# Amino-Acid Sequence Analysis and Pattern Matching Tool ($A^2SA^2P$ M T)

Shejin Murali
B.Tech Student
Department of IT, YCET
kollam-10,kerala,india

Kishore A Nair
B.Tech Student
Department of IT, YCET
kollam-10,kerala,india

Prof. Nijil Raj. N
Associate Professor Department of CSE&IT, YCET kollam-10,kerala,india

***Abstract*—Now a days researchers put much effort for col- lecting and analyzing biological data. So, here in this paper, we introduce a tool($A^2$ $SA^2$ P M T ) for this purpose.Biologists are often interested in performing a simple database search to identify proteins or genes that contain a well-defined sequence pattern. Many databases do not provide straightforward or readily available query tools to perform simple searches, such as identifying transcription binding sites, protein motifs, or repet- itive DNA sequences. However, in many cases simple pattern- matching searches can reveal a wealth of information. We present in this paper a pattern-matching tool that is used to identify short repetitive DNA sequences in human coding regions for the purpose of identifying potential mutation sites in mismatch repair deficient cells.In this paper there are three algorithms used for pattern matching, out of which the user can select the suitable algorithm. Sequences are retrieved from the database and by using this retrieved sequence or new sequence pattern matching is performed.After retrieving the data from the database the result will be downloaded in xls,doc or pdf format.which can be used to find the different properties like frequency of amino acid,hydropobcity value,chain length,etc. can be calculated and it will be downloaded by the doc,xls,pdf formate.**

***keywords: Pattern matching, sequence analysis, DNA,Protein, motifs***

## I. INTRODUCTION

The goal of our project was to study known techniques for discovering patterns in biological sequences. Patterns we want to find usually correspond to functionally or structurally important elements in proteins or DNA sequences. There is an assumption that these important regions are better conserved in evolution and therefore they occur more frequently than expected. Pattern discovery is one of the fundamental problems in bioinformatics. We have concentrated on the problem of discovering previously unknown patterns. From biological point of view it is equally important to have tools for finding known patterns in new sequences, however this is usually not so interesting from algorithmic point of view. Therefore we touch on this issue only lightly, in cases, when it is not obvious. We also do not try to compare individual methods based on their performance or their ability to find most relevant patterns. The reason is that individual approaches vary widely in the type of pattern they try to find, performance guarantees and so on. Even authors of experimental comparative studies such as [2] have difficulties to determine which method performed better on a given dataset. It is impossible to do so based only on descriptions of algorithms. Still we have tried to choose such algorithms to our study that seem to contain most interesting ideas.

## II. BIOLOGICAL MOTIVATION FOR PATTERN DISCOVERY

Nucleotide and protein sequences contain patterns or motifs that have been preserved through evolution because they are important to the structure or function of the molecule. In proteins, these conserved sequences may be involved in the binding of the protein to its substrate or to another protein, may comprise the active site of an enzyme or may determine the three dimensional structure of the protein. Nucleotide sequences outside of coding regions in general tend to be less conserved among organisms, except where they are important for function, that is, where they are involved in the regulation of gene expression. Discovery of motifs in protein and nucleotide sequences can lead to determination of function and to elucidation of evolutionary relationships among sequences

### *A. Pattern discovery in proteins*

With the accumulation of nucleotide sequences for the entire genomes of many different organisms, comes the need to make sense out of all of the information. Attempts have been made to organize all of the proteins encoded in these genomic sequences into families based on the presence of common signature sequences [3], [6]. Members of protein families are often characterized by more than one motif (on average each family has 3-4 conserved regions) which increases the certainty that a protein has been assigned to a correct family [5]. Hierarchical trees of protein clusters often reveal functional and evolutionary relationships among proteins. Starting with a single ”seed” sequence, protein families can be characterized in order to find ancient ancestor sequences [4]. First, proteins related to a query sequence are found by searching the databases for similar sequences. Sequences revealed from this initial screen are then used as query sequences to search for other family members and the process is

repeated to exhaustion. All of the sequences are aligned in order to identify conserved regions which are used to generate models that represent ancient conserved regions. The rationale behind this approach is that if protein A is related to protein B, and B is related to C, then A is also related to C. Model renement parallels divergent evolution in that each subsequent alignment reveals progressively more distant relatives.

