Romanian Biotechnological Letters *Vol. 18, No.1, 2013*
Copyright © 2013 University of Bucharest Printed in Romania. All rights reserved
SHORT COMMUNICATION

FLAG: Fast Local Alignment Generating Methodology

Received for publication, August 15, 2012
Accepted, November 20, 2012

DONE STOJANOV, SAŠO KOCESKI, ALEKSANDRA MILEVA
Faculty of Computer Science, University "Goce Delcev"- Štip, Republic of Macedonia
Email: done.stojanov@ugd.edu.mk

Abstract

A new, time and space efficient alignment methodology is presented, applicable on similar nucleotide sequences. Linear time complexity–O(m), has been determined when aligning approximately same size similar sequences. Time complexity improvement is due to the methodology, according which alignments are generated and the significant space's reduction, where the search for alignments is carried out.

Keywords: linear time and space, un-gapped, local alignment, methodology

Introduction

The time inefficiency has been the major disadvantage of local pairwise alignment techniques. Smith–Waterman's algorithm (T. SMITH & al. [1]) requires fixed *O(nm)* time, identifying one optimal (score maximized alignment), allowing gaps insertion. Instead of finding one optimal alignment, M. WATERMAN and M. EGGERT [2] came up with an idea of identifying *k* suboptimal local alignments. The main disadvantage of Waterman–Eggert's algorithm is again the nonlinear time complexity. In order to reduce the space complexity of Waterman–Eggert's algorithm, X. HUANG and W. MILLER [3] in 1991 presented a linear space solution of Waterman–Eggert's algorithm, being until then the space cheapest local alignment technique.

Newly heuristic ultrafast solutions, such as: FASTA (D. LIPMAN & al. [4]) and BLAST (S. ALTSCHUL & al. [5]), are applicable for fast search of large genetic database, identifying similar sequences regarding referent sequence, not always finding the optimal solution.

Despite the time complexity, space complexity is often found as a limiting factor when aligning large nucleotide sequences. In order to reduce space complexity of an alignment, methodology presented in (D. STOJANOV & al. [6]) represents each region of consecutive matching nucleotides with a triple, identifying region's length and starting positions at the sequences. Based on this representation, measurements performed in [6] clearly show that linear space is required, while the time complexity is $O(nm^2)$. Most of the time in [6] is wasted on examining all combinations of un-gapped local alignments within overlapping sections and the number of alignments being examined.

When aligning similar nucleotide sequences, large regions of consecutive matching nucleotides are part of the optimal un-gapped local alignment. Also, the probability of finding an optimal un-gapped local alignment within *m*–nucleotides long overlapping sections is higher than the probability of finding it in overlapping sections with less than *m* nucleotides, where *m* is the length of the smaller nucleotide sequence, subject of an alignment.

Based on the previous, fast local alignment generating methodology is presented, requiring linear time and space – *O(m)*, when aligning approximately same size, similar nucleotide sequences.

Materials and methods

Methodology

As Table 1 shows, the smaller nucleotide sequence b overlaps n-m+1 different sequence a sections with length m and sections with length less than m. Overlapping sections with length less than the length of the smaller sequence - b, are formed by left and right one place sequence b shifts, out of the length of the sequence a.

