On the Tractability of Maximal Strip Recovery

Lusheng Wang[*] Binhai Zhu[†]

September 10, 2009

Abstract

Given two genomic maps G and H represented by a sequence of n gene markers, a *strip* (syntenic block) is a sequence of distinct markers of length at least two which appear as subsequences in the input maps, either directly or in reversed and negated form. The problem *Maximal Strip Recovery* (MSR) is to find two subsequences G' and H' of G and H, respectively, such that the total length of disjoint strips in G' and H' is maximized (or, conversely, the number of markers hence deleted, is minimized). Previously, several heuristic algorithms which work well in practice, have been proposed. Theoretically, a factor-4 polynomial-time approximation is known for the MSR problem. Moreover, several close variants of MSR, MSR-d (with $d > 2$ input maps), MSR-DU (with marker duplications) and MSR-WT (with markers weighted) have been proved to be NP-complete. Before this work, the complexity of the original MSR problem was left open. In this paper, we solve the open problem by showing that both MSR and its complement (minimization) version are NP-complete, using a polynomial time reduction from One-in-Three 3SAT. We also present some fixed-parameter tractable algorithms for the (complement of) MSR problem and its variants. Let k be the minimum number of markers deleted in an optimal solution. The running times of our algorithms are $O(2^{3.61k}n + n^2)$ for MSR, $O(\begin{pmatrix} (2d+1)k \\ k \end{pmatrix} dn + dn^2)$ for MSR-d, and $O(2^{7.22k}n + n^2)$ for MSR-DU respectively.

Key words: comparative genomics, maximal strip recovery, syntenic blocks, NP-completeness, FPT algorithms.

[*] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. Email: lwang@cs.cityu.edu.hk.

[†] Corresponding author. Department of Computer Science, Montana State University, Bozeman, MT 59717, USA. Email: bhz@cs.montana.edu.

1 Introduction

A well-known problem in comparative genomics is to decompose two given genomes into syntenic blocks—segments of chromosomes which are deemed to be homologous in the two input genomes. Many methods have been proposed, but they are very vulnerable to ambiguities and errors. Recently, two heuristic methods were proposed to eliminate noise and ambiguities in genomic maps, through handling a problem called Maximal Strip Recovery (MSR — see below for the formal definition) (Choi et al., 2007; Zheng et al., 2007). Chen et al. (2008) proposed a factor-4 polynomial-time approximation algorithm for the problem, and several close variants of the problem were shown to be intractable. It was left as an open problem whether the problem can be solved in polynomial time or is NP-complete. We first review some definitions.

A genomic map is represented by a sequence of gene markers, and a gene marker can appear in several different genomic maps, in either positive or negative form. A *strip* (syntenic block) is a sequence of distinct markers that appears as subsequences in two or more maps, either directly or in reversed and negated form. Given two genomic maps G and H, the problem *Maximal Strip Recovery* (MSR) (Choi et al., 2007; Zheng et al., 2007) is to find two subsequences G' and H' of G and H, respectively, such that the total length of disjoint strips in G' and H' is maximized. Intuitively, those gene markers not included in G' and H' are noise and ambiguities.

We give a precise formulation of the generalized problem MSR-d: Given d signed permutations (genomic maps) G_i of $\langle 1, \ldots, n\rangle$, $1 \leq i \leq d$, find q sequences (strips) S_j of length at least two, and find d signed permutations π_i of $\langle 1, \ldots, q\rangle$, such that each sequence $G'_i = S_{\pi_i(1)} \ldots S_{\pi_i(q)}$ (here S_{-j} denotes the reversed and negated sequence of S_j) is a subsequence of G_i, and the total length of the strips S_j is maximized. Note that the problem Maximal Strip Recovery (MSR) (Choi et al., 2007; Zheng et al., 2007) corresponds to the problem MSR-2 in our new formulation. We refer to Fig. 1 for an example. In this example, each integer represents a marker.

Two heuristic methods based on Maximum Clique and its complement Maximum Independent Set were previously presented for the problem MSR (MSR-2) (Choi et al., 2007; Zheng et al., 2007), which do not guarantee finding the optimal solution but seem to work well for practical datasets. It was shown that these heuristic methods (Choi et al., 2007; Zheng et al., 2007) can be modified to achieve a factor-4 approximation for MSR-2 and, in general, a factor-$2d$ approximation for MSR-d. This was done by converting the problem to computing the maximal independent set in t-interval graphs, which admit a factor-$2t$ approximation (Bar-Yehuda et al., 2006).

