

# Fast and Complete Search of siRNA Off-Target Sequences

Hong Zhou<sup>1</sup>, Yufang Wang<sup>2</sup>, and Xiao Zeng<sup>3</sup>

<sup>1</sup>Saint Joseph College, West Hartford, CT, USA.

<sup>2</sup>University of Southern Mississippi, Hattiesburg, MS, USA.

<sup>3</sup>Superarray Bioscience Corporation, Frederick, MD, USA.

**Abstract** –Smith-Waterman alignment algorithm is favored in search for siRNA off-target instead of the BLAST algorithm, because BLAST tends to overlook some significant homologous sequences, especially when they are short (21 nt~27 nt). Smith-Waterman algorithm, however, suffers from its own shortcomings, especially its inefficiency in searching through a large sequence database. This paper presents a two-phase homology search strategy that preserves the strength of Smith-Waterman alignment algorithm while shortening its running time. In the first phase of this algorithm, selected siRNA sequence is divided into multiple mutually disjoint substrings, each of which is used to scan the sequence database for perfect matches against other genes. Only the sequences that have perfect match to substrings (of a given siRNA) are kept for the second phase. The second phase is the bona fide Smith-Waterman procedure. During this phase, the algorithm only checks the local vicinity sequences where a substring lands on a perfect match. This two-phased arrangement of the algorithm significantly improves the efficiency of the original Smith-Waterman algorithm by concentrating the search on localized regions instead of the whole genome sequence.

**Key words:** siRNA, Sequence Alignment, Off-Target, Smith-Waterman.

## 1 Introduction

The design of effective small interfering RNA (siRNA) is playing a central role in the applications of RNA interference (RNAi) technique in biological studies. This design process is usually fulfilled by computer algorithms [2, 3, 6, 8, 11, 12, 16, 17]. A critical requirement in siRNA design is to guarantee that the designed siRNA sequences are free of off-target effect. Although the actual mechanism of off-target effect is still unknown, it has been demonstrated that a partial sequence homology between siRNA and its unintended targets is one of the contributing factors [4, 10, 13]. It has

been suggested that if an introduced siRNA has less than 3 mismatches with an unintended mRNA, it would likely knock down the expression of this mRNA in addition to its intended target which shares 100% homology with this siRNA sequence [5, 8]. Currently, most available siRNA design tools use BLAST to identify siRNA candidates that may cause off-target effect. BLAST, although fast, is not the best algorithm designed for this type of task since it overlooks significant sequence homologies [8, 15, 18]. As an alternative, Smith-Waterman search algorithm has been employed by some design tools to identify all possible off-target sequences [8, 18].

Smith-Waterman algorithm [14] utilizes a dynamic programming approach to identify the local optimal alignment between two sequences. It guarantees to locate the existing optimal alignment based on a scoring system with a set of scores assigned to a match, a substitution, a deletion, and an insertion. Given two sequences with length of  $m$  and  $n$ , the computational complexity of Smith-Waterman algorithm is  $O(mn)$ . Since the off-target search for siRNA sequences must be conducted completely through a given sequence database (which is usually large), the Smith-Waterman algorithm alone becomes very time-consuming and impractical for this task. Derived from dynamic programming, BLAST and FASTP improved the searching efficiency greatly by using a pre-constructed lookup table to find the locally similar regions between two sequences [1, 7, 9]. In BLAST, for example, given a sequence of  $m$  letters long, this sequence can have  $m-w+1$  contiguous words each with  $w$  letters in length. These words can be scanned through the lookup table to find statistically significant word matches upon which a final alignment can be achieved. Though BLAST is very fast, the tradeoff for its efficiency is that it sometimes overlooks homologous sequences that can cause off-target effect [8, 15, 18]. In this paper, we present a two-phase homology search algorithm for siRNA off-target search that combines the effectiveness of BLAST concept and the thoroughness of Smith-Waterman algorithm. In this algorithm, we first divide the siRNA sequence into multiple mutually

disjoint substrings based on the mismatch cut-off for an off-target homology (a mismatch is defined to be either a substitution, a deletion or an insertion hereafter). An off-target homology is defined to be an off-target sequence that has less than a predefined number of mismatches (mismatch cut-off) with the siRNA sequence. When and only when a substring of the siRNA sequence finds a perfect match, Smith-Waterman dynamic programming is used to examine the local vicinity region of the matched sequence. This algorithm does not construct any lookup table from the whole genome sequences, though it significantly improves the searching efficiency by guiding the most time-consuming core Smith-Waterman alignment on the local regions that need to be further examined. Yet unlike the BLAST algorithm, this algorithm does not miss any homologous sequences in siRNA that may cause off-target effect.

