On Recovering Syntenic Blocks from Comparative Maps

Zhixiang Chen[1], Bin Fu[1], Minghui Jiang[2,⋆], and Binhai Zhu[3]

[1] Department of Computer Science, University of Texas - Pan American, Edinburg, TX 78539-2999, USA
{chen,binfu}@cs.panam.edu
[2] Department of Computer Science, Utah State University, Logan, UT 84322-4205, USA
mjiang@cc.usu.edu
[3] Department of Computer Science, Montana State University, Bozeman, MT 59717, USA
bhz@cs.montana.edu

Abstract. 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) 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. Previously only a heuristic was provided for this problem, which does not guarantee finding the optimal solution, and it was unknown whether the problem is NP-complete or polynomially solvable. In this paper, we develop a factor-4 polynomial-time approximation algorithm for the problem, and show that several close variants of the problem are intractable.

1 Introduction

In comparative genomics, a starting point is to decompose two given genomes into syntenic blocks—segments of chromosomes which are deemed to be homologous in the two input genomes. Various methods have been proposed, but they are very vulnerable to ambiguities and errors. Recently, a heuristic method was proposed to eliminate noise and ambiguities in genomic maps, through handling a problem called Maximal Strip Recovery (see below for the formal definition) [6,16]. But it was unknown whether the problem can be solved in polynomial time or is NP-complete. In this paper, we design a factor-4 polynomial-time approximation algorithm for the problem, and show that several close variants of the problem are intractable.

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) [6,16] is to find two subsequences

⋆ Partially supported by NSF grant DBI-0743670.

B. Yang, D.-Z. Du, and C.A. Wang (Eds.): COCOA 2008, LNCS 5165, pp. 319–327, 2008.
© Springer-Verlag Berlin Heidelberg 2008

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 k sequences (strips) S_j of length at least two, and find d signed permutations π_i of $\langle 1, \ldots, k\rangle$, such that each sequence $G'_i = S_{\pi_i(1)} \ldots S_{\pi_i(k)}$ (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) [6,16] is exactly the problem MSR-2 in our new formulation. We refer to Fig. 1 for an example.

$$
\begin{aligned}
G_1 &= \langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\rangle \\
G_2 &= \langle -8, -5, -7, -6, 4, 1, 3, 2, -12, -11, -10, 9\rangle \\
S_1 &= \langle 1, 3\rangle \\
S_2 &= \langle 6, 7, 8\rangle \\
S_3 &= \langle 10, 11, 12\rangle \\
\pi_1 &= \langle 1, 2, 3\rangle \\
\pi_2 &= \langle -2, 1, -3\rangle \\
G'_1 &= \langle 1, 3, 6, 7, 8, 10, 11, 12\rangle \\
G'_2 &= \langle -8, -7, -6, 1, 3, -12, -11, -10\rangle
\end{aligned}
$$

Fig. 1. An example for the problem MSR

A heuristic based on Maximum Clique (and its complement Maximum Independent Set) was previously given for the problem MSR (MSR-2) [6,16], which does not guarantee finding the optimal solution. It was unknown whether MSR is NP-complete or polynomially solvable. In this paper, we show that the previous heuristic [6,16] can be modified to achieve a factor-4 approximation for MSR; we also show that the problem becomes intractable when the number of genomic maps is increased from two to three. In Section 2 and Section 3, we prove the following two theorems:

Theorem 1. *There is a factor-4 polynomial-time approximation algorithm for MSR.*

Theorem 2. *MSR-3 is NP-complete.*

1.1 Weight Constraint on Markers

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 close 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 additional weight constraint WT:

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

In Section 4, we prove the following theorem:

Theorem 3. *MSR-WT is NP-complete.*

1.2 Number of Non-breaking Points as Score Function

A careful reader will notice that our definition of the problem MSR-2 is slightly different from the original definition of the problem Maximal Strip Recovery (MSR) [16, Page 517]: we require a minimum length of two for each strip, i.e., each strip must contain at least two distinct markers. We believe this requirement is indeed necessary (but overlooked) in the original definition. Indeed, the problem would become trivial otherwise: if a strip can have length one, then we simply need to count the number of common markers, which can be done in $O(n \log n)$ time by sorting.

