Disjoint Segments with Maximum Density

Yen Hung Chen[1], Hsueh-I Lu[2,⋆] and Chuan Yi Tang[1]

[1] Department of Computer Science,
National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.
{dr884336, cytang}@cs.nthu.edu.tw

[2] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei 106, Taiwan, R.O.C.
hil@csie.ntu.edu.tw

Abstract. Given a sequence A of numbers and two positive integers ℓ and k, we study the problem to find k disjoint segments of A, each has length at least ℓ, such that their sum of densities is maximized. We give the first known polynomial-time algorithm for the problem: For general k, our algorithm runs in $O(n\ell k)$ time. For the special case with $k = 2$ (respectively, $k = 3$), we also show how to solve the problem in $O(n)$ (respectively, $O(n + \ell^2)$) time.

1 Introduction

Let $A = \langle a_1, a_2, \ldots, a_n\rangle$ be the input sequence of n numbers. Let $A_{i,j}$ denote the consecutive subsequence $\langle a_i, a_{i+1}, \ldots, a_j\rangle$ of A. The *length* of $A_{i,j}$, denoted $|A_{i,j}|$, is $j - i + 1$. The *density* of $A_{i,j}$, denoted $d(A_{i,j})$ is $\frac{a_i+a_{i+1}+\cdots+a_j}{j-i+1}$ of $A_{i,j}$. Observe that with an $O(n)$-time preprocessing to compute all $O(n)$ prefix sums $a_1 + a_2 + \cdots + a_j$ of A, the density of any segment $A_{i,j}$ can be obtained in $O(1)$ time.

Two segments $A_{i,j}$ and $A_{i',j'}$ of A are *disjoint* if $i \le j < i' \le j'$ or $i' \le j' < i \le j$. Two segments of A *overlap* if they are not disjoint. Motivated by the locating GC-rich regions [9, 14, 15, 16], CpG islands [3, 5, 11, 18] in a genomic sequence and annotating multiple sequence alignments [17], Lin, Huang, Jiang and Chao [13] formulated and gave an $O(n \log k)$-time heuristic algorithm for the problem of identifying k disjoint segments of A with maximum sum of densities. Specifically, given two additional positive integers k and ℓ, the problem is to find k disjoint segments of A, each has length at least ℓ, such that the sum of their densities is maximized. We present the first known polynomial-time algorithm to solve the problem. Our algorithm runs in $O(n\ell k)$ time for general k. We also show that the special case with $k = 2$ (respectively, $k = 3$) can be solved in $O(n)$ (respectively, $O(n + \ell^2)$) time.

⋆ The corresponding author. Address: 1 Roosevelt Road, Section 4, Taipei 106, Taiwan, R.O.C. Webpage: www.csie.ntu.edu.tw/∼hil/.

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3515, pp. 845–850, 2005.

Related work. When $k = 1$, the problem studied in the present paper becomes the extensively studied *maximimum-density segment problem* [2, 6, 9, 10, 12]. The problem for general k is also closely related to the *GTile with bounded number of tiles* problem [1], which is a natural extension of the *maximum-sum segment* problem studied in [12, 4].

The rest of this paper is organized as follows. Section 2 describes our $O(n\ell k)$-time algorithm for general k. Section 3 shows how to solve the case with $k = 2$ in $O(n)$ time. Section 4 shows how to solve the case with $k = 3$ in $O(n + \ell^2)$ time. Section 5 concludes the paper with open questions.

2 Our Algorithm for General *k*

For a set U of segments, let $D(U) = \sum_{S \in U} d(S)$. A set of segments is *feasible* to our problem if it consists of k disjoint segments of A, each has length at least ℓ. A set U^* of segments is *optimal* if U^* is feasible and $D(U^*) \geq D(U)$ holds for any feasible set U.

Lemma 1. *There exists an optimal set U^* of segments such each segment in U^* has length less than 2ℓ.*

Proof. Suppose that U^* contains a segment $A_{i,j}$ with $|A_{i,j}| \geq 2\ell$. Then, both $U^* \cup \{A_{i,i+\ell-1}\} - \{A_{i,j}\}$ and $U^* \cup \{A_{i+\ell,j}\} - \{A_{i,j}\}$. Moreover, one of them has to be optimal, since $\max(d(A_{i,i+\ell-1}), d(A_{i+\ell,j})) \geq d(A_{i,j})$. We then use the new optimal set to replace the original U^*. The lemma is proved by continuing this process until each segment in the resulting optimal set U^* has length less than 2ℓ.