In biological data, duplicate markers are possible in some genomic maps, as the so-called paralogy set. We denote by *MSR-DU* the problem MSR with the following variation DU:

DU — Duplicate markers are allowed in the genomic maps and in different strips.

$$
\begin{aligned}
G_1 &= \langle 1,2,3,4,5,6,7,8,9,10,11,12 \rangle \\
G_2 &= \langle -9,-4,-7,-6,8,1,3,2,-12,-11,-10,-5 \rangle \\
S_1 &= \langle 1,2 \rangle \\
S_2 &= \langle 6,7,9 \rangle \\
S_3 &= \langle 10,11,12 \rangle \\
\pi_1 &= \langle 1,2,3 \rangle \\
\pi_2 &= \langle -2,1,-3 \rangle \\
G_1' &= \langle 1,2,6,7,9,10,11,12 \rangle \\
G_2' &= \langle -9,-7,-6,1,2,-12,-11,-10 \rangle
\end{aligned}
$$

Figure 1: An example for the problem MSR, with solution size being eight.

It should be noted that while duplicate markers are allowed in the genomic maps and in *different* strips in the variation MSR-DU, they cannot appear in any individual strip since each strip must be composed of a sequence of distinct markers.

Sometimes, when building genomic maps, a priori information about the gene markers can be derived from comparative analysis. For example, certain genes that are responsible for important genetic functions in several closely related species can often be identified. It is reasonable to give the corresponding gene markers larger weights. Denote by *MSR-WT* the problem MSR with the following additional weight constraint WT:

WT — The total weight of markers in the strips is between two positive integers w_1 and w_2.

In this paper, we show that MSR is in fact NP-complete, via a polynomial time reduction from One-in-Three 3SAT, which was shown to be NP-complete (Schaefer, 1978; Garey and Johnson, 1979). On the other hand, we show that (the complement of) MSR, together with its close variants MSR-d and MSR-DU, is fixed-parameter tractable. More specifically, let k be the minimum number of markers deleted in the optimal solutions of various versions of MSR, the running times of our algorithms are $O(2^{3.61k}n+n^2)$ for MSR, $O(\begin{pmatrix} (2d+1)k \\ k \end{pmatrix} dn+dn^2)$ for MSR-d, and $O(2^{7.22k}n+n^2)$ for MSR-DU respectively.

This paper is organized as follows. In Section 2, we show NP-completeness for MSR. In Section 3, we present fixed-parameter algorithms for MSR and some of its variants. In Section 4, we conclude the paper with a few open questions.

2 MSR is NP-complete

We prove MSR to be NP-complete in this section. It is clear that MSR is in NP. We show that MSR is NP-hard by a reduction from the NP-hard problem One-in-Three 3SAT (Schaefer, 1978).

Theorem 1 *MSR is NP-complete.*

Proof. We reduce from the NP-complete problem On-in-Three 3SAT to MSR. Let $\phi = f_1 \wedge f_2 \wedge \ldots \wedge f_m$ be an One-in-Three 3SAT instance, i.e., a boolean formula of m clauses in conjunctive normal form, with n variables $v_1, v_2, \ldots, v_n$, where each clause f_k is the disjunction of exactly three distinct literals, like $(v_2 \vee v_5 \vee \bar{v}_7)$. The truth assignment satisfies another constraint that exactly one literal in each clause is set to true. In the above clause, $v_2 = \text{false}, v_5 = \text{true}$, and $v_7 =$ true is a valid one-in-three truth assignment. We assume that both $m, n > 2$.

Our construction uses $11m + 4n + 30n^2m + 15nm^2$ distinct markers:

