Computational Biology and Bioinformatics
2016; 4(1): 1-9
Published online November 20, 2015 (http://www.sciencepublishinggroup.com/j/cbb)
doi: 10.11648/j.cbb.20160401.11
ISSN: 2330-8265 (Print); ISSN: 2330-8281 (Online)

A Comparative Analysis of Motif Discovery Algorithms

Angela Makolo[1, 2]

[1]Department of Computer Science, University of Ibadan, Ibadan, Nigeria
[2]Bioinformatics Research Group (UNIBReG), University of Ibadan, Ibadan, Nigeria

Email address:
a.makolo@ui.edu.ng

To cite this article:
Angela Makolo. A Comparative Analysis of Motif Discovery Algorithms. *Computational Biology and Bioinformatics*. Vol. 4, No. 1, 2016, pp. 1-9. doi: 10.11648/j.cbb.20160401.11

Abstract: One of the major challenges in bioinformatics is the development of efficient computational algorithms for biological sequence motif discovery. In the post-genomic era, the ability to predict the behavior, the function, or the structure of biological entities or motifs such as genes and proteins, as well as interactions among them, play a fundamental role in the discovery of information to help explain biological mechanisms. This necessitated the development of computational methods for identifying these entities. Consequently, a large number of motif finding algorithms have been implemented and applied to various organisms over the past decade. This paper presents a comparative analysis of the latest developments in motif finding algorithms and proposed an algorithm for motif discovery based on a combinatorial approach of pattern driven and statistical based approach. The proposed algorithm, Suffix Tree Gene Enrichment Motif Searching (STGEMS) as reported in [30], proved effective in identifying motifs from organisms with peculiarity in their genomic structure such as the AT-rich sequence of the malaria parasite, *P. falciparum*. The empirical time analysis of seven motif discovery algorithms was evaluated using four sets of genes from the intraerythrocytic development cycle of *P. falciparum*. The result shows that algorithms based on a combinatorial approach are more desirable.

Keywords: Motifs, Suffix Tree, Time Complexity, *P. falciparum*

1. Introduction

In the post-genomic era, the ability to predict the behavior, the function, or the structure of biological entities (such as genes and proteins), as well as interactions among them, play a fundamental role in the discovery of information to help explain biological mechanisms. [39].

Several functional and structural properties and also evolutionary mechanisms, can be predicted either by the comparison of new elements with already classified elements, or by the comparison of elements with a similar structure or function and using it to infer the common mechanism that is at the basis of the observed similar behavior. Such elements are commonly called *motifs* [11].

Comparison-based methods for sequence analysis find their application in several biological contexts, such as extraction of transcription factors, DNA binding sites, identification of structural and functional similarities in proteins, and phylogeny reconstruction, Therefore, the development of adequate methodologies for genomic sequence analysis is of paramount interest in computational biology. In other words sequence analysis refers to the process of subjecting a DNA, RNA or protein sequence to any of a wide range of analytical methods to understand its features, function, structure or evolution. Sequence analysis algorithms are basically classified into three.

The first classification is Gene Finding Algorithms. These algorithms are used to predict gene structure. Gene prediction or gene finding refers to the process of identifying the regions of genomic DNA that encode genes. This includes protein-coding genes as well as RNA genes, but may also include prediction of other functional elements such as regulatory regions. Gene finding is the first step in sequence analysis procedure. This is because the genes in the genome of any specie that had just been sequenced had to be annotated before any further processing can take place. The operating principle of gene finding algorithms is relatively simple; it is basically based on an inference system that can decode the twenty amino acids using the genetic code. Some popular gene finding tools are GENESCAN, GENEMAK, GENIE, HMMGENE, PHAT etc. [8, 43].

Sequence Alignment Algorithms is the second classification of sequence analysis algorithms. These are algorithms that align genomic sequences to detect similarity. Sequence Alignment is a way of arranging the sequences of DNA, RNA, or protein to

identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns. The building paradigm of Sequence Alignment algorithms is usually more complex than gene finding algorithms. Popular sequence alignment tools include BLAST, ClustalW, T-coffee, FASTA3x among others. [35, 43, 46].