According to Lemma 1, it suffices to focus on segments with lengths at least ℓ and less than 2ℓ. Let ρ be the number of such segments in A. Clearly, $\rho = O(n\ell)$. Define G to be a graph on these ρ segments such that two nodes in G are adjacent if and only if their corresponding segments overlap in A. Observe that G is an interval graph. Let the weight of each node be the density of its corresponding segment. Then, the problem to compute an optimal set U^* of segments becomes the problem to identify a maximum weight independent set of G that has size k. To the best of our knowledge, no such an algorithm is known, although the version without restriction on the size has been studied in the literature [8, 7].

Our algorithm for identifying an optimal U^* is via the standard technique of dynamic programming as shown below. For each $j = 1, 2, \ldots, n$, let A_j consist of the segments $A_{i,j}$ of A with $1 \leq i \leq j \leq n$ and $\ell \leq |A_{i,j}| < 2\ell$. For each $j = 1, 2, \ldots, n$, let $U^*_{j,t}$ denote a set of t disjoint segments of $A_{1,j}$, each has length at least ℓ and less than 2ℓ, such that $D(U^*_{j,t})$ is maximized. Note that $U^* = U^*_{n,k}$. One can easily compute all $U^*_{j,1}$ with $1 \leq j \leq n$ in $O(n\ell)$ time. For technical reason, if $j < t\ell$, then let $U^*_{j,t} = \emptyset$ and $D(U^*_{j,t}) = -\infty$. To compute all $O(nk)$ entries of $U^*_{j,t}$ in $O(n\ell k)$ time, we use the following straightforward procedure for each $t > 1$ and $j \geq t\ell$.

> Let $U^*_{j,t} = \{A_{s,j}\} \cup U^*_{s-1,t-1}$, where s is an index i that maximizes $d(A_{i,j}) + D(U^*_{i-1,t-1})$ over all indices i such that $A_{i,j}$ is a segment in A_j.

Since each A_j has size $O(\ell)$, if those $U^*_{j,t-1}$ with $j = 1, 2, \ldots, n$ are available, then all $U^*_{j,t}$ with $j = 1, 2, \ldots, n$ can be computed in $O(n\ell)$ time. One can then obtain $U^* = U^*_{n,t}$ in $O(n\ell k)$ time by iterating the above process for $t = 2, 3, \ldots, k$. Therefore, we have the following theorem.

Theorem 1. *It takes $O(n\ell k)$ time to find k disjoint segments of a length-n sequence, each has length at least ℓ, such that the sum of their densities is maximized.*

3 Our Algorithm for $k = 2$

It turns out that the linear time algorithm of Chung and Lu [2] for the case with $k = 1$ can be a useful subroutine to solve the case with $k = 2$ in linear time. For each $i = 1, 2, \ldots, n$, let P_i (respectively, Q_i) be a maximum density segment with length at least ℓ for $A_{1,i}$ (respectively, $A_{i,n}$). Clearly, P_i and Q_{i+1} are disjoint segments of A for each $i = 1, 2, \ldots, n-1$. Chung and Lu's algorithm has the nice feature that can process the input sequence in an online manner. Therefore, all P_i and Q_i with $1 \leq i \leq n$ can be computed by Chung and Lu's algorithm in $O(n)$ time. The set $\{P_i, Q_{i+1}\}$ with maximum $D(\{P_i, Q_{i+1}\})$ is clearly an optimal solution for the case with $k = 2$. Therefore, we have the following theorem.

Theorem 2. *It takes $O(n)$ time to compute a pair of disjoint segments of a length-n sequence, each has length at least ℓ, such that the sum of their densities is maximized.*

