Bioinformation by Biomedical Informatics Publishing Group *open access*

www.bioinformation.net **Prediction Model**

Algorithm to find distant repeats in a single protein sequence

Nirjhar Banerjee[1], Rangarajan Sarani[1], Chellamuthu Vasuki Ranjani[1], Govindaraj Sowmiya[1], Daliah Michael[1], Narayanasamy Balakrishnan[2], Kanagaraj Sekar[1, 2, *]

[1]Bioinformatics Centre, Centre of Excellence in Structural Biology and Bio-computing; [2]Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560 012, India; Kanagaraj Sekar* - E-mail: sekar@physics.iisc.ernet.in; Phone: 91-80-22933059; Fax: 91-80-23600683; * Corresponding author

received July 07, 2008; accepted July 24, 2008; published September 19, 2008

Abstract:
Distant repeats in protein sequence play an important role in various aspects of protein analysis. A keen analysis of the distant repeats would enable to establish a firm relation of the repeats with respect to their function and three-dimensional structure during the evolutionary process. Further, it enlightens the diversity of duplication during the evolution. To this end, an algorithm has been developed to find all distant repeats in a protein sequence. The scores from Point Accepted Mutation (PAM) matrix has been deployed for the identification of amino acid substitutions while detecting the distant repeats. Due to the biological importance of distant repeats, the proposed algorithm will be of importance to structural biologists, molecular biologists, biochemists and researchers involved in phylogenetic and evolutionary studies.

Keywords: distant repeats; genome sequences; point accepted mutation; structure-function relationship; phylogeny

Background:
Distant repeats are evolved by duplication and recombination of genes, which gives rise to amino acid repeats within the protein sequence **[1]**. Andrade and co-workers examined the important differences between certain protein families in order to study their evolution, structure, and function **[2]**. To understand more about the structural and functional relationship of the repeats in protein sequences, many types of repeats such as Ankyrin repeats, Armadillo repeats, HEAT repeats, TPR repeats, HAT repeats, Kelch repeats, Leucine-rich repeats etc., have been studied in detail **[3]**. To get better understanding of these repeats, several algorithms have been derived to find amino acid repeats in protein sequences **[4-9]**. One of the pioneers in automatic recognition of repeats using computer methods three decades ago was McLachlan **[10]**. McLachlan and Stewart used Fourier transform analysis techniques (autocorrelation techniques) **[4, 11]** which has been followed by the development of different techniques and algorithms by other groups to find the distant repeats amino acid residues.

These algorithms detect patterns of amino acid distant repeats in protein sequences. Heringa and Argos introduced a method for the determination of distant repeats in protein sequences. This method looks for internal similarities by comparing the protein sequence to itself with standard sequence–sequence alignment techniques **[5]**. Morgenstern and coworkers introduced an algorithm where a multiple sequence alignment based on segment-to-segment comparison is made to find the local similarities of amino acids in protein sequence **[6, 12, 13]**. Further, Andrade and co-workers have derived a homology-based algorithm for the identification of protein repeats using statistical significance **[3]**. There are other servers such as REPPER **[7]**, REPRO **[8]**, RADAR **[9]**, etc., which uses different algorithms for finding repeats in protein sequences. However, most of the above mentioned algorithms have restrictions in the number of residues provided in the query sequence and does not give an easily interpretable result. To overcome this, an algorithm has been derived from the recent algorithm FAIR **[14]** for finding the distant repeats in a protein sequence.

The proposed algorithm utilizes the PAM (Point Accepted Mutation) matrix to calculate distant repeats. According to Dayhoff and coworkers, the residue pairs, with scores above one, replace each other more often as alternatives in related sequences than in random sequences during evolution. This is an indication that both the residues may carry out similar functions. A score exactly equal to one indicates amino acid pairs that are found as alternatives at exactly the frequency predicted by chance. Residue pairs with scores less than one replace each other less often in random sequences and would be an evidence for these residues to be functionally disparate. PAM250 matrix is chosen, by default, because at this evolutionary distance (250 substitutions per hundred residues) only one amino acid in five remains unchanged **[15]**. Thus, the proposed algorithm prevails over the constraints that have been limiting the previous algorithms.