- $9m$ clause markers — $f^1_{i,j,k}$, $f^2_{i,j,k}$, and $f^3_{i,j,k}$, if v_i appears as the j-th literal in f_k; $\bar{f}^1_{i,j,k}$, $\bar{f}^2_{i,j,k}$, and $\bar{f}^3_{i,j,k}$, if $\bar{v}_i$ appears as the j-th literal in f_k, for $1 \leq i \leq n, 1 \leq j \leq 3, 1 \leq k \leq m$,
- $2m$ clause markers a_i and $\dot{a}_i$ for $1 \leq i \leq m$,
- $2n$ variable markers x_i and $\dot{x}_i$ for $1 \leq i \leq n$,
- $2n$ variable markers y_i and $\dot{y}_i$ for $1 \leq i \leq n$,
- m peg strings (of $15nm$ markers each) Z_k for $1 \leq k \leq m$, with $Z_k = z_{k,1} z_{k,2} \ldots z_{k,15nm}$.
- n peg strings (of $15nm$ markers each) U_i for $1 \leq i \leq n$, with $U_i = u_{i,1} u_{i,2} \ldots u_{i,15nm}$.
- n peg strings (of $15nm$ markers each) W_i for $1 \leq i \leq n$, with $W_i = w_{i,1} w_{i,2} \ldots w_{i,15nm}$.

Throughout this proof, all of the peg strings are used to enforce the truth assignment and, as will be shown a bit later, no peg string is ever deleted to obtain the optimal solution for any converted MSR instance.

For the ease of description, we simply say that $A_{i,j,k} = f^1_{i,j,k} f^2_{i,j,k} f^3_{i,j,k}$ ($\bar{f}^1_{i,j,k} \bar{f}^2_{i,j,k} \bar{f}^3_{i,j,k}$) are the *associates* of v_i ($\bar{v}_i$) in f_k and they always appear together in one of the input map G and in the final optimal solution (— but not in the other input map H, as will be explained a bit later). For each variable $v_i, 1 \leq i \leq n$, let F_i and $\bar{F}_i$, respectively, be the two sequences of clause associates in which the two literals v_i and $\bar{v}_i$ appear:

$$
\begin{aligned}
F_i &= f^1_{i,j_1,k_1} f^2_{i,j_1,k_1} f^3_{i,j_1,k_1} f^1_{i,j_2,k_2} f^2_{i,j_2,k_2} f^3_{i,j_3,k_3} \ldots f^1_{i,j_p,k_p} f^2_{i,j_p,k_p} f^3_{i,j_p,k_p}, \\
\bar{F}_i &= \bar{f}^1_{i,j'_1,k'_1} \bar{f}^2_{i,j'_1,k'_1} \bar{f}^3_{i,j'_1,k'_1} \bar{f}^1_{i,j'_2,k'_2} \bar{f}^2_{i,j'_2,k'_2} \bar{f}^3_{i,j'_2,k'_2} \ldots \bar{f}^1_{i,j'_q,k'_q}, \bar{f}^2_{i,j'_q,k'_q}, \bar{f}^3_{i,j'_q,k'_q},
\end{aligned}
$$

let

$$X_i = -\dot{x}_i F_i - x_i \;\; y_i \bar{F}_i \dot{y}_i.$$

Given three sequences $B_1 = b_{11}b_{12}\dots b_{1p}$, $B_2 = b_{21}b_{22}\dots b_{2p}$, and $B_3 = b_{31}b_{32}\dots b_{3p}$, let $(B_1 \otimes B_2 \otimes B_3)$ be the sequence obtained by listing letters in B_1, B_2, B_3 alternately; i.e., $B_1 \otimes B_2 \otimes B_3 = b_{11}b_{21}b_{31}b_{12}b_{22}b_{32}\dots b_{1p}b_{2p}b_{3p}$. For each clause $f_k, 1 \le k \le m$, let

$$Y_k = a_k(A_{k_1,1,k} \otimes A_{k_2,2,k} \otimes A_{k_3,3,k})\dot{a}_k,$$

where $A_{k_j,j,k} = a^1_{k_j,j,k}a^2_{k_j,j,k}a^3_{k_j,j,k}$, with $a_{k_j,j,k} = f_{k_j,j,k}$ if v_{k_j} is the j-th literal in f_k or $a_{k_j,j,k} = \bar{f}_{k_j,j,k}$, if $\bar{v}_{k_j}$ is the j-th literal in f_k, for $1 \le j \le 3$ and for some $1 \le k_j \le n$. More precisely,

$$Y_k = a_k a^1_{k_1,1,k} a^1_{k_2,2,k} a^1_{k_3,3,k} a^2_{k_1,1,k} a^2_{k_2,2,k} a^2_{k_3,3,k} a^3_{k_1,1,k} a^3_{k_2,2,k} a^3_{k_3,3,k} \dot{a}_k.$$