## 2 The Two-Phase Algorithm

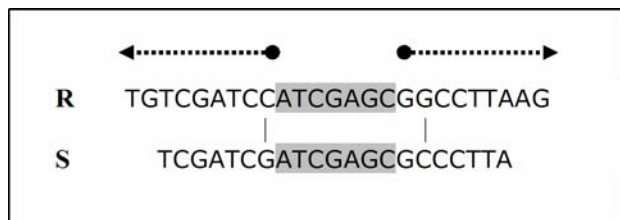
As a dynamic programming approach, Smith-Waterman algorithm finds the optimal alignment gradually by means of simple recurrences. To complete the alignment process between the two sequences of length  $m$  and  $n$ , a table of size  $m \times n$  must be constructed. Thus, the computational complexity is of order  $O(mn)$ . When searching for possible off-target alignment, a siRNA sequence must be aligned against all expressed sequences (mRNAs) except the target gene itself (or gene slice variants). In our current human mRNA RefSeq database, there are 28162 non-redundant sequences with an average length of 2589bp. Therefore, the computational cost for a 21 nt siRNA sequence would be at least  $21 \times 2589 \times (28162 - 1)$  (suppose only aligning the sense strand of the siRNA sequence against other mRNA sequences). This prohibitory computational cost makes it almost impossible to incorporate Smith-Waterman algorithm into routine siRNA design. To make Smith-Waterman algorithm practical in siRNA design, we propose a modified application of the algorithm, which was motivated by the following realization.

For a siRNA sequence of length  $m$ , an off-target homology is defined as a sequence that has less than  $x$  mismatches when aligned against the siRNA sequence. Thus, after the siRNA sequence is divided into  $x$  equal substrings (as equal as possible), at least one substring must have a perfect match with the off-target region. For the remainder of this paper, let's assume  $m=21$  and  $x=3$  unless stated otherwise. Under this condition, a homology can only have maximum two mismatches, i.e., 0, 1, or 2 mismatches. When there are maximum two mismatches, no matter where the possible two mismatches are, at least one third of the siRNA sequence must have exact match with the homological region as shown in figure 1.



**Fig. 1.** When there are 2 mismatches between the siRNA sequence (S) and the off-target region (R), at least one of the three substrings of S has exact match with R. The vertical bars mark the mismatches and the shaded substrings have the exact match.

With this insight, we envision that the off-target search process can then be divided into two phases. The first phase is a simple string matching procedure that finds the potential regions with which at least one of the substrings of the siRNA sequence finds an exact match. The second phase calls for the Smith-Waterman procedure to evaluate the best alignment between potential regions and the siRNA sequence. The potential off-target region is extended around the short sequence with which one substring of the siRNA sequence has exact match. For example, if it is the leftmost substring of the siRNA sequence that has exact match with the region, the region should be extended to the right for enough base pairs. However, if it is the middle substring, the region should then be extended to both the left and right for enough base pairs so that the potential off-target region completely covers the siRNA sequence regarding all the cases of  $(x-1)$  mismatches. Figure 2 shows the case when the middle substring has the exact match.



**Fig. 2.** When the substring in the middle has the exact match, the off-target region must be such a region that extends from the matched substring to both left and right enough base pairs to completely cover the siRNA sequence. The vertical bars mark the mismatches and the shaded region has the exact match. Please observe that it is necessary to extend 2 base pairs over the left and right ends of the siRNA sequence since there might be 2 insertions in the off-target region.

## 3 The Computational Complexity

The total computational complexity of the proposed two-phase algorithm is the sum of the computational cost of the two phases. For the first phase, various string matching algorithms can be employed. As our implementation is in computer programming language Java, Java's built-in string matching algorithm is used, which is a character by character matching procedure. To

search through a gene sequence, the maximum computational cost of this approach is of order  $x\left(\frac{m}{x}n\right)$  where  $m$  is the length of the siRNA sequence,  $n$  is the length of the searched gene sequence,  $x$  is the number of substrings, and  $\frac{m}{x}$  annotates the substring length. If we assume that each of the four different bases AGCT has equal probability to appear at any a given nucleotide position, then the probability for  $i$  consecutive bases to be matched is  $\left(\frac{1}{4}\right)^i$ . However, in the character by character matching procedure, the second base is required to be compared only if the first base is matched, and the third base is required for comparison only if the first two bases are matched, and so on. Thus, the probability of comparing two bases of the substring is  $\frac{1}{4}$  while the probability of comparing three bases of the substring is  $\frac{1}{16}$ . Therefore the total computational cost for the first phase would be  $xn\left(1 + \frac{1}{4} + \frac{1}{16} + \dots + \left(\frac{1}{4}\right)^{\frac{m}{x}-1}\right)$ , which is bounded by  $\frac{4}{3}xn$ . The second phase is Smith-Waterman alignment algorithm conducted between the siRNA sequence and the potential off-target region whose maximum length is  $x-1+m+x-1=2(x-1)+m$ . The computational cost of this alignment procedure is therefore  $m(2(x-1)+m)$ . However, the probability of finding a potential off-target region is only  $x\left(\frac{1}{4}\right)^{\frac{m}{x}}$ , so the actual computational cost of the second phase is  $x\left(\frac{1}{4}\right)^{\frac{m}{x}}[m(2(x-1)+m)]$ . The total computational cost of this algorithm can then be expressed as