Methodology:
The algorithm finds all possible distant amino acid sequence repeats in a given protein sequence. Two amino acid strings are considered repeats, if their corresponding residues are either identical or have a positive PAM matrix score (greater than or equal to 1). Therefore the strings 'KLN' and 'QLD' are distant repeats based on PAM250 matrix (K with Q has a score of 1 and N with D has a score of 2 based on PAM250 matrix). All possible available PAM matrices are incorporated and are downloaded from the EMBL website (http://eta.embl-heidelberg.de:8000/misc/mat/). The user has the option to choose a specific PAM matrix to find the distant

ISSN 0973-2063 (online) 0973-2063 (print)
Bioinformation 3(1): 28-32 (2008)

repeats in a particular protein sequence. The algorithm proposed here is derived from the recent algorithm, FAIR **[14]** and the details are described here:

Finding matches based on PAM matrix

Initially the protein sequence is stored in a string a1. The algorithm follows the same approach in finding repeats as in FAIR except that instead of finding an exact match, it looks for matches based on the PAM matrix scores. The algorithm takes the choice of matrix from the user. Then for each set of element (a1[i],a2[j]), it checks the corresponding PAM matrix whether the score is greater than one. 'pamvalue' is a Boolean character that shows true for a match and false when no such match exists. Thus, if the user gives PAM250 matrix as the choice the corresponding code will be as shown in illustration 1 in supplementary material.

Storing subsequences and repeat positions

After completion of the first part of the algorithm, the 'end-points' as well as the length of the repeats have been stored. The next part can be explained with the help of the figure (Figure 1) where the same strings 'KLN' and 'QLD' have been taken as an example. As shown in Figure 1, the array 'startd' contains the positions of the starting point of the 'first sequence' and the 'second sequence'. Similarly, the array 'endd' contains the positions of the end points of the two sequences. The manner in which the algorithm stores the repeat sequence and the starting points and end points in the vector 'vsubseq' is identical to that of FAIR **[14]**. Then the algorithm sorts the repeats to remove the identical entries so as to produce non-redundant output of distant repeats.

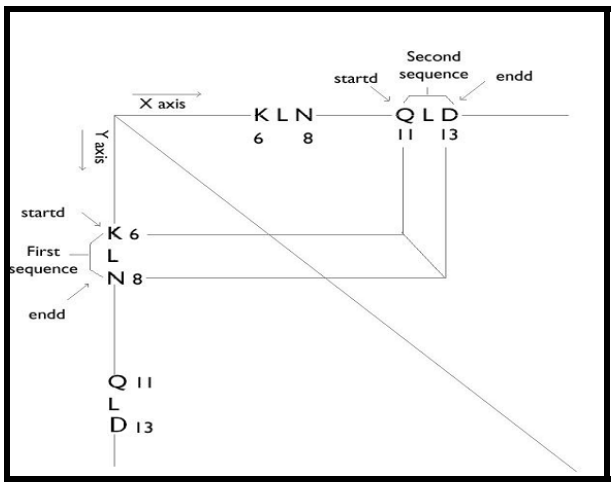

Figure 1: Alignment of subsequences (KLN and QLD) to detect the distant repeats in a protein sequence.

Discussion:

Case study 1

The sample output shown below is for the input protein sequence taken from *Homo sapiens*. The number of amino acid residues present in the input sequence is 2413. The minimum number of amino acids required to be present in a given distant repeat is 100 and the calculation is performed using the scores of PAM250 matrix. As can be seen, when the minimum number was set to 100, there are seven significant distant repeats with a minimum of 108 and maximum of 257 amino acid residues. Whereas, when the minimum number of amino acid residues in a repeat was set to 50, a significant set of repeats was identified. The algorithm produces four distant repeats of length 64 residues and implies that these domains of repeat in *Homo sapiens* (Figure 2) should have originated due to duplication events and may be involved in any eminent biological function.

Case study 2

The sample output shown in Figure 3 is for the input protein sequence taken from *Streptococcus pneumoniae* TIGR4. The number of amino acid residues present in the input sequence is 857. The minimum number of amino acids required to be present in a given distant repeat is 100 and the calculation is performed using the scores of PAM250 matrix. The case study in Figure 3 using the protein sequence from *Streptococcus pneumoniae* gives a total number of two distant repeats, when the minimum number of amino acid repeats was set to 100; two significant distant repeats with a minimum of 156 and a maximum of 308 amino acid residues were found. As can be seen, the distant repeat containing 156 amino acid residues is a sub-set of the other distant repeat containing 308 amino acid residues.