Construct two genomic maps

$$\begin{aligned} G &= W_1 \dots W_n \;\; X_1U_1 \dots X_nU_n \;\; Z_1 \dots Z_m \;\; a_m\dot{a}_m \dots a_2\dot{a}_2a_1\dot{a}_1, \\ H &= x_1y_1\dot{x}_1\dot{y}_1W_1 \dots x_ny_n\dot{x}_n\dot{y}_nW_n \;\; Y_1Z_1 \dots Y_mZ_m \;\; U_1 \dots U_n. \end{aligned}$$

Note that G and H each contains the $11m + 4n + 30n^2m + 15nm^2$ distinct markers exactly once. We show that the one-in-three 3SAT formula ϕ is satisfiable if and only if G has a subsequence G' and H has a subsequence H' such that the total length of the strips in G' and H' is exactly $3m + 2n + 30n^2m + 15nm^2$.

We first prove the "only if" direction. Let τ be a truth assignment that satisfies ϕ. For each i, let

$$X'_i = \begin{cases} F_i \;\; y_i\dot{y}_i & \text{if } \tau(v_i) = \text{true}, \\ -\dot{x}_i - x_i \;\; \bar{F}_i & \text{if } \tau(v_i) = \text{false}. \end{cases}$$

In short, X'_i is obtained from X_i by deleting the clause associates of $\bar{v}_i$ in f_k if $\tau(v_i) = \text{true}$. Symmetrically, X'_i is obtained from X_i by deleting the clause associates of v_i in f_k if $\tau(v_i) = \text{false}$. We obtain Y'_k from Y_k by first deleting a_k and $\dot{a}_k$. Then, keep the associates of the (only) literal which sets f_k to be true. In other words, if f_k is satisfied, then $|Y'_k| = 3$. (If f_k is not satisfied, then $|Y'_k| = 2$; i.e., we will have to keep $Y'_k = a_k\dot{a}_k$ — that causes a much smaller solution for the MSR instance.)

Formally, as a literal can only appear in a clause exactly once

$$Y'_k = \begin{cases} f^1_{k_j,j,k}f^2_{k_j,j,k}f^3_{k_j,j,k}, & \text{if } v_{k_j} \text{ is the } j\text{-th literal in } f_k \text{ and } \tau(v_{k_j}) = \text{true}, \\ \bar{f}^1_{k_j,j,k}\bar{f}^2_{k_j,j,k}\bar{f}^3_{k_j,j,k}, & \text{if } \bar{v}_{k_j} \text{ is the } j\text{-th literal in } f_k \text{ and } \tau(v_{k_j}) = \text{false} \end{cases}$$

Then we have

$$G'' = W_1 \dots W_n \;\; X'_1U_1X'_2U_2 \dots X'_nU_n \;\; Z_1 \dots Z_m,$$

and

$$H'' = x_1 y_1 \dot{x}_1 \dot{y}_1 W_1 \ldots x_n y_n \dot{x}_n \dot{y}_n W_n \;\; Y'_1 Z_1 \ldots Y'_m Z_m \;\; U_1 \ldots U_n.$$

G' and H' are obtained from G'' and H'' as follows. G' and H' each contains exactly one of each of the variable strips $x_i \dot{x}_i$ and $y_i \dot{y}_i$ (with $y_i \dot{y}_i$ corresponding to true, and $x_i \dot{x}_i$ to false), and all of the peg strings (strips) U_i, W_i, and Z_k. X'_is are obtained by deleting the associates of all literals which do not make f_k true and hence have been deleted from Y_k (i.e., not appearing in Y'_k). The satisfying truth assignment also guarantees that each Y'_k contains exactly three associates corresponding to the true literal in clause f_k. Hence, the total length of the strips in G' and H' is exactly $(9m)/3 + (4n)/2 + 30n^2m + 15nm^2 = 3m + 2n + 30n^2m + 15nm^2$.