Table 1. Overlapping sections

Nucleotide sequences: $a = a_0a_1\ldots a_{n-2}a_{n-1}$, $b = b_0b_1\ldots b_{m-2}b_{m-1}$		
m – nucleotides long overlapping sections	**sequence b left shifted overlapping sections**	**sequence b right shifted overlapping sections**
$\boxed{\begin{array}{l}a_0a_1\ldots a_{m-2}a_{m-1}\\ b_0b_1\ldots b_{m-2}b_{m-1}\end{array}}\begin{array}{l}\ldots a_{n-1}\\ \end{array}$ $a_0\boxed{\begin{array}{l}a_1\ldots\ldots a_{m-1}a_m\\ b_0b_1\ldots b_{m-2}b_{m-1}\end{array}}\begin{array}{l}\ldots a_{n-1}\\ \end{array}$ $\ldots\ldots\ldots\ldots\ldots\ldots$ $\begin{array}{l}a_0a_1\ldots\\ \end{array}\boxed{\begin{array}{l}a_{n-m}\ldots\ldots\ldots\ldots a_{n-1}\\ b_0b_1\ldots b_{m-2}b_{m-1}\end{array}}$	$\begin{array}{r}\\ \leftarrow b_0\end{array}\boxed{\begin{array}{l}a_0a_1\ldots\ldots a_{m-2}\\ b_1\ldots b_{m-2}b_{m-1}\end{array}}\begin{array}{l}a_{m-1}\ldots a_{n-1}\\ \end{array}$ $\begin{array}{r}\\ \leftarrow b_0b_1\end{array}\boxed{\begin{array}{l}a_0a_1\ldots\ldots a_{m-3}\\ b_2\ldots b_{m-2}b_{m-1}\end{array}}\begin{array}{l}a_{m-2}\ldots a_{n-1}\\ \end{array}$ $\ldots\ldots\ldots\ldots\ldots\ldots$	$\begin{array}{l}a_0\ldots\\ \end{array}\boxed{\begin{array}{l}a_{n-m+1}\ldots\ldots a_{n-2}a_{n-1}\\ b_0\ldots\ldots\ldots b_{m-3}b_{m-2}\end{array}}\begin{array}{l}\\ b_{m-1}\rightarrow\end{array}$ $\begin{array}{l}a_0\ldots\\ \end{array}\boxed{\begin{array}{l}a_{n-m+2}\ldots a_{n-2}a_{n-1}\\ b_0\ldots\ldots b_{m-4}b_{m-3}\end{array}}\begin{array}{l}\\ b_{m-2}b_{m-1}\rightarrow\end{array}$ $\ldots\ldots\ldots\ldots\ldots\ldots$

When comparing parallel nucleotides within overlapping sections, $\chi, \chi \geq 0$ regions composed of consecutive matching nucleotides, with at least one match, are found. As we have shown in [6], each matching region can be represented with a triple *R:*(p_b, p_a, l), where p_b is region's starting position at the sequence b, p_a is region's starting position at the sequence a, while l is region's length.

An un-gapped local alignment consists of one or more matching regions, separated with region(s) of mismatching nucleotides. The same alignment's score metrics, which has been used in [6], will be also used here, awarding positive score μ for each nucleotide match, while penalizing each nucleotide mismatch with negative score $-\delta$. Each alignment *A*:($R_1R_2\ldots R_{k-1}R_k$) is assigned a unique score, computed with the formula presented in [6]:

$$f(A : R_1R_2\ldots R_{k-1}R_k) = \mu\sum_{i=1}^{k} len(R_i) - \delta\sum_{j=2}^{k} dif(R_j, R_{j-1})$$

where: $len(R_i)$ is the length of the matching region R_i, while $dif(R_j, R_{j-1})$ is the number of mismatching nucleotides, separating matching regions R_j and R_{j-1}.

Alignments are formed according to the following strategy – ***fast local alignment generating methodology***, generating as an output score maximized un-gapped local alignment A, regarding the longest matching region within overlapping sections, where χ matching regions have been found:

Find the longest matching region: $R_\varsigma = \max len\{R_1, R_2, \ldots, R_{\chi-1}, R_\chi\}$, $1 \leq \varsigma \leq \chi$. *Take initially region* R_ς *as extending and local alignment:* $A_e \leftarrow R_\varsigma$, $A \leftarrow R_\varsigma$.

If $\varsigma = 1$, *extend* A_e, *appending regions* R_ξ, $A_e \leftarrow A_e \rhd\lhd R_\xi$, *consecutively for* $\xi = 2,3,\ldots,\chi-1,\chi$. *If* $f(A_e) > f(A)$, *then:* $f(A) \leftarrow f(A_e)$, $A \leftarrow A_e$.

If $\varsigma = \chi$, *extend* A_e, *appending regions* R_ξ, $A_e \leftarrow R_\xi \rhd\lhd A_e$, *consecutively for* $\xi = \chi-1, \chi-2, \ldots, 2, 1$. *If* $f(A_e) > f(A)$, *then:* $f(A) \leftarrow f(A_e)$, $A \leftarrow A_e$.