The last classification of sequence analysis algorithms is Motif Discovery Algorithms. These are algorithms that predict patterns from the sequence data hypothesised to have biological functions such as gene regulation. This class of algorithm is the most complicated of the three categories of sequence analysis algorithms available. Primarily due to the complicated makeup of the motifs been sought and therefore require exquisite methodologies to effectively predict them. Some popular tools in this class include MEME, WEEDER and MUSA.

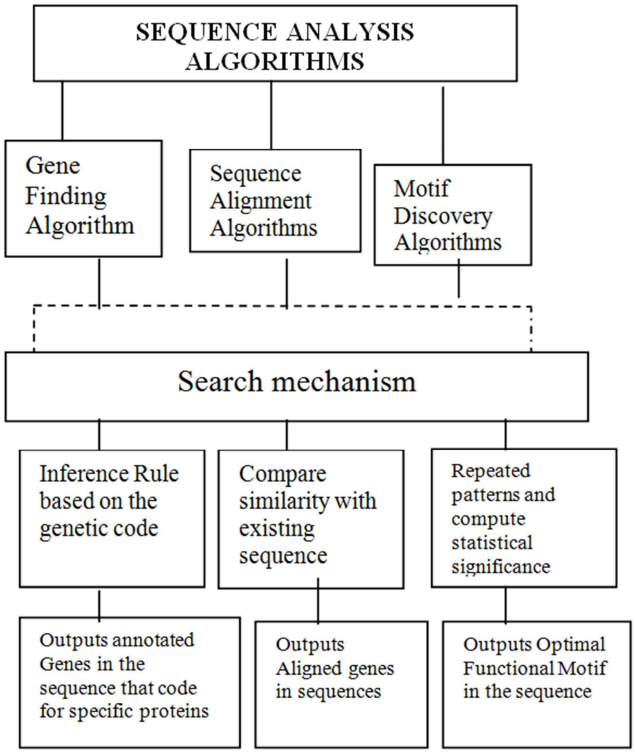

Figure 1. *Hierarchy of Sequence Analysis Algorithms.*

Motif discovery algorithms are based on the biological theory of high conservation which states that patterns repeated in a sequence data with high frequency is a potential motif or pattern of interest and needs to be mined effectively. [54]. The goal of motif discovery algorithms is to enumerate these patterns repeated with high frequency, since they have been established experimentally to have biological significance. The task is to eliminate those randomly occurring patterns, which could result in false positive prediction and report only the best motifs.

The development of adequate methodologies for motif discovery is of unquestionable interest for several different fields in computational biology, therefore different researchers have adopted several approaches to extract these patterns such as pattern-driven approach, statistical based approach and machine learning based approach. All known motif discovery algorithms are, based on one or a combination of two or three of these approaches. Among the most popular methods are those based on the pattern driven approach methods, which uses several heuristics to extract candidate motifs and thereafter performs a validation check using statistical methods to extract candidate motifs with optimal features based on the statistical significance analysis.

Motif discovery is an application area in the field of data mining in computer science. It is concerned with identifying and extracting relevant patterns hypothesized to have biological significance. Usually, a large data set is provided, then the data mining task involves the use of efficient techniques to mine the relevant patterns contained in the data set [40].

Pattern discovery in DNA sequences is one of the most challenging problems in molecular biology and computer science. In its simplest form, the problem can be formulated as follows: given a set of biological sequences, find an unknown pattern that occurs frequently. If a pattern of n letters long appears exactly in every sequence, a simple enumeration of all n-letter patterns that appear in the sequences gives the solution. However, when one works with DNA sequences, it is not that simple because patterns include mutations, insertions or deletions of nucleotides.

Although there are experimental approaches for extracting regulatory motifs, such as DNA footprinting and Chromatin Immuno-Precipitation (chIP) methods; these approaches, however, are time consuming and very laborious. These weaknesses justify the need for computational methods to complement the experimental methods [41, 46].