For example, an one-in-three 3SAT formula of the following four clauses (over four variables)

$$f_1 = (\bar{v}_1 \vee v_2 \vee \bar{v}_3) \quad f_2 = (v_1 \vee v_2 \vee \bar{v}_4) \quad f_3 = (v_2 \vee v_3 \vee v_4) \quad f_4 = (\bar{v}_1 \vee \bar{v}_2 \vee \bar{v}_4)$$

corresponds to the two genomic sequences

$$\begin{aligned} G = \quad & W_1 W_2 W_3 W_4 \\ & -\dot{x}_1 f^1_{1,1,2} f^2_{1,1,2} f^3_{1,1,2} - x_1 \;\; y_1 \bar{f}^1_{1,1,1} \bar{f}^2_{1,1,1} \bar{f}^3_{1,1,1} \bar{f}^1_{1,1,4} \bar{f}^2_{1,1,4} \bar{f}^3_{1,1,4} \dot{y}_1 U_1 \\ & -\dot{x}_2 f^1_{2,2,1} f^2_{2,2,1} f^3_{2,2,1} f^1_{2,2,2} f^2_{2,2,2} f^3_{2,2,2} f^1_{2,1,3} f^2_{2,1,3} f^3_{2,1,3} - x_2 \;\; y_2 \bar{f}^1_{2,2,4} \bar{f}^2_{2,2,4} \bar{f}^3_{2,2,4} \dot{y}_2 U_2 \\ & -\dot{x}_3 f^1_{3,2,3} f^2_{3,2,3} f^3_{3,2,3} - x_3 \;\; y_3 \bar{f}^1_{3,3,1} \bar{f}^2_{3,3,1} \bar{f}^3_{3,3,1} \dot{y}_3 U_3 \\ & -\dot{x}_4 f^1_{4,3,3} f^2_{4,3,3} f^3_{4,3,3} - x_4 \;\; y_4 \bar{f}^1_{4,3,2} \bar{f}^2_{4,3,2} \bar{f}^3_{4,3,2} \bar{f}^1_{4,3,4} \bar{f}^2_{4,3,4} \bar{f}^3_{4,3,4} \dot{y}_4 U_4 \\ & Z_1 Z_2 Z_3 Z_4 a_4 \dot{a}_4 a_3 \dot{a}_3 a_2 \dot{a}_2 a_1 \dot{a}_1 \end{aligned}$$

$$\begin{aligned} H = \quad & x_1 y_1 \dot{x}_1 \dot{y}_1 W_1 \; x_2 y_2 \dot{x}_2 \dot{y}_2 W_2 \; x_3 y_3 \dot{x}_3 \dot{y}_3 W_3 \; x_4 y_4 \dot{x}_4 \dot{y}_4 W_4 \\ & a_1 \; \bar{f}^1_{1,1,1} f^1_{2,2,1} \bar{f}^1_{3,3,1} \bar{f}^2_{1,1,1} f^2_{2,2,1} \bar{f}^2_{3,3,1} \bar{f}^3_{1,1,1} f^3_{2,2,1} \bar{f}^3_{3,3,1} \; \dot{a}_1 Z_1 \\ & a_2 \; f^1_{1,1,2} f^1_{2,2,2} \bar{f}^1_{4,3,2} f^2_{1,1,2} f^2_{2,2,2} \bar{f}^2_{4,3,2} f^3_{2,2,2} f^3_{1,1,2} \bar{f}^3_{4,3,2} \; \dot{a}_2 Z_2 \\ & a_3 \; f^1_{2,1,3} f^1_{3,2,3} f^1_{4,3,3} f^2_{2,1,3} f^2_{3,2,3} f^2_{4,3,3} f^3_{2,1,3} f^3_{3,2,3} f^3_{4,3,3} \; \dot{a}_3 Z_3 \\ & a_4 \; \bar{f}^1_{1,1,4} \bar{f}^1_{2,2,4} \bar{f}^1_{4,3,4} \bar{f}^2_{1,1,4} \bar{f}^2_{2,2,4} \bar{f}^2_{4,3,4} \bar{f}^3_{1,1,4} \bar{f}^3_{2,2,4} \bar{f}^3_{4,3,4} \; \dot{a}_4 Z_4 \\ & U_1 U_2 U_3 U_4. \end{aligned}$$

The truth assignment

$$\tau(v_1) = \text{true} \quad \tau(v_2) = \text{false} \quad \tau(v_3) = \text{false} \quad \tau(v_4) = \text{true}$$

corresponds to

$$G' = W_1 W_2 W_3 W_4 \; f^1_{1,1,2} f^2_{1,1,2} f^3_{1,1,2} y_1 \dot{y}_1 U_1 \; -\dot{x}_2 - x_2 \bar{f}^1_{2,2,4} \bar{f}^2_{2,2,4} \bar{f}^3_{2,2,4} U_2$$