4 Our Algorithm for $k = 3$

Suppose that S_{o1}, S_{o2} and S_{o3} form an optimal set of segments for the case with $k = 3$. We first find a maximum-density segment $S_M = A_{m_i,m_j}$ in A. We also compute maximum-density segments $S_L = A_{l_i,l_j}$ in A_{1,m_i-1} and $S_R = A_{r_i,r_j}$ in $A_{m_j+1,n}$, respectively. Then we find the optimal two disjoint density segments $\{S_{L1}, S_{L2}\}$ in A_{1,m_i-1} and $\{S_{R1}, S_{R2}\}$ in $A_{m_j+1,n}$. Let $\{S_{M'}, S_{M''}\}$ be the element in

$$\{\{S_L, S_R\}, \{S_{L1}, S_{L2}\}, \{S_{R1}, S_{R2}\}\}$$

that has maximum sum of densities. Moreover, we find the maximum-density segment $S_{LL} = A_{ll_i,ll_j}$ in A_{1,l_i-1} and the maximum-density segment $S_{RR} = A_{rr_i,rr_j}$ in $A_{r_j+1,n}$. Furthermore, we find the maximum density segment S_{LLL} in A_{1,ll_i-1} and the maximum-density segment S_{RRR} in $A_{rr_j+1,n}$. For brevity, we use $S_x \sim S_y$ (respectively, $S_x \leftrightarrow S_y$) to denote that segments S_x and S_y overlap (respectively, are disjoint). Let U be the set of segments which are intersect to S_M with length from ℓ to $2\ell - 1$. Finally, for each segment S in U, we perform the following Algorithm 1 to find three disjoint segments $\{S_1, S_2, S_3\}$ with $\{S_1, S_2, S_3\} \cap S \neq \emptyset$.

Algorithm 1:

1. For each segment $S_v = A_{v_i,v_j}$ in U, let $S_2 = S_v$. **do**

1.1. (Case 1: $S_v \sim a_{m_i}$ but $S_v \leftrightarrow a_{m_j}$): Find the maximum-density segment $S_{R'}$ in $A_{v_j+1,m_j+2\ell-2}$. Then let $S_3 = S_{R'}$.
If $S_v \leftrightarrow S_L$ **then** $S_1 = S_L$
else
If $S_v \sim S_L$ but $S_v \leftrightarrow S_{LL}$ **then** find the maximum-density segment $S_{L'}$ in $A_{l_i-2\ell+2,v_i-1}$ then let S_1 be the maximum density segment between $S_{L'}$ and S_{LL}.
else find the maximum-density segment $S_{L'}$ in $A_{ll_i-2\ell+2,v_i-1}$ then let S_1 be the maximum density segment between $S_{L'}$ and S_{LLL}.

1.2. (Case 2: $S_v \sim a_{m_j}$ but $S_v \leftrightarrow a_{m_i}$): Find the maximum-density segment $S_{L''}$ in $A_{m_i-2\ell+2,v_i-1}$. Then let $S_1 = S_{L''}$.
If $S_v \leftrightarrow S_R$ **then** let $S_3 = S_R$
else
If $S_v \sim S_R$ but $S_v \leftrightarrow S_{RR}$ **then** find the maximum-density segment $S_{R''}$ in $A_{v_j+1,r_j+2\ell-2}$ then let S_3 be the maximum density segment between $S_{R''}$ and S_{RR}.
else find the maximum-density segment $S_{R''}$ in $A_{v_j+1,rr_j+2\ell-2}$ then let S_3 be the maximum density segment between $S_{R''}$ and S_{RRR}.

1.3. (Case 3: $S_v \subset S_m$): Find the maximum-density segments $S_{L'''}$ and $S_{R'''}$ in $A_{m_i-2\ell+2,v_i-1}$ and $A_{v_j+1,m_j+2\ell-2}$. Let $\{S_1, S_3\} = \{S_{L'''}, S_{R'''}\}$.

end for

2. Let $\{S_a, S_b, S_c\}$ be the maximum total density segments in all these three disjoint segments $\{S_1, S_2, S_3\}$.

Finally, if

$$D(\{S_a, S_b, S_c\}) \leq D(\{S_M, S_{M'}, S_{M''}\}),$$

then let $\{S_{o1}, S_{o2}, S_{o3}\}$ be $\{S_{M'}, S_M, S_{M''}\}$; otherwise, let $\{S_{o1}, S_{o2}, S_{o3}\}$ be $\{S_a, S_b, S_c\}$. Though there are $O(\ell^2)$ iterations in Algorithm 1, we only need $O(\ell^2)$ time in total. We can pre-process to find all $S_{R'}$ in case 1, all $S_{R'''}$ in case 3, all $S_{L''}$ in case 2 and all $S_{L'''}$ in case 3 in $O(\ell^2)$ time. Because the lengths of $A_{m_i-2\ell+2,v_i-1}$ and $A_{v_j+1,m_j+2\ell-2}$ are $O(\ell)$ and the length of S_M is at most $2\ell - 1$. Also pre-process to find all $S_{L'}$ in case 1 and all $S_{R''}$ in case 2 take $O(\ell^2)$ time. As a result, the time complexity of Algorithm 1 is $O(\ell^2)$.

