JOURNAL OF CLINICAL AND DIAGNOSTIC RESEARCH

**How to cite this article :**
GARG D. MULTI HIT, DROPOFF PERCENTAGE AND NCM-2: THREE IMPROVEMENTS IN BLAST. Journal of Clinical and Diagnostic Research [serial online] 2007 August[cited: 2007 August 1]; 4:340-347
Available from
http://www.jcdr.net/back_issues.asp?issn=0973-709x&year=2007&month=August &volume=1&issue=4&page=339-346 &id=32

ORIGINAL ARTICLE

# Multi Hit, Dropoff Percentage and NCM-2: Three Improvements in BLAST

GARG D


## ABSTRACT

Various algorithms are in use in medical processes to improve the speed, sensitivity and accuracy of the computations and analyses involved in those experiments.
The aim of this paper is to suggest three improvements, namely Multi Hit, Dropoff percentage and NCM-2 in the BLAST algorithm.
BLAST (Basic Local Alignment Search Tool) is a popular tool used for determining the patterns in genomic sequences. As the data is increasing exponentially, the need for advanced and complex algorithms for improving the accuracy, speed and sensitivity of pattern discovery tools in bioinformatics is also increasing.
**First Improvement:** The initialization of the word matches in a pairwise sequence alignment works either on single hit or two-hit algorithms. Instead, if we use a 3-hit or n-hit in general then the results improve in general and improve dramatically for some specific species and sequences.
**Second Improvement:** BLAST is using a drop-off score to calculate the highest scoring pairs between two sequences. A change has been proposed to calculate the threshold score that determines the inclusion of the subsequence in the result. Instead of using a drop-off score, if we use a drop-off percentage, it gives better results for some sequences.
**Third Improvement**: We propose an NCM-2 approach for normalizing BLAST values for simple regions. This approach is based upon the natural properties of the Amino acid sequences.
The algorithms have been run on Linux ES platform with Compaq Presario 2GB RAM and compared to the original BLAST.


## Introduction

Nowadays, medical practitioners are increasingly relying on sequence comparison tools to understand the various properties of the genomes under consideration. BLAST is being used for pairwise alignments instead of multisequence alignments. BLAST can perform local as well as global alignments.

In case of global alignments, it sacrifices some of the advantages of the local alignments. It is the most popular tool being used by the biologists and by people involved in the clinical trials of various drugs. Detection of the SARS disease source and then finding a remedy for it is a good example of success of BLAST-like tools.

These kinds of algorithms are also being used in pulse oximetry to remove the background noise. These tools have been used for solving various other medical problems. The examples and references are available at NCBI [http://ncbi.nlm.nih.gov]

The central idea of the BLAST algorithm is that a statistically significant alignment that will be of use to the doctors is likely to contain a high-scoring pair

Corresponding Author Dr Deepak Garg, A P.
Computer Science Department, Thapar University,
Patilala-147004, India
Email : deep108@yahoo.com,
Ph: +91-9815599654

of aligned words. BLAST first scans the database of genomes for words that score at least *T* when aligned with some word within the query genome sequence. Any aligned word pair satisfying this condition is called a `hit`. In the second part, BLAST checks whether each hit lies within an alignment with a score sufficient for the sequence to be part of the result set. It will be achieved by extending a hit in both directions until the running alignment's score has dropped more than *X* below the maximum score yet attained. This extension step is computationally quite expensive. The extension step typically accounts for two third time of BLAST execution time. It is therefore desirable to reduce the number of extensions performed [1],[2].

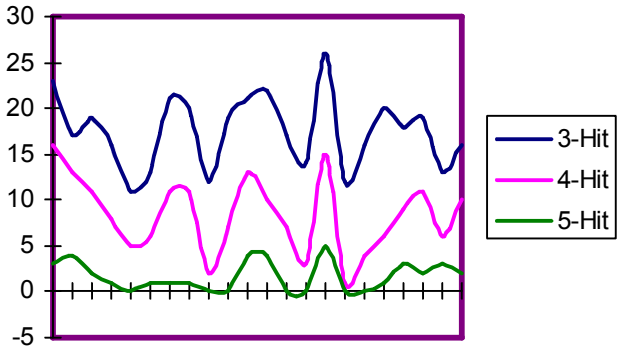
A two-hit algorithm was made to solve this problem. It is observed that an HSP (High Scoring Pairs) of interest is much longer than a single word pair and may therefore entail multiple hits on the same diagonal. The multiple hits should be in relatively short distance of one another. Specifically, a window length *A* is chosen, and it invokes an extension only when two non-overlapping hits are found within the distance *A* of one another on the same diagonal. Any hit that overlaps with the most recent one is ignored. We require two hits rather than one to invoke an extension. Therefore, the threshold parameter *T* must be lowered to retain comparable sensitivity. The effect is that many more single hits are found, but only a small fraction has an associated second hit on the same diagonal that triggers an extension. The great majority of hits may be dismissed after the minor calculation of looking up, for the appropriate diagonal, the co-ordinate of the most recent hit. After checking whether it is within distance *A* of the current hit's coordinate, the old is finally replaced with the new co-ordinate. Empirically, the computation saved by requiring fewer extensions more than offsets the extra computation required to process the larger number of hits.