$$-\dot{x}_3 - x_3 \bar{f}^1_{3,3,1}\bar{f}^2_{3,3,1}\bar{f}^3_{3,3,1}U_3\ f^1_{4,3,3}f^2_{4,3,3}f^3_{4,3,3}y_4\dot{y}_4U_4\ Z_1Z_2Z_3Z_4.$$

and

$$H' = y_1\dot{y}_1W_1\ x_2\dot{x}_2W_2\ x_3\dot{x}_3W_3\ y_4\dot{y}_4W_4\ \bar{f}^1_{3,3,1}\bar{f}^2_{3,3,1}\bar{f}^3_{3,3,1}Z_1\ f^1_{1,1,2}f^2_{1,1,2}f^3_{1,1,2}Z_2$$
$$f^1_{4,3,3}f^2_{4,3,3}f^3_{4,3,3}Z_3\ \bar{f}^1_{2,2,4}\bar{f}^2_{2,2,4}\bar{f}^3_{2,2,4}Z_4\ U_1U_2U_3U_4,$$

We do not list U_i, W_i and Z_k as they are just long sequences of distinct markers.

We next prove the "if" direction. Let G', H' be a subsequence of G, H respectively such that the total length of the strips in G' and H' is exactly $3m+2n+30n^2m+15nm^2$. It is clear that all the peg strings (strips) U_i, W_i and Z_k must be in the optimal solution for the corresponding MSR instance. The reason is that if we break any strip in U_i, W_i or Z_k, say we want to use strip $\dot{y}_1y_2$ by deleting W_1 and U_1, even if we somehow put all the $11m+4n$ non-peg markers in the optimal solution, the optimal solution size hence obtained would be less than $30n^2m + 15nm^2 < 3m + 2n + 30n^2m + 15nm^2$. In fact, breaking any one of U_i, V_i or Z_k, which is of length $15nm$, will decrease the optimal solution size to below $30n^2m + 15nm^2$. This is because $11m + 4n < 15m + 15n < 15mn$, when $m, n > 2$.

The alternating pattern of the clause markers in Y_k and $F_i, \bar{F}_i$ ensures that there is at most one common strip of length at most three between any Y_k and $F_i, \bar{F}_i$. If no strip of length three in Y_k is selected, then $a_k\dot{a}_k$ will be a strip of length two. Hence the length of the clause strips in the optimal solution will be less than $3m$. So, in the optimal solution for this MSR instance, if we have $3m$ of clause strips then we must have exactly one strip of length three from each Y_k and the three markers must belong to some clause associates to match the corresponding ones in some $F_i, \bar{F}_i$. Similarly, the alternating pattern of the variable markers and the corresponding peg markers in G and H ensures that in the optimal solution there are n variable strips of length two in G' and H', that is, either $x_i\dot{x}_i$ or $y_i\dot{y}_i$ for $1 \leq i \leq n$.

Therefore, in the optimal solution for this MSR instance, we have a valid truth assignment for ϕ: if clause markers in F_i are in the solution, we set v_i as true; if clause markers in $\bar{F}_i$ are in the solution, we set v_i as false. Obviously, this assignment will satisfy each clause exactly once. Therefore, the one-in-three 3SAT formula ϕ is satisfied by this truth assignment.

The reduction time is clearly $O((m+n)^3)$ time. This completes the proof of Theorem 1. □

It should be noted that $-\dot{x}_i \cdots - x_i$ in F_i and $\bar{F}_i$ could be changed to $x_i \cdots \dot{x}_i$ and the proof still works. So MSR is in fact NP-complete even when all the markers are of positive signs. Moreover, it is clear that the complement of MSR is NP-complete as well: the above proof implies that the one-in-three 3SAT formula ϕ is satisfiable if and only if we have to delete $8m + 2n$ markers in G (resp. H) to obtain the subsequence G' (resp. H'). Therefore we have the following corollary.

Corollary 1 *The minimization (or the complement) version of MSR is NP-complete.*

3 FPT Algorithms for MSR and Its Variants

In this section, we consider solving (the complement of) MSR with an FPT algorithm. Basically, an FPT algorithm for an optimization problem Π with optimal solution value k is an algorithm which solves the problem in $O(f(k)n^c)$ time, where f is any function only on k, n is the input size and c is some fixed constant not related to k. More details on FPT algorithms can be found in the monograph on parameterized complexity (Downey and Fellows, 1999). We first prove the following lemma.