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

Fig. 2. An example for the two score functions

An alternative way to avoid such short strips is to define the score of each strip in a different way: instead of using the strip length as the score, we can use the number of non-breaking points (or adjacencies) in a strip [3,15]. Then a strip of length l will have a score of $l - 1$ when $l \geq 2$, and a zero score when $l = 1$. We believe that, in practice, the two score functions can lead to different levels of effectiveness of the strip recovery algorithm. We refer to Fig. 2 for an example. For the two subsequences G_1' and G_2', the total strip length (there are four strips) is 8 and the number of non-breaking points is 4. For the two subsequences G_1'' and G_2'', the total strip length (there is only one strip) is 6 and the number of non-breaking points is 5. In the biological context, both solutions may be desirable.

Denote by *MSR-NB* the problem MSR with the alternative score function NB:

NB: The score function uses the number of non-breaking points instead of the strip length.

Also define two problems MSR-NB-3 and MSR-NB-WT, analogous to the two problems MSR-3 and MSR-WT. The difference between the two score functions, the strip length and the number of non-breaking points, has led to contrasting computational complexities (NP-hard versus polynomially solvable) of another biological problem [12,10]. For our three problems MSR-NB, MSR-NB-3, and MSR-NB-WT, however, we can obtain results similar to those for MSR, MSR-3, and MSR-WT. We have the following three theorems (the proofs are omitted to avoid repeating trivial technical details):

Theorem 4. *There is a factor-4 polynomial-time approximation algorithm for MSR-NB.*

Theorem 5. *MSR-NB-3 is NP-complete.*

Theorem 6. *MSR-NB-WT is NP-complete.*

1.3 Duplicate Markers

In our definition of MSR-d, each marker appears exactly once in each genomic map and hence at most once in the strips. In the biological context, however, duplicate markers are rare but still possible, as so-called *paralogy set*: "Two or more strips may contain exactly the same markers but differ in where they appear, by virtue of the paralogy sets their markers belong to." [16, Page 516].

Define the following variation DU of the problem MSR:

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

Note that although duplicate markers may appear in the strips, each strip in itself does not contain duplicate markers since it is a sequence of *distinct* markers. That is, duplicate markers may only appear in different strips.

Denote by *MSR-NB-WT-DU* the problem MSR with the score function NB, the weight constraint WT, and the variation DU. It is not surprising that recovering syntenic blocks becomes harder when duplicate markers are allowed. Recognizing the similarity between the problem MSR-NB-WT-DU and the recently studied problem Exemplar Non-breaking Similarity [3], we prove the following theorem:

Theorem 7. *MSR-NB-WT-DU is W[1]-complete. Moreover, unless P equals NP, MSR-NB-WT-DU does not admit any approximation of factor $n^{1-\epsilon}$ for some ϵ, $0 < \epsilon < 1$.*

Several more variants of the problem MSR can be defined analogously, in particular, MSR-DU and MSR-NB-DU. Our factor-4 approximations for MSR and MSR-NB, with slight modifications, also hold for MSR-DU and MSR-NB-DU; however, the complexities of MSR and its several variants remain unknown. In Section 6, we conclude with two open problems.

2 A Factor-4 Approximation for MSR

We prove Theorem 1 in this section. A heuristic for the problem Maximal Strip Recovery (MSR) was previously proposed [6,16]. This simple heuristic works as follows:

1. Extract a set of pre-strips from the two sequences;
2. Compute an independent set of strips from the pre-strips.

This approach is inefficient because the number of pre-strips could be exponential in the sequence length, and furthermore the problem Maximum Weight Independent Set (MWIS) is NP-hard.