If $1 < \varsigma < \chi$, *extend* A_e, *appending left positioned regions* R_ξ, $A_e \leftarrow R_\xi \rhd\lhd A_e$, *consecutively for* $\xi = \varsigma - 1, \varsigma - 2, ..., 2, 1$. *If f(A$_e$)>f(A), then:* $f(A) \leftarrow f(A_e)$, $A \leftarrow A_e$. *Take the local alignment found at this stage as extending alignment,* $A_e \leftarrow A$, *now being subject of right positioned extension, appending regions* R_ξ, $A_e \leftarrow A_e \rhd\lhd R_\xi$, *consecutively for* $\xi = \varsigma + 1, \varsigma + 2, ..., \chi - 1, \chi$. *If f(A$_e$)>f(A), then:* $f(A) \leftarrow f(A_e)$, $A \leftarrow A_e$.

If $f_{\max}$ is a score of the highest scoring un-gapped local alignment, found within *m*–nucleotides long overlapping sections, also has to be checked whether exists an alignment with higher score than $f_{\max}$, within overlapping sections with less than *m* nucleotides, formed by left one place sequence *b* shifts, out of the length of the sequence *a*.

Proposition 1.1: An alignment with a higher score than $f_{\max}$, could be found within sequence *b* left shifted overlapping sections, with lengths ranging between: *m*-1 and $[f_{\max} / \mu] + 1$, including those values.

Proof: Within overlapping sections of length *l*, the maximum possible score of an un-gapped local alignment is $l\mu$. Accordingly, a higher score alignment than $f_{\max}$ could be found if $l\mu > f_{\max}$, where from we get that $l > f_{\max} / \mu \geq [f_{\max} / \mu]$.

According Proposition 1.1, there is no need for search of an alignment with a higher score than $f_{\max}$, within sequence *b* left shifted overlapping sections, with lengths less than $[f_{\max} / \mu] + 1$, once an alignment with a highest score $f_{\max}$, has been found within *m*–nucleotides long overlapping sections.

Proposition 1.2: An alignment with a higher score than $f_{\max}$, could be found within sequence *b* right shifted overlapping sections, with lengths ranging between: *m*-1 and $[f_{\max} / \mu] + 1$, including those values, if $f_{\max}$ is a score of the optimal(highest scoring) alignment, found after examining alignments within *m*–nucleotides long overlapping sections and sequence *b* left shifted overlapping sections, according to the fast local alignment generating methodology.

The proof of Proposition 1.2 is analogous to the proof of Proposition 1.1. While searching for the optimal un-gapped local alignment, within *m*–nucleotides long overlapping sections, left and right shifted overlapping sections, data vector – identifying current un-gapped local alignment with a highest score, is kept in the memory. Vector's content dynamically updates if a new, higher scoring un-gapped local alignment than the current highest one is found. The last update of this vector identifies the optimal un-gapped local alignment.

An example

Fast local alignment generating methodology will be demonstrated on a concrete example, taking nucleotide sequences: *a*: TGCTAACTTTGATTGCCTA and *b*: TGAATCCCTTGAATGAAC as samples. Since the length of the sequence *a* is 19, while the length of the sequence *b* is 18, sequence *b* overlaps *n*-*m*+1=19-18+1=2 different sequence *a* sections with length 18. Alignments within overlapping sections are generated according to the fast local alignment generating methodology – Table 2, awarding +2 for each nucleotide match, while penalizing each nucleotide mismatch with -1.

Table 2. Examining alignments within 18–nucleotides long overlapping sections

18–nucleotides long overlapping sections	**matching region(s), found / region's score**	**local alignment found according the fast local alignment generating methodology/alignment's score**
TGCTAACTTTGATTGCCTA TGAATCCCTTGAATGAAC	$R_1 : (0,0,2)$ / $f(R_1) = 4$ $R_2 : (6,6,1)$ / $f(R_2) = 2$	CTTTGATTG CCTTGAATG

	$R_3:(8,8,4)\ /\ f(R_3)=8$ $R_4:(13,13,2)\ /\ f(R_4)=4$	$f(A:R_2R_3R_4)=12$
TGCTAACTTTGATTGCCTA TGAATCCCTTGAATGAAC	$R_1:(3,4,1)\ /\ f(R_1)=2$ $R_2:(5,6,1)\ /\ f(R_2)=2$ $R_3:(8,9,1)\ /\ f(R_3)=2$	AAC ATC $f(A:R_1R_2)=3$