## Multi hit

There are very rare chances that the occurrence of disease patterns will only be at one or two places in the genome sequence. Generally, it will be spread at various places in the genome sequence. When we look up the database with the query sequences to find some similarity between the sequence under scrutiny and the other sequences present in the database, the result will have the details of any functional or relational match between the two [3],[4].

The proposed new approach extends a two-hit algorithm to an N-hit algorithm. Here, the value of N can be given by the user, depending upon the requirements. The value of N will then act as a tradeoff between speed and sensitivity. According to the N-hit algorithm, we can choose a window length *A*. The algorithm step invokes an extension only when N non-overlapping hits are found on the same diagonal and the difference between each of them is *A*. Any hit that overlaps with the most recent one is ignored.

## Table/Fig 1




**Comparison of protein database sequences using 3-hit, 4-hit and 5-hit algorithm**

Efficient execution requires an array to record for each diagonal, the first co-ordinate of the most recent (N-1) hits found. Database sequences are scanned sequentially. Therefore, this co-ordinate always increases for successive hits. Rather than one or two hits, we require N hits to invoke an extension. Therefore, the threshold parameter *T* must be lowered depending upon the value of N to retain comparable sensitivity. Compared to the one-hit or two-hit methods, a small fraction of hits will have an associated (N-2) hits on the same diagonal that triggers an extension. The great majority of hits may be dismissed after the minor calculation of looking up, for the appropriate diagonal. Then the coordinate of the most recent hit is checked that whether it is within distance *A* of the current hit's

coordinate, and finally replacing the old with the new coordinate. The N-hit algorithm's overall performance speed was found to be faster than that of the two-hit algorithm. The reason is that the N-hit algorithm saves a lot of computation time in extension because it has to extend very few word hits lying on the same diagonal within a particular distance. The number of word hits produced by the three-hit algorithm is more than the number of hits produced by the two-hit algorithm. This increase in the hits increases the computation time taken to process it. The number of hits can be controlled by changing the value of T. However, the fewer extensions to be made to offset this extra computation time are required to process the larger number of hits. A BLAST comparison of Broad Bean Leg-hemoglobin - I (87) (SWISS-PROT accession no. P02232) and Horse b-globin (88) (SWISS-PROT accession no. P02062) is done. A one-hit algorithm produces 15 hits with score at least 13. A two-hit algorithm produces an additional 22 non-overlapping hits with score at least 11. Finally, a three-hit algorithm produces 18 additional non-overlapping hits.

The probability of missing an HSP (High Scoring Pairs) is further reduced by a three-hit algorithm. This is because the value of the threshold T is further lowered in this case as compared to a two-hit algorithm. The disadvantage of this approach is that the number of hits produced could be very large for a large query and database. In that case, the main memory can run out of space. Another disadvantage is that we can miss weak similarities either if they fail to produce N word hits or if the threshold value is not set to a low value. Moreover, if the sequences being compared are not very similar, the N-hit algorithm is theoretically at a disadvantage at finding short regions of similarity, such as individual protein domains and short coding exons in vertebrate genome sequences, particularly if T is not lowered to compensate for the use of the N-hit algorithm. As we increase the value of N, the suitability of the N value continues to decrease. The sensitivity of the result also decreases to a greater extent after the value of N reaches 5. Typically, the value of N will be 2, 3, 4 or 5. The sensitivity of the result is also found to decrease with even N=3 for some sequences. Therefore, depending upon the type of experiment and the pattern matching that will suit the biologist; N can be taken as 1 or 2. Depending on the requirements of the biologist, he can set the value of N. This gives him the advantage of having a less number of non-genuine sequences.