Conclusion:

An algorithm has been proposed to identify all the distant repeats present in a given protein sequence. PAM matrix scores are deployed for the identification of the distant repeats. Identification of such repeats in a protein sequence would aid the researchers to study the correlation of distant repeats with respect to their structure and function in the evolutionary process. Thus, distant repeats can be exploited to study the individual protein by their evolutionary conserved repeats and for modeling the three-dimensional structure of unknown proteins by their similar folding topology.

Acknowledgment:

The facilities of the Bioinformatics Centre (DIC), the Interactive Graphics Based Molecular Modeling facility (IGBMM) and the Supercomputer Education and Research Centre (SERC) are gratefully acknowledged. The corresponding author (KS) thanks the Department of Information Technology for financial support.

References:

[01] M. M. Edward *et al*., *J. Mol. Biol*., 293: 151 (1998) [PMID: 10512723]

[02] M. A. Andrade *et al*., *J. Struct. Biol.*, 134: 117 (2001) [PMID: 11551174]

[03] M. A. Andrade *et al*., *J. Mol. Biol.*, 298: 521 (1999) [PMID: 10772867]

[04] A. D. McLachlan, *Biopolymers*, 16: 1271 (1977) [PMID: 880354]

[05] J. Heringa and P. Argos, *Proteins*, 17: 391 (1993) [PMID: 8108381]

[06] B. Morgenstern *et al*., *Bioinformatics*, 14: 290 (1998) [PMID: 9614273]

[07] M. Gruber *et al*., *Nucleic Acids Res*., 33: W239 (2005) [PMID: 15980460]

[08] R. A. George and J. Heringa, *Trends in Biochem. Sci*., 25: 515 (2000) [PMID: 11203383]

ISSN 0973-2063 (online) 0973-2063 (print)
Bioinformation 3(1): 28-32 (2008)

[09] A. Heger and L. Holm, *Proteins*, 41: 224 (2000) [PMID: 10966575]
[10] A. D. McLachlan, *J. Mol. Biol.*, 61: 409 (1971) [PMID: 5167087]
[11] A. D. McLachlan and M. Stewart, *J. Mol. Biol.*, 103: 271 (1976) [PMID: 950663]
[12] B. Morgenstern *et al*., *Proc. Natl. Acad. Sci. USA*, 93: 12098 (1996) [PMID: 8901539]
[13] B. Morgenstern and W. Atchley, *Mol. Biol. Evol*., 16: 1654 (1999) [PMID: 10605108]
[14] N. Banerjee *et al*., *Curr. Science*, 95: 188 (2008)
[15] M. O. Dayhoff *et al*., *Atlas of Protein Sequence and Structure*, 5: 345 (1978)

