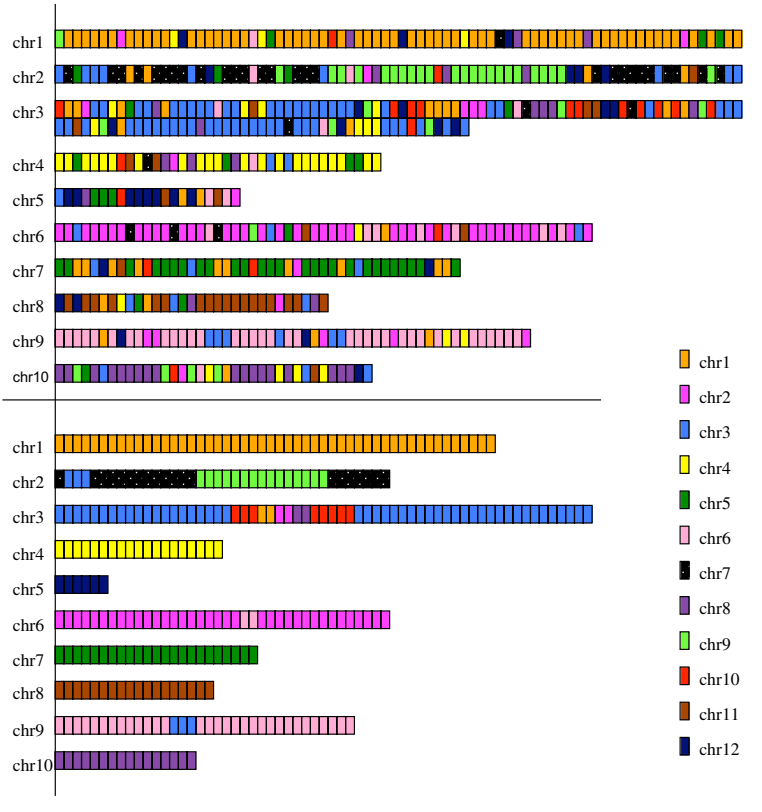

Fig. 3. Top: 567 markers on sorghum chromosomes, colour-keyed by the rice chromosome containing their homologs. Bottom: 303 compatible markers remaining in optimal set of compatible strips. Note that long regions of a single colour generally consist of several synteny blocks whose order and orientation differ from one genome to the other.

Algorithm 1. An improved algorithm for the MWIS problem

Input: $G = (V, E)$, w
Output: w($IS(G)$)
compute an ordering of $V =< v_1, v_2, \ldots, v_n >$, (default: 5-SWEEP LBFS ordering);
best-so-far = $s_1 =$ w(v_1);
for $i = 2$ *to* n **do**
  compute disruption-list and consecutive-prefix if v_i is included in the solution set;
  best-possible $= s_{i-1} +$ w(v_i);
  found =false;
  MWIS-BRANCH-AND-BOUND (w(v_i), disruption-list, consecutive-prefix, best-so-far, best-possible, found);
  s_i=best-so-far ;
return s_n;

MWIS-BRANCH-AND-BOUND (current-weight, disruption-list, consecutive-prefix, best-so-far, best-possible, found)
if disruption-list *is empty* **then**
  if current-weight + *weight of mis(*consecutive-prefix*)* $>$ best-so-far **then**
    best-so-far = current-weight + *weight of mis*(consecutive-prefix);
    if best-so-far = best-possible **then**
      found =true;
  return ;
if current-weight + *upper bound of* disruption-list + *weight of mis(*consecutive-prefix*)* $\leq$ best-so-far **then**
  return; `/* prune the search tree */`;
else if current-weight + *weight of mis(*consecutive-prefix*)* = best-possible **then**
  best-so-far =best-possible ;
  found =true;
  `/* found the best possible, terminate the search */`;
  return ;
while disruption-list *is not empty* **do**
  $d =$dequeue(disruption-list);
  new-current-weight = current-weight + w(d);
  compute new-disruption-list and new-consecutive-prefix if d is included in the solution set;
  MWIS-BRANCH-AND-BOUND(new-current-weight, new-disruption-list, new-consecutive-prefix, best-so-far, best-possible, found);
  if found =*true* **then**
    return ;
return;

8 Conclusion

We have studied the conversion of the MSR problem to the MWIS problem, based on the elimination of as few markers as possible from the genomes being compared.

We have improved the preparation of conflict graph input to the MWIS by proving many cases of the six types of small pre-strip need not be considered. Our main result is an improved algorithm for the general MWIS problem that has superior performance

where the data is "close" to an interval graph structure, precisely the type of data that pre-strip conflicts generate. Our implementation of this new algorithm easily handles data sets with 700 vertices and more, realistic values for available comparative maps.

Our analysis of the rice-sorghum comparison, by comparing the trade-off between loss of markers versus inflation of genomic distance, confirms that fully 37% of the common markers cannot be confidently assigned to synteny blocks, in the sense that either such blocks would conflict with larger blocks already in the solution, or else the inclusion of each of the markers would require an extra rearrangement event to account for its presence, which is exactly the effect expected from a randomly placed marker.

The extent to which our method "cleans up" the comparative map is rather drastic, and we have probably excluded many correctly mapped markers, but their inclusion could not be justified on the basis of the present inventory of common markers.

The fact that the map produced by our method shows evidence of a rather small number of translocations between chromosomes, certainly less than 10, suggests that inversion (more than 40 events) is the dominant rearrangement process in the evolution of these cereals.

References

1. Choi, V.: BARNACLE: An assembly algorithm for clone-based sequences of whole genomes. Ph.D dissertation, Rutgers University (2002)
2. Corneil, D.G., Olariu, S., Stewart, L.: The LBFS structure and recognition of interval graphs. ms. cf: The ultimate interval graph recognition algorithm? In: SODA 1998, pp. 175–180 (2006)
3. Kumlander, D.: A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search. ms. and poster. In: 4th European Congress of Mathematics (2005)
4. Liang, Y.D., Dhall, S.K., Lakshmivarahan, S.: On the problem of finding all maximum weight independent sets in interval and circular-arc graphs. IEEE Symposium on Applied Computing, 465–470 (1991)
5. Ostergard, P.R.J.: A new algorithm for the maximum-weight clique problem. Nordic Journal of Computing 8, 424–436 (2001)
6. Ostergard, P.R.J.: A fast algorithm for the maximum clique problem. Discrete Applied Mathematics 120, 195–205 (2002)
7. Tesler, G.: Efficient algorithms for multichromosomal genome rearrangements. Journal of Computer and System Sciences 65, 587–609 (2002)
8. Ware, D., Jaiswal, P., Ni, J., et al.: Gramene: a resource for comparative grass genomics. Nucleic Acids Research 30, 103–105 (2002)
9. Zheng, C., Zhu, Q., Sankoff, D.: Removing noise and ambiguities from comparative maps in rearrangement analysis. Transactions on Computational Biology and Bioinformatics (forthcoming, 2007), doi:10.1109/TCBB.2007.1075