## Dropoff Percentage

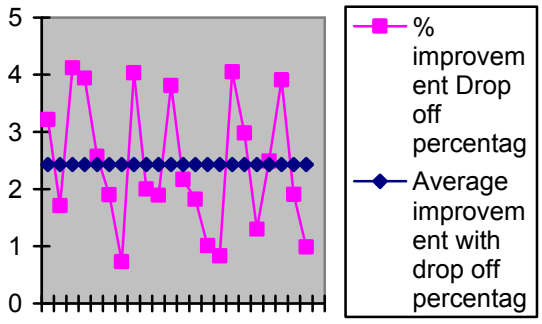
The problem of not getting the exact answers to the queries is haunting medical practitioners. A Google search yields many results. Similarly, we get too many results in a BLAST comparison. Researchers are performing various experiments to solve this problem. We propose a drop off percentage in place of a drop off score. A drop off score is the value of score and tells how much the score is allowed to drop off since the last maximum. If the X value is set to a high, the quality of the alignment is degraded. On the other hand, if a smaller value is set for X, there are chances of missing some alignments.

The drawback in this approach is that the value of X depends on the substitution scores, the gap initiation and the extension costs. Therefore, the easier way to calculate the drop off will be to define a drop off percentage. Drop off percentage will be the number of mismatches allowed after a significant number of matches. In this case, there will be no need to refer to the substitution matrix. Therefore, there will be no increase in the speed. To make the concept clear, we will try to align two sentences. To make the example simpler, we will ignore the spaces and refrain from allowing gaps in the alignment. Take for example, the following two sentences:
ACTGTAGCTACAGCTATACGTAGCAGAC
ACTGTATATACAGTGCGAGCTCTC TCAC
The two sentences first have six matches and then two mismatches before the next match. For this, the drop off percentage comes out to be (2 / 6) *100 = 33.33 %.

The extension can be either carried or terminated according to the parameter drop-off percentage that can be set depending upon the requirements of the biologist regarding the sensitivity of the sequences. The higher the drop off percentage allowed, the more the dissimilar sequences that can arise in the result. The lower the drop off percentage, the higher the possibility of near exact matches arising in the result.

Next, we have five matches before a mismatch:
TACAG C
TACAG T
After that, we have four mismatches continuously before a match:
CTAT A
TGCG A
When we are at the second mismatch, the drop off percentage is (2/5)*100=40. After the third mismatch, the drop off percentage is 60. After the fourth mismatch, the drop off percentage comes out to be (4 / 5) * 100 = 80. Depending upon our choice of value for a drop off percentage, the extension is terminated at a point and trimmed back to the previous state, that is, the last match. The results that come out with the drop off percentage are the same as those with a drop off score. The drop off method also terminates the extension at N but the approach behind the termination is different. To explain the approach, we will try to align the same two sentences by using a scoring scheme in which identical letters score +1 and mismatches score –1. To keep the example simple, we will ignore the spaces and refrain from allowing gaps in the alignment. Although only the extension to the right is shown, an extension also occurs to the left of the seed word. Here, a variable X that represents a drop off score must be selected.

## Table/Fig 2




## Comparison of protein database sequences using a drop off percentage

It represents how much the score is allowed to drop off since the last maximum. Let us set X to 5 and see what happens. Here, we have to keep a track of the sum score and the drop off score.
ARN DCQCK GHILK MFP YTMP
ARN DCQET GHILK STW VTRR
123 45 6 5 4 5 6 7 8 9 8 7 6 5 6 5 4 << score
000 00 0 1 2 1 0 0 0 0 1 2 3 4 3 4 5 << number of positions for drop off

The maximum score for this alignment is 9, and the extension is terminated when the score drops to 4. After terminating, the alignment is trimmed back to the maximum score. The maximum score was at N. Therefore, the alignment ends at N. The alignment produced here is the same as that produced by our proposed approach. It also aligns the two sentences to N. However, it involves the sum score as well. The sum score has to be regularly referred to the scoring matrix. This slows down the speed of the search.

The value of the drop off parameter depends on the substitution scores, the gap initiation and the extension costs. It regularly needs to refer to the scoring matrix. Therefore, it slows down the speed of the search. The easier way to calculate the drop off is suggested. Drop off percentage is the number of mismatches allowed after a significant number of matches. In this case, there is no need to refer to the substitution matrix. Therefore, there is an increase in the speed. The results that are thrown up with the drop off percentage are marginally better than those thrown up by the drop off score. However, the approach behind the drop off percentage is different and gives flexibility to the biologist.