2. Materials and Methods

Various researchers have adopted several approaches to extract motifs. The main approaches are pattern-driven approach, statistical based approach and machine learning based approach. We shall attempt a review of some common motif discovery tools by grouping them into these three approaches while differentiating the tools used for simple motif and structured motif extraction. The identification of structured motifs (several simple motifs separated by spaces) is more involving because of the variable length spaces that are present in its makeup; therefore, the algorithms for its extraction require special tuning to identify the relevant motifs. [1].

a) Pattern-driven Motif Discovery Algorithms

Pattern-driven method enumerates all the patterns in order to determine those appearing with a high frequency in the input sequence. It also considers the number of possible substitutions and thereafter provides a ranking for the extracted patterns according to some statistical measure of significance. The drawback in this approach is that they can have many false predictions, since they are not good at discriminating the relevant extracted motifs from the

potentially numerous false hits. In addition, this method requires a large number of parameters to be specified. [3].

The common techniques used in Pattern-driven Motif Discovery include enumeration (listing items in an order), suffix tree, graph, hash table and link list. However, our review of pattern-driven motif discovery tools will be based on the tools that utilized suffix tree and enumeration techniques, which are more relevant to this study.

The first simple motif discovery algorithm to use the suffix tree was developed by [45]. The suffix tree was used to represent the sequences, returning all the traversal from the root node to the leaf node as unique patterns. *The use of the suffix tree for preprocessing and organizing the input data resulted in an accelerated search for motifs. This implementation addressed to a large extent the speed bottleneck inherent in pattern-driven based methods*. This was followed by [2] who developed the VERBUMCULUS algorithm and applied it to protein sequences. In [13] a variant of the suffix tree called a mismatch tree is used to develop the MITRA algorithm, which detected complex motifs with mutations successfully. WEEDER algorithm by [36] also used the suffix tree and identified simple motifs allowing the flexibility of parameter specification by users [14, 3, 17, 25].

A linear time suffix tree construction was achieved in SLI-REST (Suffix Link on Internal nodes-Reverse Engineering Suffix Tree) by incorporating edges (suffix links) of different types into all the internal nodes of the suffix tree. To realize the input suffix tree and links, a word is generated through a bi-coloured directed graph on a subset of the suffix tree's internal nodes defined on its edges [9]

The first structured motif extraction algorithm that used the suffix tree was developed by [32]. They extended the simple motif extraction algorithm developed by [45] to extract *structured motifs*. Their algorithm, SMILE proposed two solutions for extracting structured motifs on the suffix tree. In the first solution, the structured motif template consists of two components with a gap range between them, the algorithm starts by building a generalized suffix tree for the input sequences and then extracts the first component. In order to extract the other component, a jump is made in the sequences from the end of the first component to the second within the gap range. In the second solution, the suffix tree is modified temporarily so as to extract the second component from the modified suffix tree directly. SMILE proved inefficient in terms of its time and space complexity, which were exponential in the number of gaps between the two components.

In [6], attempt was made to reduce the time complexity during the extraction of the structured motifs by SMILE and developed a parallel algorithm, called PSMILE. PSMILE used the technique of partitioning the structured motif searching space; this achieved a time speedup, which is linear on the number of available processing units. A year later, the same authors developed the RISO algorithm, an improvement on the SMILE algorithm. [5, 6, 7] This improvement is twofold: the first, instead of constructing the whole suffix tree for the input sequence, built a suffix tree only up to a certain level, which was called the factor tree; this resulted in saving appreciable space. Secondly, a new data structure called box-link was introduced to store the information about how to jump within the DNA sequences from one simple motif component to the subsequent one in the structured motif. This accelerated the extraction process and avoided the exponential time and space consumption that prevailed in the case of SMILE. In RISO, after the generalized factor tree was built, the box-links were constructed by exhaustively enumerating all the possible structured motifs in the sequences and they were added to the leaves of the factor tree. Then the extraction process began, during which the factor tree was temporarily and partially modified in order to extract the subsequent simple motifs. RISO needed a lot of computation at this stage since the box-link construction, the structured motif occurrences were exhaustively enumerated and the threshold of the sequences was never used to prune the candidate structured motifs [21, 24, 28].

An improvement on the RISO algorithm was provided by [38] by developing the RISOTTO algorithm. RISOTTO incorporated boxlinks data structure with the suffix tree. While traversing the tree, RISOTTO adopted a depth-first visit of the motif tree and does not attempt to extend the node if the maximal length was determined or the quorum was no longer satisfied. The main improvement of RISOTTO on RISO was its ability to store information concerning maximal extensibility of factors. This was done in order to avoid extending motifs that are unlikely candidates. RISOTTO was shown to outperform RISO in terms of computational speed. However, it incurred an extra cost due to the space required to store the extensibility information.