Lemma 1 *Before any marker is deleted, if xy or $-y-x$ appears in both G_1 and G_2 (or, if xy appears in G_1 and $-y-x$ appears in G_2, and vice versa), then there is an optimal solution for MSR which has xy or $-y-x$ as a strip.*

Proof. Wlog, we only consider the case when xy appears in G_1 and $-y-x$ appears in G_2. The cases when xy $(-y-x)$ appears in both G_1 and G_2 are similar. Let the length-4 substring in G_1 containing xy be $p_1(x)xys_1(y)$, and let the length-4 substring in G_2 containing xy be $p_2(y)-y-xs_2(x)$. We assume that $p_1(x) \neq -s_2(x)$ and $s_1(y) \neq -p_2(y)$, as otherwise the lemma is obviously true.

If x is deleted to obtain any optimal solution, then $p_1(x)y$ in G_1 is a breakpoint. The reason is that $p_2(y)-y$ and $-ys_2(x)$ in G_2 cannot be equal to $p_1(x)y$ or its signed reversal — the former is due to the positive sign on y in $p_1(x)y$, and the latter is due to $s_1(y) \neq -p_2(y)$. Similarly, $ys_1(y)$ in G_1 is a breakpoint (as $p_2(y)-y$ and $-ys_2(x)$ in G_2 cannot be equal to $p_1(x)y$ or its signed reversal). Therefore, when x is deleted the strip xy is destroyed, which is a contradiction. If y is deleted, the same argument follows.

If both x, y are deleted to obtain any optimal solution, we consider three cases.

1. If a maximal substring S_1 of G_1 ending at $p_1(x)$ and a maximal substring S_1' of G_1 starting at $s_1(y)$ are strips of length at least two, then we can put x, y back, and delete $p_1(x), s_1(y)$ to obtain a solution of larger size.

2. If one of S_1, S_1' (say, S_1, which must be equal to $p_1(x)$) has length one, then we can delete S_1, put x, y back to obtain a solution of larger size.

3. If both of S_1, S_1' have length one, then we can delete $p_1(x), s_1(y)$, put x, y back to obtain a solution which is of the same size as the current optimal solution.

Hence, the lemma is proven. □

We note that the above lemma also holds when a strip is of length greater than two.

Let Σ be the alphabet for the input maps G_1 and G_2. The above lemma gives us a kernelization procedure.

1. Identify a set of strips from the two sequences, without deleting any gene marker.

2. For each strip identified, change it to a new letter in Σ_1, with $\Sigma_1 \cap \Sigma = \emptyset$. Let the resulting sequences be G'_1, G'_2.

Let Σ_1 be the set of new letters used in the kernelization process, with $\Sigma_1 \cap \Sigma = \emptyset$. We have the following lemmas.

Lemma 2 *There is an optimal MSR solution of size k for G_1 and G_2 if and only if the solution can be obtained by deleting k markers in Σ from G'_1 and G'_2 respectively.*

Proof. In the kernelization process, without deleting any gene marker, we change each (existing) strip into a letter in $\Sigma_1 - \Sigma$. Following Lemma 1, these letters in $\Sigma_1 - \Sigma$ do not have to be deleted to obtain an optimal solution for MSR. □

Lemma 3 *In G'_1 (resp. G'_2), there are at most $5k$ letters (markers) in Σ.*

Proof. Following Lemma 2, the optimal solution for MSR can obtained by deleting markers (letters) only in Σ from G'_1 (resp. G'_2). For each letter x deleted in G'_1, there are at most two other letters y_1 and z_1 in Σ, preceding and succeeding x. (In this case, y_1xz_1 is a substring in G'_1.) The same claim holds for the letter x deleted in G'_2, i.e., for each letter x deleted in G'_2, there are at most two other letters y_2 and z_2 in Σ, preceding and succeeding x. Therefore, x is associated with at most five letters in Σ, e.g., $\{x, y_1, z_1, y_2, z_2\}$. Consequently, we have at most $5k$ letters in Σ in G'_1 (resp. G'_2). □

Theorem 2 *There is an FPT algorithm for MSR which runs in $O(2^{3.61k}n + n^2)$ time.*