## Drop off percentage with multi-hit strategy

The number of sequences to be extended decreases in the case of multiple hits. Therefore, the drop off percentage calculation has to be done less number of times. Applying both the approaches in an integrated algorithm and then testing them on the same protein sequences yielded interesting results as tabulated in table 1. The algorithm is given in [Table/fig 3] The overall average improvement was 15%. The reason lies in the addition of the calculation used for finding multiple hits and to calculate the drop off percentage. In the case of these approaches being implemented individually,

only the corresponding part of the algorithm is run. This results in less computation.

**Table/Fig 3**

Step 1. Start
Step 2. Choose an appropriate value for window length A.
Step 3. Lower the value of the threshold T in order to yield more hits.
Step 4. Put the value of N.
Step 5. Seek N non-overlapping hits found with distance A of one another on the same diagonal
Step 6. Invoke (ungapped) an extension to determine if hits lie within a statistically significant alignment with query.
Step 7. Go for a dropoff percentage instead of a dropoff score
Step 8. Choose an appropriate value for the drop off percentage
Step 9. Input the drop off percentage
Step 10. Calculate the value of X%= (number of matches)*(drop off percentage/100)
Step 11. Extend until the sum of number of mismatches or number of gaps combined is less than or equal to X%
Step 12. End

## Algorithm for multi-hit with drop off percentage

The sequence homology searches are designed to detect high scoring matches. Sometimes, it has been found that many high scoring results are not useful to the biologist who was looking for either a functional or structural similarity in those sequences. The reason is the presence of sequence areas that are of less complexity but result in high scores either due to repetition or due to compositionally biased regions. These types of sequences that can provide you with seemingly good scores but are of no use should not be part of the result set. [1].

There are many good theories that have been developed. However, there is still scope for improvement. The assumption that the making of the sequences to be compared is similar to the overall making of amino acids in the collection database [2],[3],[4],[5] does not hold either for simple sequences or for sequences of less complexity.

**Table/Fig 4**

| Protein | Species | Accession No | % improvement |
|---|---|---|---|
| 53BP1 | H.sapiens | 488592 | 13 |
| BARD | Homosapiens | 1710175 | 10 |
| C19G10.07 | Schizosaccharomyces pombe | 1723501 | 20 |
| CDC9 | Candida albicans | 1706483 | 13 |
| Crb2 | S.pombe | 1449177 | 19 |
| DNA ligase | Thermus scotoductus | 1352293 | 15 |
| DPB11 | S.cerevisiae | 1352999 | 21 |
| ECT2 | Mus musculus | 423597 | 11 |
| F26D2.b | Caenorhabditis elegans | 1914176 | 14 |
| F37D6.1 | C.elegans | 1418521 | 22 |
| KIAA0170 | H.Sapiens | 1136400 | 12 |
| KIAA0259 | H.sapiens | 1665785 | 17 |
| PPOL | Sacrophaga peregrine | 1709741 | 20 |
| RAP1 | S.cerevisiae | 173558 | 7 |
| RAP1 homolog | K.lactis | 422087 | 14 |
| REV1 | Saccharomyces cerevisiae | 1324209 | 12 |
| T10M13.12 | Arabidopsis thaliana | 2104545 | 16 |
| T13F2.3 | C.elegans | 1667334 | 21 |
| T19E10 | C.elegans | 1067065 | 14 |
| TDT | Mus domestica | 2149634 | 17 |
| UNE452 | S.cerevisiae | 1151000 | 17 |
| XRCC1 | M.musculus | 627867 | 16 |

## Comparison of protein database sequences using 3-hit combined with drop off percentage NCM-2 (Neutral, Conservative, Match & Mismatch)

The distribution of local similarity scores follows the extreme value distribution for gapped as well as ungapped alignments. The initial algorithms were

two phase algorithms: In the first phase, we find the regions of lesser complexity. In the second phase, we apply the filtering or masking criteria before aligning the sequences [6],[7]. One uses the threshold value and the other uses the entropy to detect areas of lesser complexity. These algorithms don't find out all the areas of lesser complexity but are dependent on the choice of the value for various parameters. Due to the filtering or the masking process, there may be a chance of losing some biologically important information for the scientists. [8]

These disadvantages were taken care of to some extent in [9] where only one type of residue was masked. Another study adjusts the parameters depending upon the sequences that are being compared. The composition of the sequences plays an important role. However, when the database against which the query is being matched has the query sequence itself; it has been found that the query sequence is not shown as the first match. On the contrary, it is shown below many other results and sometimes not shown at all. The reason lies in the fact that adjustment is done using different parameters for different pairs. SEG, CAST and XNU are the previously available techniques for this purpose.