When comparing parallel nucleotides within the first overlapping sections, four matching regions are found. There are two matching nucleotides within the first region, one matching nucleotide within the second region, four matching nucleotides within the third region, while the number of matching nucleotides within the fourth region is two. The third matching region is the longest one, initially taken as extending and local alignment: $A_e \leftarrow R_3, A \leftarrow R_3$. A_e is left extended, $A_e \leftarrow R_2 \rhd\lhd A_e : R_2R_3$, resulting with an alignment with score 9. Since extended alignment's score is higher than the score of the local alignment, local alignment A is updated with A_e, $A \leftarrow A_e : R_2R_3$. Further left positioned extension of A_e results with an alignment: $A_e \leftarrow R_1 \rhd\lhd A_e : R_1R_2R_3$, with score 9. Currently extended alignment's score equals local alignment's score, causing no change of the local alignment and its score.

Optimal left extended alignment, regarding the longest matching region, is $A{:}R_2R_3$. Now this alignment is taken as extending alignment, being subject of right positioned extension, $A_e \leftarrow A : R_2R_3$. After appending R_4, an alignment: $A_e \leftarrow A_e \rhd\lhd R_4 : R_2R_3R_4$, with score 12, is obtained. Extended alignment's score is higher than the score of the local alignment A, causing local alignment's update with A_e, $A \leftarrow A_e : R_2R_3R_4$.

Within the second overlapping sections, three matching regions, with one matching nucleotide, are found. The local alignment according to the fast local alignment generating methodology is $A{:}R_1R_2$, with score 3. Local alignment within the second overlapping sections is not higher scoring than the local alignment found within the first overlapping sections, whereby we can conclude that the optimal un-gapped local alignment, found within 18–nucleotides long overlapping sections is $A{:}R_2R_3R_4$=(6, 6, 1, 8, 8, 4, 13, 13, 2), with score 12.

According Proposition 1.1, an alignment with a higher score than 12, might exist within left shifted overlapping sections, with lengths between 11 and 7. Examining alignments within left shifted overlapping sections, according to the fast local alignment generating methodology, no alignment with a higher score is found.

Finally according Proposition 1.2, alignments within right shifted overlapping sections, with lengths ranging between 11 and 7, are examined. Since also within those overlapping sections, no alignment with a higher score is found, un-gapped local alignment found within the first 18–nucleotides long overlapping sections: $A{:}R_2R_3R_4$=(6, 6, 1, 8, 8, 4, 13, 13, 2), is the optimal one being found.

Table 3. Sequence *b* left and right shifted overlapping sections

Sequence *b* left shifted overlapping sections	**Sequence *b* right shifted overlapping sections**
TGCTAACTTTGATTGCCTA TGAATCCCTTGAATGAAC	TGCTAACTTTGATTGCCTA TGAATCCCTTGAATGAAC
TGCTAACTTTGATTGCCTA TGAATCCCTTGAATGAAC	TGCTAACTTTGATTGCCTA TGAATCCCTTGAATGAAC
…	…