$$\frac{4}{3}xn + kxm\left(\frac{1}{4}\right)^{\frac{m}{x}}(2(x-1)+m), \quad (1)$$

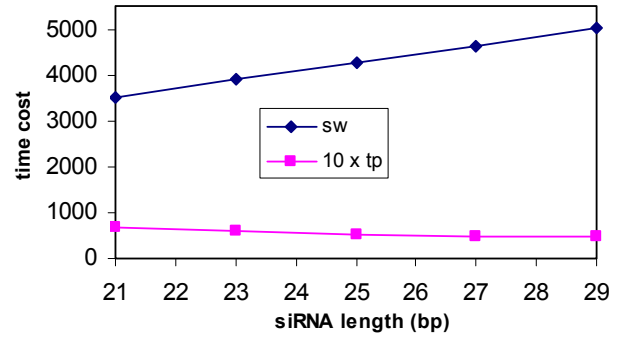
where the constant  $k$  denotes the extra complexity of the Smith-Waterman algorithm compared to the simple character-by-character string matching algorithm.

Equation (1) shows that the computational cost of the algorithm is linearly proportional to the number of substrings  $x$ . Specifically, when  $x=3$  and  $m=7$ , compared to the original Smith-Waterman alignment approach with computational complexity of  $O(mn)$ , theoretically the computational cost is reduced about 5 times. Practically, due to the simplicity of the character by character matching procedure, the efficiency improvement is much higher.

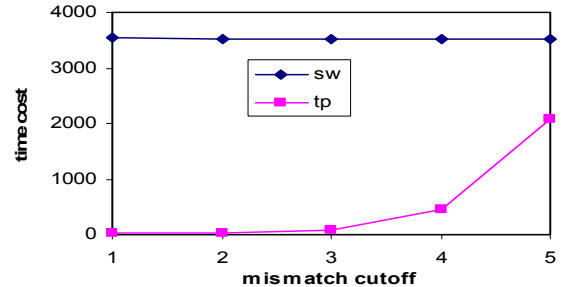
## 4 Results and Discussion

The computational cost of Smith-Waterman alignment algorithm is linearly proportional to the length of siRNA sequence  $m$ . However, regarding the two-phase

algorithm, as shown in equation (1), when  $x$  is fixed, larger  $m$  exponentially reduces the probability of phase two operations while it only linearly increases the computational cost of phase two operations, which can slightly improve the overall efficiency. As elucidated in figure 3 where  $x=3$ , the computational cost of Smith-Waterman alignment algorithm is linearly increased along with the siRNA length, while the computational cost of the two-phase algorithm is slightly decreased (all experimental results discussed in this paper were obtained using the complete collection of human mRNAs in the NCBI RefSeq database, human genome build 35.1). Figure 3 also demonstrates the 50 – 100 times of efficiency gain of the two-phase algorithm.



**Fig. 3.** The two-phase algorithm can gain 50 – 100 times in efficiency. sw: Smith-Waterman algorithm alone. tp: the proposed two-phase algorithm. The time cost value of the two-phase algorithm is multiplied 10 times for a better visual comparison.



**Fig. 4.** The value of  $x$  impacts the efficiency of the two-phase algorithm. sw: Smith-Waterman algorithm alone. tp: the proposed two-phase algorithm.

As each substring must be searched for exact alignment,  $x$ , the number of substrings, is in fact determining the number of repetitions of the first phase procedure. In addition, when  $m$  is fixed,  $x$  also determines the substring lengths. A larger  $x$  value means shorter substrings and increases the probability of phase two operations. As shown in figure 4 where  $m=21$ , while  $x$

has little impact on the original Smith-Waterman alignment algorithm, a larger  $x$  value does increase the computational cost of the proposed algorithm. When  $x=5$ , the efficiency gain of the proposed algorithm is less than 50%.