SEG uses the parameters of complexity state vector, sliding window and trigger for low complexity. First, it identifies approximate segments of low complexity using sliding window. Next, it optimizes these segments. If the measurement of the underlying area is lesser than a given threshold, it is recognized as an area of lesser complexity.

CAST uses the dynamic programming approach. The regions scoring above the cut-off or threshold in the local similarity search with sequences composed of single amino acid type are defined as regions of lesser complexity. XNU uses intrinsic repeats (biased regions of distinct amino acid without clear repeating patterns) and internal repeats (tandem configurations of discrete units). These repeats are identified by using a dot-plot matrix of the query sequence by scoring the local similarity with a PAM matrix and estimating the statistical significance of the score. [10]

CARD uses the regions that are delimited by identical pairs of sub-sequences depending upon the overlapping positions of two sub-sequences to identify the lesser complex regions. If the position of the sequence is found to be either tandem or overlapped, the region containing the two identical sequences is marked as the lesser complexity region. Some algorithms use Jenson-Shannon divergence and Kullback-Leibler divergence between the background distributions of all the amino acids. [11] Another method called composition-based statistics changes the statistical parameters based on the compositions of the sequences compared. This method worked well in some cases by eliminating the similarities because of unusual sequence compositions. NCM-2 (Neutral, Conservative, Match and Mismatch) is defined as that when two sequences of amino acids are matched based on some scoring matrix, for example, BLOSUM62 and PAM. Depending on the score of the comparison between two amino acids, the comparison can be called neutral, conservative, match or mismatch. This also takes care of the significant similarities that might come along, apart from the chance similarities. It only uses the first order statistics and ignores the order of the amino acids to be able to eliminate these significant matches. Significant matches of low-complexity segments are made of long, continuous sub-alignments of identical matches. If the identification of the distribution of segments of identical and similar matches can be done, then the significance of alignment can be estimated. The focus here is on the length of the segments under consideration. Similar segment means that it has only matches or conservative matches. The significant similarities have the property either that these cannot be extended to left or right or that they are not significant similarities.

## Methodology Used

A test set was constructed by selecting the 15 families from the Pfam database that are high in ranking. The Pfam has about 5000 protein families. Forty protein sequences from these families were chosen randomly. This made up a total of 600 sequences. This test set contains fragments as well as full protein sequences. Some proteins have multiple domains and some do not have any domain at all.

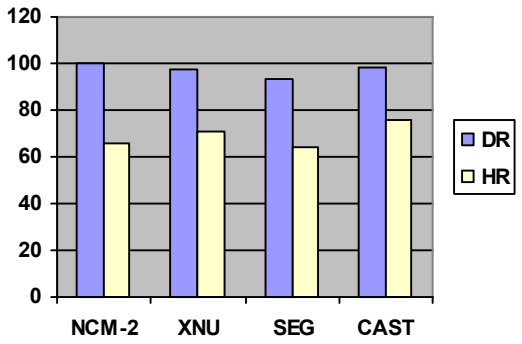
This data set has both protein fragments and full protein sequences. The SEG was run with the parameter values W = 12, K(1) = 2.2 bit and K(2) = 2.5 bit. The threshold score was 40 for CAST. The cutoff = 0.01, max-search offset =4 and min-search-offset=1 was used for XNU.

**Table/Fig 5**

| | NCM-2 | XNU DR/HR | SEG DR/HR | CAST DR/HR |
|---|---|---|---|---|
| HCV NS1 | 100/16 | 98/18 | 99/13 | 100/0 |
| Cytochrome b C | 100/52 | 100/44 | 56/8 | 100/79 |
| PPR | 100/90 | 99/94 | 100/89 | 99/89 |
| Rvp | 100/55 | 98/83 | 97/83 | 99/94 |
| EGF | 100/92 | 98/98 | 90/96 | 99/97 |
| Ank | 100/93 | 98/97 | 96/96 | 99/99 |
| COX | 100/35 | 100/28 | 100/10 | 100/41 |
| Efhand | 100/86 | 96/94 | 93/97 | 94/92 |
| AB Trans | 100/55 | 100/62 | 98/44 | 100/69 |
| Pkinase | 100/78 | 95/86 | 97/84 | 97/95 |
| LRR | 100/86 | 87/99 | 82/99 | 92/99 |
| RuBisCo Large N | 100/13 | 98/1 | 100/0 | 100/42 |
| WD40 | 100/95 | 96/99 | 97/91 | 99/98 |
| Iq | 100/84 | 97/96 | 100/92 | 98/87 |
| Oxidored q1 | 100/62 | 98/60 | 98/61 | 100/55 |
| | 100/66.1 | 97.2/70.6 | 93.4/64.2 | 98.4/75.7 |