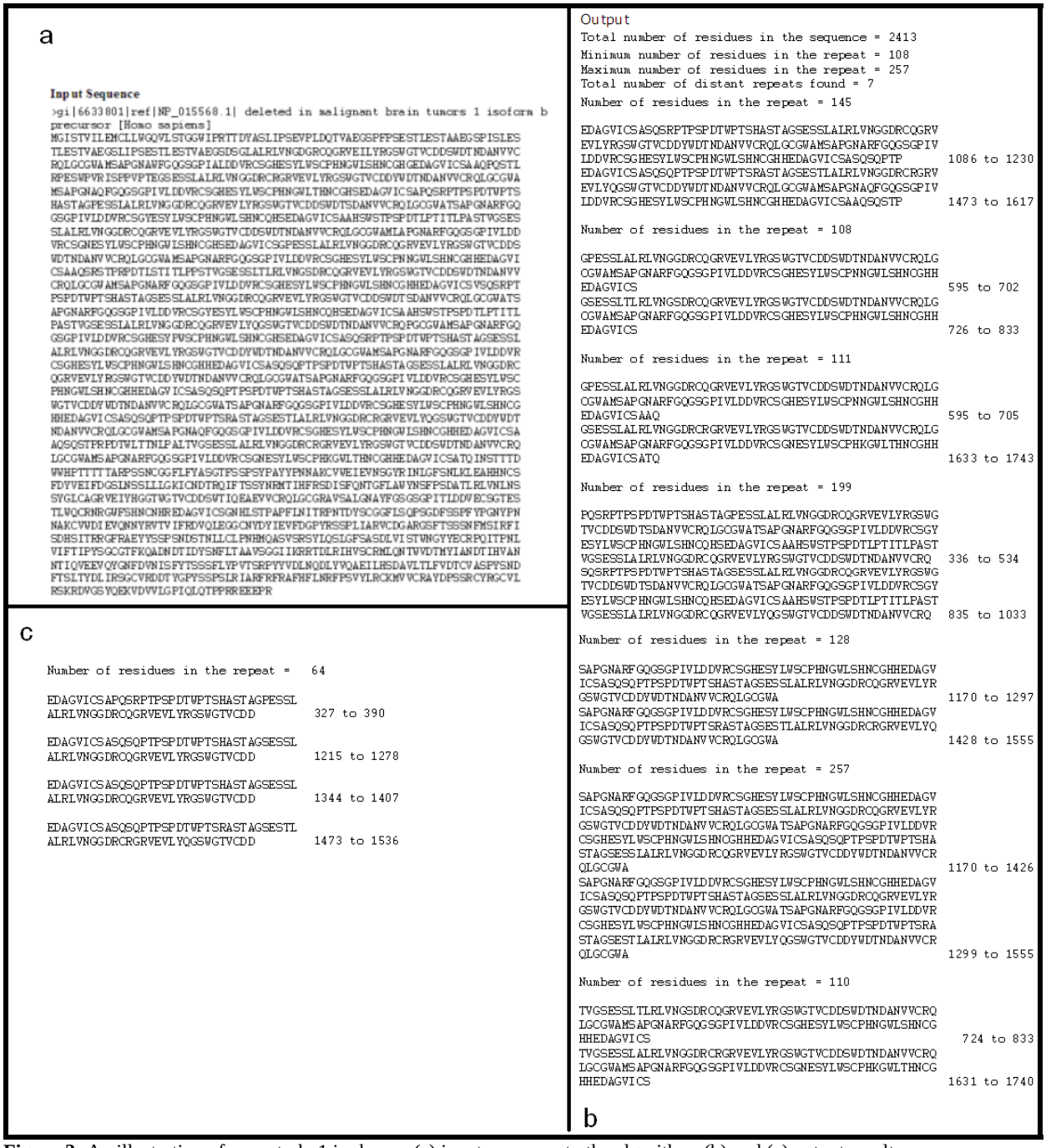

Figure 2: An illustration of case study 1 is shown. (a) input sequence to the algorithm; (b) and (c) output results.

ISSN 0973-2063 (online) 0973-2063 (print)
Bioinformation 3(1): 28-32 (2008)

```
Input Sequence

>gi|14971548|gb|AAK74270.1| cell wall surface anchor family protein
[Streptococcus pneumoniae TIGR4]
MKFNPNQRYTRWSIRRLSVGVASVVVASGFFVLVGQPSSVRADGLNPTPGQVLPEETSGTKEGDLSEKPG
DTVLTQAKPEGVTGNTNSLPTPTERTEVSEETSPSSLDTLFEKDEEAQKNPELTDVLKETVDTADVDGTQ
ASPAETTPEQVKGGVKENTKDSIDVPAAYLEKAEGKGPFTAGVNQVIPYELFAGDGMLTRLLLKASDNAP
WSDNGTAKNPALPPLEGLTKGKYFYEVDLNGNTVGKQGQALIDQLRANGTQTYKATVKVYGNKDGKADLT
NLVATKNVDININGLVAKETVQKAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIPYELFAGDGMLTR
LLLKASDKAPWSDNGDAKNPALSPLGENVKTKGQYFYQVALDGNVAGKEKQALIDQFRANGTQTYSATVN
VYGNKDGKPDLDNIVATKKVTININGLISKETVQKAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIP
YELFAGDGMLTRLLLKASDKAPWSDNGDAKNPALSPLGENVKTKGQYFYQLALDGNVAGKEKQALIDQFR
ANGTQTYSATVNVYGNKDGKPDLDNIVATKKVTININGLISKETVQKAVADNVKDSIDVPAAYLEKAKGE
GPFTAGVNHVIPYELFAGDGMLTRLLLKASDKAPWSDNGDAKNPALSPLGENVKTKGQYFYQVALDGNVA
GKEKQALIDQFRANGTQTYSATVNVYGNKDGKPDLDNIVATKKVTIKINVKETSDTANGSLSPSNSGSGV
TPMNHNHATGTTDSMPADTMTSSTNTMAGENMAASANKMSDTMMSEDKAMLPNTGETQTSMASIGFLGLA
LAGLLGGLGLKNKKEEN

Output

Total number of residues in the sequence = 857

Minimum number of residues in the repeat = 156
Maximum number of residues in the repeat = 308
Total number of distant repeats found = 2

Number of residues in the repeat = 156

ININGLVAKETVQKAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIPYELFA
GDGMLTRLLLKASDKAPWSDNGDAKNPALSPLGENVKTKGQYFYQVALDGNVAG
KEKQALIDQFRANGTQTYSATVNVYGNKDGKPDLDNIVATKKVTININ            290 to 445
ININGLISKETVQKAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIPYELFA
GDGMLTRLLLKASDKAPWSDNGDAKNPALSPLGENVKTKGQYFYQVALDGNVAG
KEKQALIDQFRANGTQTYSATVNVYGNKDGKPDLDNIVATKKVTIKIN            594 to 749

Number of residues in the repeat = 308

ININGLVAKETVQKAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIPYELFAG
DGMLTRLLLKASDKAPWSDNGDAKNPALSPLGENVKTKGQYFYQVALDGNVAGKE
KQALIDQFRANGTQTYSATVNVYGNKDGKPDLDNIVATKKVTININGLISKETVQ
KAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIPYELFAGDGMLTRLLLKASD
KAPWSDNGDAKNPALSPLGENVKTKGQYFYQLALDGNVAGKEKQALIDQFRANGT
QTYSATVNVYGNKDGKPDLDNIVATKKVTININ                           290 to 597
ININGLISKETVQKAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIPYELFAG
DGMLTRLLLKASDKAPWSDNGDAKNPALSPLGENVKTKGQYFYQLALDGNVAGKE
KQALIDQFRANGTQTYSATVNVYGNKDGKPDLDNIVATKKVTININGLISKETVQ
KAVADNVKDSIDVPAAYLEKAKGEGPFTAGVNHVIPYELFAGDGMLTRLLLKASD
KAPWSDNGDAKNPALSPLGENVKTKGQYFYQVALDGNVAGKEKQALIDQFRANGT
QTYSATVNVYGNKDGKPDLDNIVATKKVTIKIN                           442 to 749
```