Theorem 3. *It takes $O(n + \ell^2)$ time to compute three disjoint segments of a length-n sequence, each has length at least ℓ, such that the sum of their densities is maximized.*

Proof. Since the time complexity of Algorithm 1 is $O(\ell^2)$, our algorithm runs in $O(n + \ell^2)$ time. It remains to prove the correctness of our algorithm. For any three disjoint segments $\{S_1, S_2, S_3\}$ in A, we will show

$$D(\{S_{o1}, S_{o2}, S_{o3}\}) \geq D(\{S_1, S_2, S_3\}).$$

For convenience, let S_1 be the left segment, let S_2 be the middle segment, and let S_3 be the right segment for the three disjoint segments $\{S_1, S_2, S_3\}$ in A. First, if each of S_1, S_2 and S_3 does not overlap with S_M, then

$$D(\{S_M, S_{M'}, S_{M''}\}) \geq D(\{S_1, S_2, S_3\}).$$

If only one segment of $\{S_1, S_2, S_3\}$ overlaps with S_M, then

$$D(\{S_M, S_{M'}, S_{M''}\}) \geq D(\{S_1, S_2, S_3\}).$$

Hence, the rest of the proof assumes that at least two segments of $\{S_1, S_2, S_3\}$ overlaps with S_M and

$$D(\{S_1, S_2, S_3\}) > D(\{S_{M'}, S_M, S_{M''}\}).$$

Without loss of generality, we may assume that segment $S_2 = S_v = A_{v_i,v_j}$ overlaps with S_M. Then we consider the following three cases. Case 1: $S_v \sim a_{m_i}$ but $S_v \leftrightarrow a_{m_j}$, case 2: $S_v \sim a_{m_j}$ but $S_v \leftrightarrow a_{m_i}$, and case 3: $S_v \subset S_m$. We prove the result for case 1 and case 3. The case 2 can be shown similar to case 1. For case 1, let $S_{R'}$ is the maximum-density segment in $A_{v_j+1,m_j+2\ell-2}$ and $S_3 = S_{R'}$. Because $d(S_1) \leq d(S_L)$ and $d(S_2) \leq d(S_M)$, the segment S_3 must be a subsequence in $A_{v_j+1,m_j+2\ell-2}$; otherwise, we have

$$D(\{S_L, S_M, S_R\}) \geq D(\{S_1, S_2, S_3\}).$$

Hence, we only choose a best S_1 in A_{1,v_i-1}. We consider the following three cases. (1) if $S_v \leftrightarrow S_L$, we only let $S_1 = S_L$ because S_L is the maximum-density segment in A_{1,m_i-1}. (2) If $S_v \sim S_L$ but $S_v \leftrightarrow S_{LL}$. For S_1, we only consider the segments S_{LL} and $S_{L'}$, where $S_{L'}$ is a maximum-density segment in $A_{l_i-2\ell+2,v_i-1}$. Because $S_1 \sim S_L$, segment S_1 is either in A_{1,l_i-1} or in $A_{l_i-2\ell+2,v_i-1}$. (3) $S_v \sim S_L$ and S_{LL}. For S_1, we only consider the segments S_{LLL} and $S_{L'}$, where $S_{L'}$ is a maximum-density segment in $A_{ll_i-2\ell+2,v_i-1}$. Because $S_1 \sim S_{LL}$, segment S_1 is either in A_{1,ll_i-1} or in $A_{ll_i-2\ell+2,v_i-1}$. For case 3, let $S_{L'''}$ is the maximum-density segment in $A_{m_i-2\ell+2,v_i-1}$ and $S_{R'''}$ is the maximum-density segment in $A_{v_j+1,m_j+2\ell-2}$. Because $d(S_v) \leq d(S_M)$, we only let $\{S_1, S_2, S_3\} = \{S_{L'''}, S_v, S_{R'''}\}$. Otherwise, we have

$$D(\{S_L, S_M, S_R\}) \geq D(\{S_1, S_2, S_3\}).$$

□

5 Conclusion

We have shown the first known polynomial-time algorithm to compute multiple disjoint segments whose sum of densities is maximized. An immediate open question is whether the problem can be solved in $o(n\ell k)$ time. Also, it would be interesting to see our techniques for $k = 2, 3$ to be generalized to the cases with larger k.