Proof. Following Lemma 2 and Lemma 3, we can choose k letters in Σ from G'_1, G'_2. The number of choices, is hence bounded by

$$\begin{pmatrix} 5k \\ k \end{pmatrix} \approx 2^{3.61k},$$

using Stirling's formula. For each choice, we can check whether it is valid, i.e., whether all remaining markers are in some strip in G'_1 and G'_2. This can be done in linear time if we spend $O(n^2)$ time in advance, i.e., building a correspondence between all of the identical markers in G_1, G_2. So the overall running time of the algorithm is $O(2^{3.61k}n + n^2)$ time. Note that the algorithm will report 'no solution of size k', if none of the choices leads to a valid solution. □

It is obvious that the algorithm also works for MSR-d, except that the kernel size becomes $(2d + 1)k$. For MSR-DU, the algorithm is similar. But we need to make $\begin{pmatrix} 5k \\ k \end{pmatrix}$ choices of letters in Σ from each of G'_1 and G'_2. So the running time will be $O(2^{7.22k}n + n^2)$ time.

Corollary 2 *MSR-d can be solved in $O(\binom{(2d+1)k}{k} dn + dn^2)$ time and MSR-DU can be solved in $O(2^{7.22k} n + n^2)$ time.*

For MSR-WT, if the weights for markers are arbitrary then obviously Lemma 1 does not hold anymore and the above algorithm will not work. But if the weights are set so that Lemma 1 still holds, e.g., the weights must be one or two, then we will still be able to obtain a similar result.

4 Concluding Remarks

We note that (the minimization version of) the MSR problem can be thought of as the complement of the problem MWIS in 2-interval graphs, also known as the problem *2-Interval Pattern* (Vialette, 2004), which has been extensively studied because of its application to RNA secondary structure prediction (Ber-Yehuda et al., 2006; Blin et al., 2007; Chen et al., 2007; Crochemore et al., 2008; Jiang, 2007). This probably explains why there is an FPT algorithm for the minimization version, or the complement, of MSR.

It would be interesting to know whether our FPT algorithms can be further improved. The running times we have obtained for the complements of MSR and its variants are not efficient enough to make them truly useful in practice. To make such an FPT algorithm practical for MSR datasets, which usually has k between 50 to 150, it must be more efficient.

5 Acknowledgments

Lusheng Wang is fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 120905]. Binhai Zhu is partially supported by the NSF grant DMS-0918034. We also thank referees for several useful suggestions and comments.

References

R. Bar-Yehuda, M.M. Halldórsson, J.(S.) Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM Journal on Computing*, 36:1–15, 2006.

G. Blin, G. Fertin, and S. Vialette. Extracting constrained 2-interval subsets in 2-interval sets. *Theoretical Computer Science*, 385:241–263, 2007.

Z. Chen, B. Fu, M. Jiang, and B. Zhu. On recovering syntenic blocks from comparative maps. In *Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA'08)*, LNCS 5165, pages 319–327, 2008.

E. Chen, L. Yang, and H. Yuan. Improved algorithms for largest cardinality 2-interval pattern problem. *Journal of Combinatorial Optimization*, 13:263–275, 2007.

V. Choi, C. Zheng, Q. Zhu, and D. Sankoff. Algorithms for the extraction of synteny blocks from comparative maps. In *Proceedings of the 7th International Workshop on Algorithms in Bioinformatics (WABI'07)*, pages 277–288, 2007.

M. Crochemore, D. Hermelin, G.M. Landau, D. Rawitz, and S. Vialette. Approximating the 2-interval pattern problem. *Theoretical Computer Science*, 395:283–297, 2008.

R. Downey and M. Fellows. *Parameterized Complexity*, Springer-Verlag, 1999.

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

M. Jiang. A 2-approximation for the preceding-and-crossing structured 2-interval pattern problem. *Journal of Combinatorial Optimization*, 13:217–221, 2007.

M. Jiang. Improved approximation algorithms for predicting RNA secondary structures with arbitrary pseudoknots. In *Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management (AAIM'07)*, LNCS 4508, pages 399–410, 2007.

M. Jiang. A PTAS for the weighted 2-interval pattern problem over the preceding-and-crossing model. In *Proceedings of the 1st Annual International Conference on Combinatorial Optimization and Applications (COCOA'07)*, LNCS 4616, pages 378–387, 2007.

T. Schaefer. The complexity of satisfiability problem. In *Proceedings of the 10th ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226, 1978.

S. Vialette. On the computational complexity of 2-interval pattern matching problems. *Theoretical Computer Science*, 312:223–249, 2004.

C. Zheng, Q. Zhu, and D. Sankoff. Removing noise and ambiguities from comparative maps in rearrangement analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4:515–522, 2007.