Our factor-4 approximation algorithm for MSR is slightly modified from the previous heuristic [6,16]. For a sequence S and two indices i and j, denote by $S[i, j]$ the substring of S starting at i and ending at j. The algorithm works as follows:

1. Compose $O(n^4)$ 2-intervals, one for each pair of substrings of the two genomic maps G_1 and G_2. For each 2-interval with indices i_1 and j_1 in G_1 and indices i_2 and j_2 in G_2, assign it a weight equal to the maximum length of a common subsequence (may be reversed and negated) of the two substrings $G_1[i_1, j_1]$ and $G_2[i_2, j_2]$.
2. Compute a 4-approximation for MWIS in the intersection graph of the 2-intervals using the fractional local-ratio algorithm for split interval graphs [1].

This completes the proof of Theorem 1.

We note that the problem MWIS in 2-interval graphs is also known as the problem *2-Interval Pattern* [14], which has been extensively studied [1,2,5,7,9,10,11,14] because of its application to RNA secondary structure prediction.

3 MSR-3 Is NP-Complete

We prove Theorem 2 in this section. It is clear that MSR-3 is in NP. We show that MSR-3 is NP-hard by a reduction from the NP-hard problem Separated 2-Interval Pattern [1]. Let L_1 and L_2 be two parallel lines. Denoted by $D = (I, J)$ a *separated 2-interval* that is the union of two closed intervals $I \subset L_1$ and $J \subset L_2$. Given a set of n separated 2-intervals, the problem *Separated 2-Interval Pattern* is to find a maximum independent set in the corresponding intersection graph. By a standard technique in interval graphs, we can assume without loss of generality that the $4n$ endpoints of the $2n$ intervals of the n separated 2-intervals are distinct.

Our construction uses $2n^2+2n$ distinct markers: $2n^2$ *peg markers* $p_{i,j}$ for $1 \leq i \leq n$ and $1 \leq j \leq 2n$, and $2n$ *interval markers* u_i and v_i for $1 \leq i \leq n$.

Use the $2n^2$ peg markers to construct n pairs of *peg sequences* P_i and Q_i of equal length $2n$, for $1 \leq i \leq n$:

$$P_i = p_{i,1} \ldots p_{i,2n},$$
$$Q_i = -p_{i,2n} \ldots - p_{i,1}.$$

Let S be the sequence $u_1 v_1 \ldots u_n v_n$ of $2n$ interval markers. For each separated 2-interval $D_i = (I_i, J_i)$, label the two left endpoints of I_i and J_i with the marker u_i, and label the two right endpoints of I_i and J_i with the marker v_i. Then the $2n$ markers for the $2n$ endpoints of the n intervals I_i, ordered along the line L_1, is a permutation S_1 of S; similarly, the $2n$ markers for the $2n$ endpoints of the n intervals J_i, ordered along the line L_2, is another permutation S_2 of S.

Construct three genomic maps:

$$G_0 = u_1 v_1 P_1 \ldots u_n v_n P_n,$$
$$G_1 = S_1 Q_1 \ldots Q_n,$$
$$G_2 = S_2 Q_1 \ldots Q_n.$$

Note that each genomic map is a signed permutation of the $2n^2 + 2n$ distinct markers. We will show that the set of n separated 2-intervals has an independent set of size at least

k if and only if G_1, G_2, and G_3 have three subsequences G'_1, G'_2, and G'_3, respectively, with a total strip length of at least $2n^2 + 2k$.

We note the following important property of our construction:

Proposition 1. *If a strip of G'_1, G'_2, and G'_3 contains a peg marker $p_{i,j}$, then it does not contain any interval marker or any peg marker $p_{i',j'}$ such that $i' \neq i$.*

We first prove the "only if" direction. Suppose that the set of n separated 2-intervals has an independent set of size k, that is, there are k disjoint separated 2-intervals $D_{i_1}, \ldots, D_{i_k}$. Let G'_1, G'_2, and G'_3, respectively, be the subsequences of G_1, G_2, and G_3 that contain the $2n^2$ peg markers and the $2k$ interval markers $u_{i_1}, v_{i_1}, \ldots, u_{i_k}, v_{i_k}$. Then G'_1, G'_2, and G'_3 have $n + k$ strips: the n strips $P_1, \ldots, P_n$, each of length $2n$, and the k strips $u_{i_1}v_{i_1}, \ldots, u_{i_k}v_{i_k}$, each of length 2. The total strip length is $2n^2 + 2k$.