Results and Discussion

An implementation

Fast local alignment generating methodology has been implemented in C++. While searching for the optimal solution, during the execution, memory keeps two data vectors, whose content is dynamically changed. Data vector - set of triples, identifying matching regions found within current overlapping sections and data vector - set of triples, identifying an un-gapped local alignment with a highest score, found until then. As has been previously explained, each triple is a unique identifier of a matching region, holding region's length and region's starting positions at the sequences. For each set of matching regions, found within current overlapping sections, fast local alignment generating function – FLAG is called, generating as an output an optimal un-gapped local alignment, regarding the longest local matching region. Afterwards, alignment's score is compared with the score of the optimal alignment found until then. If a higher score alignment is found, the optimal alignment and its score are updated.

```
**function** FLAG(**input:** { $R_1, R_2, ..., R_{\chi-1}, R_\chi$ }, **output:** $A$){
 if( $\chi != 0$ ){
  if( $\chi == 1$ ){
    $score \leftarrow \mu \times length(R_1)$
    $A \leftarrow R_1$
  }
  else
  {
   Find $R_\varsigma = \max len\{R_1, R_2, ..., R_{\chi-1}, R_\chi\}$
    $A \leftarrow R_\varsigma$
    $A_e \leftarrow R_\varsigma$
    $score \leftarrow \mu \times length(A)$
   if( $\varsigma == 1$ ){
    for( $\xi = 2; \xi <= \chi; \xi++$ )
    {
     $A_e \leftarrow A_e \rhd\lhd R_\xi$
     if($f(A_e)$ >score){
       $score \leftarrow f(A_e)$
       $A \leftarrow A_e$
     }
    }
   }
  else if( $\varsigma == \chi$ ){
   for( $\xi = \chi - 1; \xi >= 1; \xi--$ )
   {
    $A_e \leftarrow R_\xi \rhd\lhd A_e$
    if($f(A_e)$ >score){
     $score \leftarrow f(A_e)$
      $A \leftarrow A_e$
    }
   }
```

```
    }
    else
    {
     for( $\xi = \varsigma - 1; \xi >= 1; \xi --$ )
     {
       $A_e \leftarrow R_\xi \rhd\lhd A_e$
       if($f(A_e)$ >score){
          $score \leftarrow f(A_e)$
          $A \leftarrow A_e$
       }
     }
     $A_e \leftarrow A$
     for( $\xi = \varsigma + 1; \xi <= \chi; \xi ++$ )
     {
         $A_e \leftarrow A_e \rhd\lhd R_\xi$
        if($f(A_e)$ >score){
          $score \leftarrow f(A_e)$
          $A \leftarrow A_e$
        }
       }
     }
    }
   }
  }
```

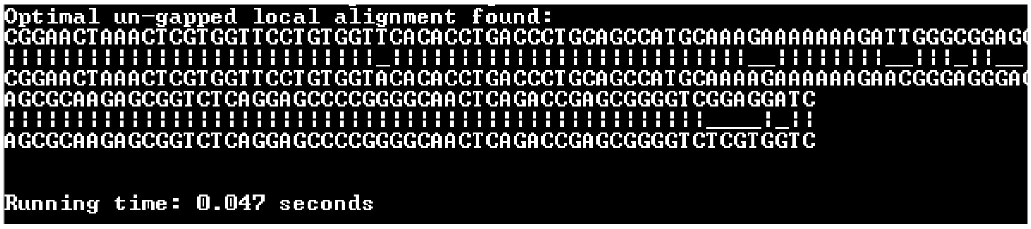
Test results

Implementation's running time has been measured, aligning ten pairs different length nucleotide sequences, on Fujitsu computer with Core(TM) 2 Duo CPU at 2.67GHz and 2 GB RAM. Score metrics, awarding +2 for each nucleotide match, while penalizing each mismatch with -1, has been used. Approximately same size similar sequences have been aligned. According results presented in Table 4, implementation's linear time complexity *O(m)* is more than evident, when aligning approximately same size similar sequences. Following ours previous space efficient implementation [6], two data vectors – set of triples, identifying matching regions found within overlapping sections and current optimal alignment are kept in the memory during the execution, resulting with linear space complexity – *O(m)*.

Table 4. Implementation's running time

sequence *a*	**sequence's** *a* **length -** l_a	**sequence** *b*	**sequence's** *b* **length –** l_b	**running time t (sec)**
Columnea latent viroid clone 1-6	374	Columnea latent viroid RNA	370	**0.047**
Cherry chlorotic rusty spot associated small satellite-like dsRNA B	609	Cherry chlorotic rusty spot associated small satellite-like dsRNA C	606	**0.125**
Ageratum leaf curl Cameroon betasatellite, isolate	1380	Ageratum leaf curl Cameroon betasatellite, isolate	1379	**0.343**