## 5 Conclusion

The efficiency gain of the two-phase algorithm is achieved by setting up a first phase filter that relieves the burden for the most inefficient but reliable Smith-Waterman alignment algorithm. Using siRNA design tool as an example here, we have demonstrated that the two-phase algorithm can achieve efficiency gain up to two orders of magnitude over the original Smith-Waterman alignment algorithm alone. Thus, the two-phase algorithm can be incorporated in the routine search for potential off-target region in siRNA design, a task that BLAST algorithm can not fulfill satisfactorily.

## 6 Acknowledgment

The authors would like to thank Dr. Joseph Manthey for his valuable discussion about the computational cost analysis and his critical reading of this manuscript.

## 7 References

- [1] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., Lipman, D. J., "Basic local alignment search tool," *J Mol Biol.*, vol. 215, pp.403–410, 1990.
- [2] Cui, W., Ning, J., Naik, U. P., Duncan, M. K., "OptiRNAi, an RNAi design tool," *Comput Methods Programs Biomed.*, vol.75, pp.67–73, 2004.
- [3] Henschel, A., Buchholz, F., Habermann, B., "DEQOR: a web-based tool for the design and quality control of siRNAs," *Nucleic Acids Res.*, vol.32, pp.W113–W120, 2004.
- [4] Jackson, A. L., Bartz, S. R., Schelter, J., Kobayashi, S. V., Burchard, J., Mao, M., Li, B., Cavet, G., "Expression profiling reveals off-target gene regulation by RNAi," *Nat Biotechnol.*, vol.21, pp.635–637, 2003.
- [5] Kim, D. H., Behlke, M. A., Rose, S. D., Chang, M. S., Choi, S., Rossi, J. J., "Synthetic dsRNA Dicer substrates enhance RNAi potency and efficacy," *Nat Biotechnol.*, vol.23, pp.222–226, 2005.
- [6] Levenkova, N., Gu, Q., Rux, J. J., "Gene specific siRNA selector," *Bioinformatics*, vol.20, pp.430–432, 2004.
- [7] Lipman, D. J., Pearson, W. R., "Rapid and sensitive protein similarity searches," *Science.*, vol.227, pp.1435–1441, 1985.
- [8] Naito, Y., Yamada, T., Ui-Tei, K., Morishita, S., Saigo, K., "siDirect: highly effective, targetspecific siRNA design software for mammalian RNA interference," *Nuclear Acids Research*, vol.32, pp.W124–129, 2004.
- [9] Pearson, W. R., Lipman, D. J., "Improved tools for biological sequence comparison," *PNAS*, vol.85, pp.2444–2448, 1988.
- [10] Persengiev, S. P., Zhu, X., Green, M. R., "Nonspecific, concentration-dependent stimulation and repression of mammalian gene expression by small interfering RNAs (siRNAs)," *RNA*, vol.10, pp.12–18, 2004.
- [11] Reynolds, A., Leake, D., Boese, Q., Scaringe, S., Marshall, W. S., Khvorova, A., "Rational siRNA design for RNA interference," *Nat Biotechnol.*, vol.22, pp.326–330, 2004.
- [12] Sætrom, P., Snove, O. Jr., "A comparison of siRNA efficacy predictors," *Biochemical and Biophysical Research Communications*, vol.321, pp.247–253, 2004.
- [13] Scacheri, P. C., Rozenblatt-Rosen, O., Caplen, N. J., Wolfsberg, T. G., Umayam, L., Lee, J. C., Hughes, C. M., Shanmugam, K. S., "Short interfering RNAs can induce unexpected and divergent changes in the levels of untargeted proteins in mammalian cells," *PNAS*, vol.101, pp.1892–1897, 2004.
- [14] Smith, T. F., Waterman, M. S., "Identification of common molecular subsequences," *J Mol Biol.*, vol.147, pp.195–197, 1981.
- [15] Snove, O., Jr., Holen, T., "Many commonly used siRNAs risk off-target activity," *Biochem Biophys Res Commun.*, vol.319, pp.256–263, 2004.
- [16] Ui-Tei, K., Naito, Y., Takahashi, F., Haraguchi, T., Ohki-Hamazaki, H., Juni, A., Ueda, R., Saigo, K., "Guidelines for the selection of highly effective siRNA sequences for mammalian and chick RNA interference," *Nucleic Acids Res.*, vol.32, pp.936–948, 2004.
- [17] Yuan, B., Latek, R., Hossbach, M., Tuschl, T., Lewitter, F., "siRNA Selection Server: an automated siRNA oligonucleotide prediction server," *Nucl. Acids Res.*, vol.32, pp.W130–W134, 2004.
- [18] Zhou, H., Zeng, X., Wang, Y., Seyfarth, B. R., "A three-phase algorithm for computer aided siRNA design," *Informatica (Slovene)*, to appear.