By this method, proteins are assigned to a family based on sequence homology as determined primarily by alignment. If an alignment finds homology between a query protein and a particular family of proteins, a phylogenetic relationship between them is automatically assumed [1]. There are two problems with this assumption: 1) significant sequence similarities are not always indicative of close evolutionary relations, and 2) despite limited sequence homology, proteins can have structural and mechanistic similarities, and even common ancestry not apparent through alignment. Perhaps structural information should also be considered when attempting to classify proteins that are highly divergent in homology, yet functionally equivalent.

## III. ALGORITHMS

### *A. Introduction*

Algorithmic approaches to pattern discovery exhibit surprising variety. They can be classified according to different more or less orthogonal criteria. In our report we group algorithms together mainly based on the approaches they use. In this introduction we introduce other possible classifications of the algorithms. We concentrate on two issues: how is the biological task formulated in computer science terms, and what kind of patterns are used in the programs. We also introduce notation used throughout this section.

a) Pattern discovery vs. pattern matching.: So far we have discussed the problem of pattern discovery, i.e. the algorithm is supposed to discover pattern unknown in advance. However in biology many consensus sequences are known and it is important to have tools that allow to find occurrences of known patterns in new sequences. This problem will be called pattern matching. Program for pattern matching can be quite general, i.e. they get pattern as a part of input, or they can be built to recognize only one particular kind of pattern. In this case authors usually try to fine-tune the parameters of the system to get better sensitivity and specificity of the algorithm. From computer science point of view these programs are not so interesting, however they are very useful for biologists.

b) Finding significant patterns.: Motif discovery is not always formulated as a classification problem. For example if we want to find a regulatory element, we might have a set of regions likely to contain this element. However it does not mean, that this element cannot occur in other places in genome or that all of these sequences must contain common regulatory element. Also in a context of protein family motifs we are interested in finding conserved regions that may indicate structurally or functionally important elements, regardless whether they have enough specificity to distinguish between this family and other families. In this context it is more complicated to formulate the question precisely. Usually people define class of patterns they want to find and they are interested in discovering the highest scoring pattern from this class that has enough support.

Support of a pattern usually means the number of sequences in which the pattern occurs. The requirement is that,the pattern should occur in all sequences or there is a minimum number of occurrences specified by user. In some cases the number of occurrences is not specified but it is part of a scoring function, longer pattern with fewer occurrences can be sometimes more interesting than shorter pattern with more occurrences. The situation is even more complicated in the case of probabilistic patterns, such as Hidden Markov models. Deterministic patterns either match sequence or not (zero or one), whereas probabilistic models give a probability between 0 and 1. Therefore there are different degrees of matching". It is necessary to set some threshold on what should be considered a match or to include these matching probabilities to the score of the pattern.

Methods for scoring patterns also differ from paper to paper. Score can describe only the pattern itself (e.g. its length, degree of ambiguity etc.) or it can be based on the occurrences of the pattern (their number, how much these occurrences differ from the pattern). Scoring functions are sometimes based on statistical significance. For example we may ask, what is the probability that the pattern would have so many occurrences if the sequences were generated by random. If this probability is small, the pattern is statistically significant.

The goal of an algorithm may be to find the best (i.e. usually the highest scoring patterns), or to find several best scoring patterns, or all patterns with some predefined level of support and score.

c) Input sequences: The input of pattern discovery programs usually consists of several sequences, expected to contain the pattern. We will denote $p$ the alphabet of all possible characters occurring in the sequences. Thus $p = \{A, C, G, T\}$ for DNA sequences and $p$ is a set of all 20 amino acids for protein sequences. Most of the algorithms can be easily adapted to work with any finite alphabet (this is true for algorithms, but not necessarily for their implementations). Thus the pattern finding algorithm can be used also outside bioinformatics, or on other types of biological data. Some algorithms use not only sequences, but also other information. For example pattern discovery is much easier in aligned sequences. Also we

may use information about secondary or tertiary structure, evolutionary relationships between sequences and so on. However most of the time we will concentrate on the discovery from unaligned sequences only.