## Comparison of NCM-2 with other techniques

**Table/Fig 6:**




**Comparison of performance data for NCM-2 with three other techniques**

XNU, SEG and CAST use masking operations. Here, DR represents the detection ratio and is calculated as

DR = (Number of domains in masked sequence/ Number of domains in unmasked given sequence) * 100

HR represents Hit ratio and is calculated as

HR = (number of masked residues outside the domains of the sequence/number of masked residues in the filtered sequence) * 100

## Results

The objective of filtering is to mask non-domain regions without masking the domain regions. The detection ratio may decrease by masking the domain regions. We get a maximum detection ratio. NCM-2 does not have any masking operation and the Pfam database entries themselves have their domains identified with the algorithm inbuilt into them. Therefore, NCM-2 detects 100% of the simple regions. Any algorithm without masking will identify 100% of the sequences. The objective of masking was to filter out high scoring database subsequences coming in the result of the alignment showing higher similarity. The same task is now being performed by the semantic introduced above. It is based on the composition of the sequence and also depends on the number of matches, mismatches, neutral matches and conservative matches. Time is saved because there is no need to perform the masking operation. The same saving in time can be utilized to calculate the probabilities related to C,N,M+,M- of NCM-2. As shown in Table/Fig 4], NCM-2 will detect 100 percent of the sequences due to non masking as compared to XNU(97.2 ), SEG (93.4 ), CAST(98.4). This method also takes care of the non-genuine sequences and eliminates them from the list of significant matches.

## Conclusion

By introducing the new approach of multi-hit with drop off percentage, there was a significant improvement in the running time of BLAST. The biologist also has the flexibility to get the result of his own choice by changing either the value of the number of hits or the drop off percentage or both. The introduction of multiple hits will have an effect on the sensitivity if the number of hits selected is more. In the case of three hits, the algorithm shows an improvement ranging from 11 to 26 percent on

different sequences. The introduction of the drop off percentage results in less number of calculations because there is no need to go to the scoring matrix and the resulting improvement ranges from 1 to 5 percent. The combined algorithm implementation of multi-hit with drop off percentage shows an improvement in the range of 7 to 21 percent. Therefore, using this method, the biologist can derive the dual benefits of flexibility and speed without compromising on any other aspect. Due to NCM-2 no information will be lost and genuine sequences will travel to the result of the homology searches.

**Conflict of Interest:** None declared

## References

[1] Golding GB. Simple sequence is abundant in eukaryotic proteins. Protein Science 1999;8:1358-1361.

[2] Karlin S, Altschul SF. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proceedings of the National academy of sciences USA 1990;87:2264-2268.

[3] Karlin S, Altschul SF. Applications and statistics for multiple high scoring segments in molecular sequences. Proceedings of the National academy of sciences USA 1993;90:5873-5877.

[4] Dembo A, Karlin S. Strong limit theorems of empirical functional for large exceedances of partial sums off i.i.d variables. Ann Prob 1991;19:1737-1755.

[5] Dembo A, Karlin S, Zeitouni O. a and b Critical Phenomena for sequence matching with scoring and Limit distribution of maximal non-aligned two sequence segmental score. Ann Prob1994;22:1993-2021, 2022-2039.

[6] Claverie JM, States DJ. Information enhancement methods for large scale sequence analysis.Computers and Chemistry 1993;17,191-201.

[7] Wooton JC. Sequences with unusual amino acid compositions Current Opinion in Structural Biology 1994;4:413-421.

[8] Yona G, Levitt M. 2000. A unified sequence structure classification of proteins: combining sequence and structure in a map of protein space. Proc Recomb 2000; 308-317.

[9] Promponas et. Al. CAST: an iterative algorithm for the complexity analysis of sequence tracts. Bioinformatics 16; 915-922.

[10]. Claverie JM, States DJ. Information enhancement methods for large scale sequence analysis. Comput Chem 1993;17:191-201.

[11]. Gusfield D. Algorithms on strings, trees and sequence. Algorithms on Strings, trees and sequences, Cambridge University Press 1990. NY. PP. 89-107.