We next prove the "if" direction. Suppose that G_1, G_2, and G_3 have three subsequences G'_1, G'_2, and G'_3, respectively, with a total strip length of at least $2n^2 + 2k$, $k > 0$. We say that a sequence S *contributes* to a strip R if R contains a marker in S. If a strip contains two interval markers for two different separated 2-intervals, then the two interval markers would enclose a peg sequence P_i in the genomic map G_0. Hence, by Proposition 1, the peg sequence P_i would not contribute to any strip. Note that the total length of the strips is at most the length of each genomic map, which is $2n^2 + 2n$. Also note that the length of each peg sequence is $2n$. If the peg sequence P_i does not contribute to any strip, then the total length of the strips would be at most $2n^2$, which is less than $2n^2 + 2k$, a contradiction. Therefore, if a strip contains an interval marker u_i or v_i of a separated 2-interval D_i, then the strip must contain only the two interval markers u_i and v_i. The total length of strips of peg markers is at most the total length of the n peg sequences, which is $2n^2$. The remaining strip length of at least $2k$ must come from at least k strips of interval markers, which correspond to an independent set of k separated 2-intervals.

The reduction time is clearly polynomial in the size of the Separated 2-Interval Pattern instance. This completes the proof of Theorem 2.

4 MSR-WT Is NP-Complete

We prove Theorem 3 in this section. It is easy to see that MSR-WT is in NP. We show that MSR-WT is NP-hard by a reduction from the NP-hard problem One-In-Three 3SAT [13]. Let $\phi = f_1 \wedge f_2 \wedge \ldots \wedge f_m$ be a boolean formula with m clauses in conjunctive normal form, with n variables $x_1, x_2, \ldots, x_n$. Each clause f_i is the disjunction of exactly three literals, like $(x_2 \vee x_5 \vee \bar{x}_7)$. We will construct two genomic maps G and H and show that ϕ is one-in-three satisfiable (i.e., each clause has exactly one true literal) if and only if G and H have two subsequences G' and H', respectively, such that

1. The total length of the strips in G' and H' is at least some integer ℓ;
2. The total weight of the markers in G' and H' is equal to some integer w (note that we set $w_1 = w_2 = w$ in this proof).

Our construction uses $3m + n$ distinct markers: $3m$ clause markers

$$f_{1,1}, f_{1,2}, f_{1,3}, \ldots, f_{m,1}, f_{m,2}, f_{m,3},$$

and n peg markers

$$g_1, \ldots, g_n.$$

For each clause f_i, label its three literals with the three clause markers $f_{i,1}, f_{i,2}, f_{i,3}$. For each variable x_i, let F_i and $\bar{F}_i$, respectively, be the two sequences of markers for the literals x_i and $\bar{x}_i$:

$$F_i = f_{i_1,j_1} \ldots f_{i_u,j_u},$$
$$\bar{F}_i = f_{i'_1,j'_1} \ldots f_{i'_v,j'_v}.$$

Put

$$S_i = F_i \bar{F}_i,$$
$$T_i = \bar{F}_i F_i.$$

Then construct two genomic maps

$$G = S_1 g_1 \ldots S_n g_n,$$
$$H = T_1 g_1 \ldots T_n g_n.$$

Assign each peg marker g_i a weight that is a decimal number with a one followed by $n - i$ zeros. Assign each clause marker for f_i a weight that is a decimal number with a one followed by $m + n - i$ zeros. Then set the threshold weight w to a decimal number with $m + n$ consecutive ones, and set the threshold strip length ℓ to $m + n$. Note that each genomic map is a permutation of the $3m + n$ markers. Also note that w is exactly the total weight of m clause markers, one for each clause, and the n peg markers.