References

1. P. Berman, P. Bertone, B. DasGupta, M. Gerstein, M.-Y. Kao and M. Snyder: Fast Optimal Genome Tiling with Applications to Microarray Design and Homology Search. *Journal of Computational Biology*, 11:766–785, 2004.
2. K.-M. Chung and H.-I. Lu: An Optimal Algorithm for the Maximum-Density Segment Problem. *SIAM Journal on Computing*, 34:373–387, 2004.
3. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison: *Biological Sequence Analysis*. Cambridge University Press, 1998.
4. T.-H. Fan, S. Lee, H.-I. Lu, T.-S. Tsou, T.-C. Wang, A. Yao: An Optimal Algorithm for Maximum-Sum Segment and Its Application in Bioinformatics. In *Proceedings of the 8th International Conference on Implementation and Application of Automata*, Lecture Notes in Computer Science 2759, 251–257, Santa Barbara, July 2003, Springer-Verlag.
5. M. Gardiner-Garden, and M. Frommer: CpG Islands in Vertebrate Genomes. *Journal of Molecular Biology*, 196:261–282, 1987.
6. M.H. Goldwasser, M.-Y. Kao, and H.-I. Lu: Linear-Time Algorithms for Computing Maximum-Density Sequence Segments with Bioinformatics Applications. *Journal of Computer and System Sciences*, 70:128–144, 2005.
7. U. I. Gupta, D. T. Lee, and J. Y.-T. Leung: Efficient Algorithms for Interval Graphs and Circular-Arc Graphs, *Networks* 12:459–467, 1982.
8. J. Y. Hsiao, C. Y. Tang, and R. S. Chang: An Efficient Algorithm for Finding a Maximum Weight 2-Independent Set on Interval Graphs, *Information Processing Letters*, 43(5):229-235, 1992.
9. X. Huang: An algorithm for Identifying Regions of a DNA Sequence That Satisfy a Content Requirement. *Computer Applications in the Biosciences*, 10:219–225, 1994.
10. S. K. Kim: Linear-Time Algorithm for Finding a Maximum-Density Segment of a Sequence. *Information Processing Letters*, 86:339-342, 2003.
11. F. Larsen, R. Gundersen, R. Lopez, and H. Prydz: CpG Islands as Gene Marker in the Human Genome. *Genomics*, 13:1095–1107, 1992.
12. Y.-L. Lin, T. Jiang, K.-M. Chao: Efficient Algorithms for Locating the Length-Constrained Heaviest Segments with Applications to Biomolecular Sequence Analysis. *Journal of Computer and System Sciences*, 65:570–586, 2002.
13. Y.-L. Lin, X. Huang, T. Jiang, K.-M. Chao: MAVG: Locating Non-overlapping Maximum Average Segments in a Given Sequence. *Bioinformatics*, 19:151–152, 2003.
14. A. Nekrutenko and W.-H. Li: Assessment of Compositional Heterogeneity within and between Eukaryotic Genomes. *Genome Research*, 10:1986–1995, 2000.
15. P. Rice, I. Longden, and A. Bleasby: EMBOSS: The European Molecular Biology Open Software Suite. *Trends in Genetics*, 16:276–277, 2000.
16. P. H. Sellers: Pattern Recognition in Genetic Sequences by Mismatch Density. *Bulletin of Mathematical Biology*, 46:501–514, 1984.
17. N. Stojanovic, L. Florea, C. Riemer, D. Gumucio, J. Slightom, M. Goodman, W. Miller, and R. Hardison: Comparison of Five Methods for Finding Conserved Sequences in Multiple Alignments of Gene Regulatory Regions. *Nucleic Acids Research*, 27:3899–3910, 1999.
18. D. Takai, and P.A. Jones: Comprehensive Analysis of CpG Islands in Human Chromosomes 21 and 22. *Proceedings of the National Academy of Sciences*, 99:3740–3745, 2002.