SatB6		SatB14		
Cyclovirus Chimp11	1750	Cyclovirus Chimp12	1747	**0.218**
Stachytarpheta leaf curl virus - [Hn6.1]	2751	Stachytarpheta leaf curl virus - [Hn5.4]	2748	**0.437**
Adeno-associated virus 3	4726	Adeno-associated virus 3B	4722	**1.029**
Mouse parvovirus 4a	4800	Mouse parvovirus 4b	4794	**0.702**
Banana streak IM virus	7769	Banana streak Imove virus strain IRFA910	7768	**1.685**
Gremmeniella abietina type B RNA virus XL1	10375	Gremmeniella abietina type B RNA virus XL2	10 374	**3.105**
O'nyong-nyong virus strain SG650	11822	Igbo Ora virus strain IBH10964	11821	**2.729**



```
Optimal un-gapped local alignment found:
CGGAACTAAACTCGTGGTTCCTGTGGTTCACACCTGACCCTGCAGCCATGCAAAGAAAAAAAGATTGGCGGAG
|||||||||||||||||||||||||||_||||||||||||||||||||||||||__||||||||__|||_||__
CGGAACTAAACTCGTGGTTCCTGTGGTACACACCTGACCCTGCAGCCATGCAAAAGAAAAAAGAACGGGAGGA
AGCGCAAGAGCGGTCTCAGGAGCCCCGGGGCAACTCAGACCGAGCGGGGTCGGAGGATC
|||||||||||||||||||||||||||||||||||||||||||||||||||____|_||
AGCGCAAGAGCGGTCTCAGGAGCCCCGGGGCAACTCAGACCGAGCGGGGTCTCGTGGTC


Running time: 0.047 seconds
```

Figure 1. Columnea latent viroid RNA and Columnea latent viroid clone 1-6 alignment.

Given a data set $S=\{<la_i, lb_i, t_i>, i=1\dots10\}$ obtained during the experimental evaluation, Principal Components Analysis (PCA) was used to fit a linear regression that minimizes the perpendicular distances from the data to the fitted line. This problem is equivalent to search for the linear sub-space which maximizes the variance of projected points, the latter being obtained by eigen decomposition of the covariance matrix. Eigen vectors corresponding to large eigen values are the directions in which the data has strong component, or equivalently large variance. PCA finds an orthogonal basis that best represents given data set. In our case the fitted line can be described with the following equation:

$$\vec{r} = \vec{n} + t * \vec{p}$$

Where n = [0.707038;0.707175;1.852933e-04] is the point on the fitted line and p = [4.635600e+03; 4.632900e+03; 1.042000] is the line direction vector, and $t \in R$.

The fitted line together with the orthogonal distances from each point to the line is shown on Fig. 2. It shows a linear dependency of aligning time for similar nucleotide sequences.

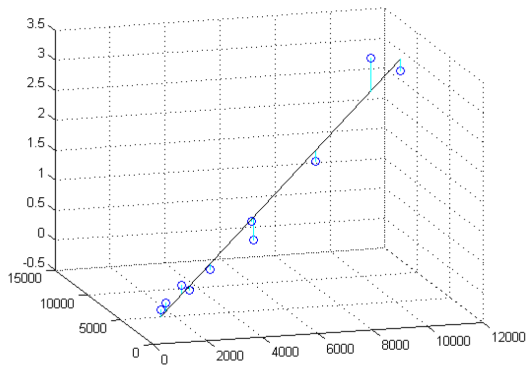


Figure 2. Fitted line with the orthogonal distances from each point to the line.

Conclusions

Linear time and space alignment technique has been presented, applicable on similar nucleotide sequences. The time complexity improvement is due to the methodology, according which un-gapped local alignments are generated within overlapping sections and the reduced alignments' search space. Also, the space complexity remains linear, based on region's space efficient representation.

References

1. T. SMITH, M. WATERMAN, Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**(1), 195, 197 (1981).
2. M. WATERMAN, M. EGGERT, A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *Journal of Molecular Biology*, **197**,723, 728 (1987).
3. X. HUANG, W. MILLER, A time-efficient, linear-space local similarity algorithm. *Advances in Applied Mathematics*, **12**,337, 357 (1991).
4. D. LIPMAN, W. PEARSON, Rapid and sensitive protein similarity searches. *Science*, **227(**4693**)**,1435, 1441 (1985).
5. S. ALTSCHUL, W. GISH, W. MILLER, E. MYERS, D. LIPMAN, Basic local alignment search tool. *Journal of Molecular Biology*, **215(**3**)**,403, 410 (1990).
6. D. STOJANOV, A. MILEVA, S. KOCESKI, A new, space-efficient local pairwise alignment methodology. *Advanced Studies in Biology*, **4(**2**)**,85, 93 (2012).