We first prove the "only if" direction. Let $x_1 = b_1, \ldots, x_n = b_n$ be a one-in-three truth assignment that satisfies ϕ. For each i, obtain two subsequences S'_i and T'_i, respectively, from the two sequences S_i and T_i:

$$S'_i = T'_i = \begin{cases} F_i & \text{if } b_i = \text{true}, \\ \bar{F}_i & \text{if } b_i = \text{false}. \end{cases}$$

Then let

$$G' = S'_1 g_1 \ldots S'_n g_n,$$
$$H' = T'_1 g_1 \ldots T'_n g_n.$$

There is only one strip in G' and H' since $G' = H'$. By the definition of one-in-three truth assignment, each clause f_i contains exactly one true literal, hence exactly one of the three clause markers $f_{i,1}, f_{i,2}, f_{i,3}$ appears in G' (H'). Therefore the total strip length is exactly ℓ, and the total weight of markers is exactly w.

We next prove the "if" direction. Let G' and H' be two subsequences of G and H, respectively, with total strip length at least ℓ and total marker weight exactly w. The

weight condition ensures that exactly one of the three markers for each clause appears in G' and H'. The strip length condition then implies that there is only one strip of length exactly ℓ. Indeed $G' = H'$. The distribution of the clause markers among the "buckets" S'_i and T'_i corresponds to a one-in-three truth assignment for ϕ.

The reduction time is clearly polynomial in the length of ϕ. This completes the proof of Theorem 3.

5 Complexity of MSR-NB-WT-DU

We prove Theorem 7 in this section. We achieve this by showing that the problem MSR-NB-WT-DU contains another difficult problem as a special case. Given a genome $\mathcal{G}$, which is a sequence of genes possibly with duplicates, a genome $\mathcal{G}'$ is *exemplar* of $\mathcal{G}$ if $\mathcal{G}'$ contains the same set of genes as $\mathcal{G}$ does but has no duplicates. Given two genomes $\mathcal{G}$ and $\mathcal{H}$, the problem *Exemplar Non-Breaking Similarity* (ENBS) [3] is to compute two exemplar genomes $\mathcal{G}'$ and $\mathcal{H}'$ such that the number of non-breaking points between $\mathcal{G}'$ and $\mathcal{H}'$ is maximized.

Let $g_1, \ldots, g_n$ be the n distinct genes in the two genomes $\mathcal{G}$ and $\mathcal{H}$. Use n distinct markers, one marker g'_i for each gene g_i, $1 \leq i \leq n$, and construct two genomic maps G and H of markers corresponding to the two genomes $\mathcal{G}$ and $\mathcal{H}$ of genes. The optimization goal of maximizing non-breaking similarity in ENBS corresponds to exactly the score function NB in MSR-NB-WT-DU. To ensure that each gene g_i appears exactly once in $\mathcal{G}'$ and in $\mathcal{H}'$, assign each gene marker g'_i a special weight that is a decimal number with a one followed with $n - i$ zeros, then set the threshold weights w_1 and w_2 to an integer that is a decimal number with n ones.

ENBS is a very difficult problem [3]: (i) ENBS is W[1]-complete; (ii) even if each of the n genes appears exactly once in $\mathcal{G}$ and at most twice in $\mathcal{H}$, ENBS still cannot be approximated within a factor of $n^{1-\epsilon}$ for some ϵ, $0 < \epsilon < 1$, unless P equals NP. Since ENBS is a special case of MSR-NB-WT-DU, these lower bounds automatically apply to MSR-NB-WT-DU. This completes the proof of Theorem 7.