## IV. METHODOLOGY

The design process(see in FigureIV) have three steps which are a)sequence extraction,b)pattern matching,c)properties extraction.In the first step upload the protein id or DNA id in the form of xls formate to the tool.Then the sequence which corresponds to the ID is retrieved from the Internet(PDB,BLAST,UNIPORT) and displayed on the Screen in table format.And also it will downloaded as the xls or doc format,second step is Pattern matching. There are 3 different algorithms(1.Knuth Morris Pratt(KMP),2.Boyer Moore(BM),3.Brute Force(BF)) are used for pattern matching. The user can choose any of these algorithm from the tool for pattern matching.The required protein sequence or DNA sequence file can be uploaded to the tool,output will be displayed by the different form of patterns and its occurrence according to the selection of algorithms. in third and final step, some of the properties for the protein or DNA sequence can be calculated. The properties can be like Protein frequency, Sequence length, Hydrophobicity, molecular weight, and DNA frequency can be calculated.In the second step we consider three algorithms which are Brute Force Pattern Matching (BFPM),Boyer Moore(BM),Knuth Morris Pratt(KMP)

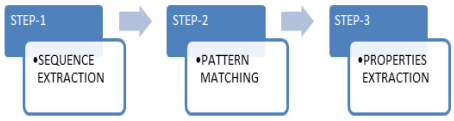


Fig. 1. Design process

### A. Brute Force Pattern Matching (BFPM)

The brute force algorithm is a powerful technique for pattern matching,when we wish to search for or to optimize some function, applying the brute-force pattern matching algorithm, as probably the first algorithm for solving the pattern matching problem-we simply test all the possible placements of P relative to T. The brute-force pattern matching algorithm could not be simpler. It consists of two nested loops, with the outer loop indexing through all possible starting indices of the pattern in the text, and the inner loop indexing through each character of the pattern, comparing it to its potentially corresponding character in the text. Thus, the correctness of the brute- force pattern matching algorithm follows immediately[7] The running time of brute-force pattern matching in the worst case is not good. However, because, for each candidate index in T, we can perform up to m character comparisons to discover that P does not match T at the current index. The running time of the brute force method is O((n -m + l)m), which is O(nm). Thus, in the worst case,when n and m are roughly equal, this algorithm has a quadratic running time.

### B. The Boyer-Moore Algorithm (BM)

The Boyer-Moore (BM) pattern matching algorithm[7], sometimes avoid comparisons between P and a sizable fraction of the characters in T unlike Brute-force in which it is always necessary to examine every character in T in order to locate a pattern P as a substring . The only caveat is that, whereas the brute-force algorithm can work even with a potentially unbounded alphabet, the BM algorithm assumes the alphabet is of fixed,finite size. It works the faster when the alphabet is moderately sized and the pattern is relatively long. The worst-case running time complexity of the BM algorithm is O(nm + |Σ|). Namely, the computation of the last function takes time O(m + |Σ|) and the actual search for the pattern takesO(nm) time in the worst case, the same as the brute-force algorithm.

| Pattern Matching Technique | Time Complexity |
|---|---|
| Brute-Force(BF) | O((n-m+1)∗m) |
| Boyer-Moore(BM) | O(n∗m+Σ\|)\| |
| Knuth-Morris-Pratt(KMP) | O(n+m) |

By comparing the time-complexities of the three patterns matching algorithms the Knuth MorrisPattern matching Algorithm has been implemented in the second step to detect the pattern of DNA or protein sequence in a gene database

### C. Knuth-Morris Pattern matching algorithm(KMP)

In studying the worst-case performance of the brute-force and boyer-moore pattern matching algorithms on specific instances of the problem, we should notice a major inefficiency. Specifically, we may perform many comparisons while testing a potential placement of the pattern against the text, yet if we discover a pattern character that does not match in the text, then we throwaway all the information gained by these comparisons and start over again from scratch with the next incremental placement of the pattern. The Knuth-Morris-Pattern matching (or ”KMP”) algorithm avoids this waste of information and, in so doing, it achieves a running time of O(n + m), which is optimal in the worst case. That is, in the worst case any pattern matching algorithm will have to examine all the characters of the text and all the characters of the pattern at least once.

## V. RESULT AND DISCUSSION

This tool mainly have three modules which will help the researchers for preparing the data as well as the properties of data,figure The user can give the protein ID or DNA ID on the textbox provided here. The sequence can be retrieved for the corresponding protein ID or

the DNA ID. The retrieved sequence will be in FASTA format. This retrieved protein sequence is copied and pasted in the textbox provided for the SEQUENCE. Then perform the pattern matching by selecting any one of the algorithm provided. Here there are two different pattern matching algorithm. They are Boyer Moore algorithm and Brute Force algorithm.

The retrieved sequence which is in FASTA format, that is

Fig. 2. Sequence extraction GUI

obtained by giving a protein ID or DNA ID is given and we select Boyer Moore algorithm for pattern matching. The output obtained when Boyer Moore algorithm is used given in the figure. By using this algorithm, we get the first position of the pattern that the user give in the text box from the sequence.