Figure 3: An illustration of case study 1 is shown with input sequence to the algorithm and output results.

Edited by P. Kangueane

Citation: Banerjee *et al.*, Bioinformation 3(1): 28-32 (2008)

ISSN 0973-2063 (online) 0973-2063 (print)
Bioinformation 3(1): 28-32 (2008)

Bioinformation by Biomedical Informatics Publishing Group *open access*

www.bioinformation.net **Prediction Model**

Supplementary material

Illustration 1

Code for finding matches

```
pamvalue=pam250(a1[i],a2[j]);
if(pamvalue) { current[j]=previous[j-1] + 1; }
```

pam250 is a function which will return a Boolean value (true or false) corresponding to the match of a1[i] and a2[j]. As an example,

```
if ((i1=='N')&&((i2=='D')||(i2=='Q')||(i2=='E')||(i2=='H')||(i2=='K')    ||(i2=='S'))) return true;
```

In the above code, i1 corresponds to a1[i] and i2 corresponds to a2[j]. The previous step assigns the 'current' length of the repeat to the j^{th} element of the array 'current'. While performing the next iteration, the array 'previous' is assigned the value of array 'current' and the above step is repeated. For example, let the string 'KLN' have a distant repeat 'QLD' such that 'KLN' occurs from positions 6 to 8 and 'QLD' from positions 11 to 13 (Figure 1). Hence the arrays will be a1 = {...KLN…QLD...} and a2 = {...QLD…KLN...}. The changes in the required elements of both the arrays are shown below (as only the upper half above the main diagonal is required the positions 10 to 12 are shown). Initially: current=0 and previous=0; After it finds the first match (K with Q): current [10]=1 and previous=0; Then, previous is assigned the value of current: Previous [10]=1;current [10]=1 and rest all=0. When it finds the next match (L with L): current [11]=2 and the rest remains the same. Similarly after finding the last match (N with D): current [12]=3. Thus, we find that "3" is the length of the repeat and 12^{th} position is the 'end-point' (starting from 0).The above explanation clearly shows how the current array stores both the position and the length of the repeat. The step of pushing the repeat sequences and their positions into vector 'vsparse' is exactly similar to the methodology employed in FAIR.

ISSN 0973-2063 (online) 0973-2063 (print)
Bioinformation 3(1): 28-32 (2008)