We note that the W[1]-completeness [8] of MSR-NB-WT-DU has the following implication [4]: Let p be the optimal solution value for MSR-NB-WT-DU. Then, unless an unlikely collapse occurs in the parameterized complexity theory, MSR-NB-WT-DU is not solvable in time $f(p)n^{o(p)}$ for any function f.

6 Open Problems

We conclude the paper with two open problems:

1. Are the four problems MSR (MSR-2), MSR-NB, MSR-DU, and MSR-NB-DU NP-complete? (We conjecture that at least MSR-DU and MSR-NB-DU are NP-complete.)
2. Are the two problems MSR-WT and MSR-NB-WT-DU still intractable with only one-sided weight constraint, i.e., the total weight of the strips is at least w_1? (Note that our proofs of Theorem 3 and Theorem 7 use the property that the weight constraint is from both directions.)

References

1. Bar-Yehuda, R., Halldórsson, M.M., Naor, J.(S.), Shachnai, H., Shapira, I.: Scheduling split intervals. SIAM Journal on Computing 36, 1–15 (2006)
2. Blin, G., Fertin, G., Vialette, S.: Extracting constrained 2-interval subsets in 2-interval sets. Theoretical Computer Science 385, 241–263 (2007)
3. Chen, Z., Fu, B., Yang, B., Xu, J., Zhao, Z., Zhu, B.: Non-breaking similarity of genomes with gene repetitions. In: Ma, B., Zhang, K. (eds.) CPM 2007. LNCS, vol. 4580, pp. 119–130. Springer, Heidelberg (2007)
4. Chen, J., Huang, X., Kanj, I., Xia, G.: Linear FPT reductions and computational lower bounds. In: Proceedings of the 36th ACM Symposium on Theory of Computing (STOC 2004), pp. 212–221 (2004)
5. Chen, E., Yang, L., Yuan, H.: Improved algorithms for largest cardinality 2-interval pattern problem. Journal of Combinatorial Optimization 13, 263–275 (2007)
6. Choi, V., Zheng, C., Zhu, Q., Sankoff, D.: Algorithms for the extraction of synteny blocks from comparative maps. In: Proceedings of the 7th International Workshop on Algorithms in Bioinformatics (WABI 2007), pp. 277–288 (2007)
7. Crochemore, M., Hermelin, D., Landau, G.M., Rawitz, D., Vialette, S.: Approximating the 2-interval pattern problem. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 426–437. Springer, Heidelberg (2005)
8. Downey, R., Fellows, M.: Parameterized Complexity. Springer, Heidelberg (1999)
9. Jiang, M.: A 2-approximation for the preceding-and-crossing structured 2-interval pattern problem. Journal of Combinatorial Optimization 13, 217–221 (2007)
10. Jiang, M.: Improved approximation algorithms for predicting RNA secondary structures with arbitrary pseudoknots. In: Kao, M.-Y., Li, X.-Y. (eds.) AAIM 2007. LNCS, vol. 4508, pp. 399–410. Springer, Heidelberg (2007)
11. Jiang, M.: A PTAS for the weighted 2-interval pattern problem over the preceding-and-crossing model. In: Dress, A.W.M., Xu, Y., Zhu, B. (eds.) COCOA. LNCS, vol. 4616, pp. 378–387. Springer, Heidelberg (2007)
12. Lyngsø, R.B.: Complexity of pseudoknot prediction in simple models. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 919–931. Springer, Heidelberg (2004)
13. Schaefer, T.: The complexity of satisfiability problem. In: Proceedings of the 10th ACM Symposium on Theory of Computing (STOC 1978), pp. 216–226 (1978)
14. Vialette, S.: On the computational complexity of 2-interval pattern matching problems. Theoretical Computer Science 312, 223–249 (2004)
15. Watterson, G., Ewens, W., Hall, T., Morgan, A.: The chromosome inversion problem. Journal of Theoretical Biology 99, 1–7 (1982)
16. Zheng, C., Zhu, Q., Sankoff, D.: Removing noise and ambiguities from comparative maps in rearrangement analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4, 515–522 (2007)