The retrieved sequence which is in FASTA format that is

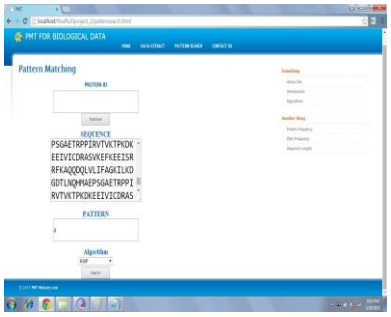


Fig. 3. Pattern matching using different algorithms GUI

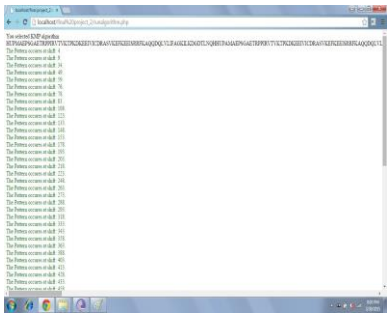

Fig. 4. Output of Pattern matching GUI

obtained by giving a protein ID or DNA ID is given and we select Brute Force algorithm for pattern matching.

The output obtained when Brute Force algorithm is given above. By using this algorithm, we get different combinations of the pattern that the user given in the text box provided from the sequence.

The user can find frequency of the nucleotides in a

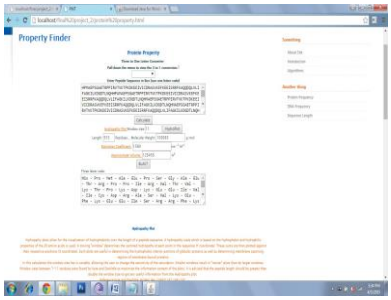


Fig. 5. Sequence property generation GUI

sequence. The retrieved sequence which is in FASTA format that is obtained by giving a protein ID or DNA ID is copied and pasted in the text box provided. By clicking the Generate button, we get the frequencies of the protein sequence that given by the user. The output obtained for the frequency of nucleotides in a sequence is given above.

## VI. CONCLUSION

In conclusion this paper proposes a tool for improving the data process in computational biology.The proposed method help the researchers for the data analysis and preparing the properties of protein and DNA sequence and also this tool can be used to find the different combinations of the pattern from the existing protein and DNA sequence. This tool provides a good GUI and an user-friendly websites to access the corresponding attributes from the database. The project is purely concentrated on researchers. So with the help of this tool the researchers can easily access the information from the database and from the file they can select their required information also.In future we can improve the accuracy of the properties which was extracted from the sequence analysis.

## ACKNOWLEDGMENT

At first I would like to express my heartfelt gratitude to our guide Prof.Nijil Raj.N, Head of The department, computer science and information technology , Younus College of Engineering and Technology, for providing every facility, constant supervision. It gives us immense pleasure to acknowledge a debt of gratitude to our principal Dr.Abdul Mageed, Younus College of Engineering and Technology. Thanks to all the teaching and non-teaching staff of Younus College of Engineering and Technology, for their support and also to my Class-mates for their valuable Co-operation.

## REFERENCES

[1] Pfeier F. Barker, W. C. and D. G. George. ,superfamily classication in pir-international protein sequence database. Methods in Enzymology, 266:59–71, 1996.

[2] Hudak, J. Hudak, and M. A. Mcclure. A comparative analysis of computational motif-detection methods.page-188-189. Pacic Symposium on Biocomputing (PSB), 1999.

[3] Linial N. Tishby N. Linial, M. and G. Yona. Global self-organization of all known protein sequences reveals inherent biological signatures ,268(2):539-546. Journal of Molecular Biology, 1997.

[4] Liu J. S. Lipman D. J. Neuwald, A. F. and C. E. Lawrence. Extracting protein alignment models from the sequence database. Nucleic Acids Research, 1997.

[5] Wu T. D. Nevill-Manning, C. G. and D. L. Brutlag. Highly specific protein sequence motifs forgenome analysis. Proceedings of the National Academy of Sciences of the United States of America, 95(11):5865–5871, 1998.

[6] Floratos A. Ouzounis C. Gao Y. Rigoutsos, I. and L. Parida. Dictionary building via unsupervised hierarchical motif discovery in the sequence space of natural proteins. Proteins, 37(2):264-267., 1999.

[7] Dr.L.S.S.Reddy S.Rajesh, S.Prathima. ,unusual pattern detection in dna database using kmp algorithm. International Journal of Computer Applications, 1 NO:22::0975 – 8887, 2010.