Another popular structured motif extraction algorithm, EXMOTIF by [54] used a variant of the suffix tree, consisting of inverted index of symbol positions. This was used to enumerate all structured motifs by positional joins over the index. EXMOTIF was reported to outperform RISO in both approximate and exact matching and superior to RISSOTO in showing the actual occurrences of the structured motifs instead of the relative frequency of the occurrence as obtained using RISOTTO. [17, 37]

Apart from simple and structured motif, suffix tree has been employed to extract motifs from large ranked list of sequences by [26, 27]. They used the same approach called DRIMust, [27] is a web server that support the search of imbalanced motif while [26] allows searching of variable gap motifs and long motifs over large alphabets. They return motifs that are over-represented at the top of the list with their corresponding P-values, which serve as Position Specific Scoring Matrix (PSSM) and the threshold used for the top is data driven through enrichment analysis (minimum hyper-geometric analysis). They are both efficient in searching for long motifs even in large data set that are not fixed sequences, with short running time.

b) Statistical Based Motif Discovery Tools

Statistical based method uses a two-phase iterative procedure where in the first step the likeliest occurrences of the motif are identified, and the second step adjusts the model

for the motif, which is usually represented by a position scoring weight matrix (PSWM) model based on the occurrences of the motifs determined in the previous step. In the first iteration the parameters of the initial model are usually set randomly. The limitation in this method is sensitivity to noise in the data and the fact that they are not guaranteed to converge to a global maximum since they employ some form of local search, such as Gibbs sampling, expectation maximization (EM) or greedy algorithms that may converge to a locally optimal solution [12,46,40,47].

Some common techniques used by statistical based method include expectation maximization, profiling using position specific scoring matrix and gene enrichment analysis.

MEME (Expectation Maximization Motif Elicitation) by [4] and PHYME [36] used expectation maximization and position scoring matrix. While DRIM, by [13] and GEMS by [52] used gene enrichment analysis. EXTREME is an extension of MEME algorithm, which used an online EM to discover novel and infrequent motifs in large dataset like ChIP-Seq and DNase-Seq data [42]. Also, stochastic EM approach with an improved approximation to the likelihood function instead of normal deterministic EM was used in [23] to make the algorithm escape the local maxima and converge to models with higher energies.

In [52], the GEMS algorithm used the gene enrichment technique and incorporates the statistical test of hypergeometric mean instead of the geometric mean used by [13]. GEMS also introduced the position weight matrix optimization principle, which improved the accuracy of the motifs discovered. GEMS algorithm is not an ab-initio motif discovery tool, it requires an already existing cluster as candidate motif to perform gene enrichment analysis on.

Finding DNA motifs with adjacent and non-adjacent positional dependencies was established in [51]. triPWDM model was used to capture interdependencies within 3 neighboring nucleotides while diSPWDM model was used to dynamically capture pairing dependencies at any two positions in the motif. Gibbs sampling approach was employed to update the model parameter and dependencies structure.

c) Machine Learning Based Motif Discovery Tools

Several motif discovery algorithms used different machine learning techniques as their operating principle. The most common machine learning technique used in motif inference tool is the genetic algorithm. The advantage of such genetic algorithm based methods is that they are likely to locate the global optimum in a typically difficult search space. On the other hand, they are stochastic and so they may fail to report consistent results in different runs.

A few popular motif discovery tools based on genetic algorithm are FMGA (Finding Motif with Genetic Algorithm) by [29], GAME (Genetic Algorithm Motif Elicitation) by [50], MOGOMOD (Multi-Objective Genetic Algorithm Motif Discovery) by [20], GARPS (Genetic Algorithm with Random Projection Strategy) by [18] and [15]

MOGAMOD used the multi-objective genetic algorithm to discover optimal motifs in sequential data. Multi-objective optimization involves having a solution which is a family of pareto-optimal set or non-dominated solutions. The optimal motif discovery problem was converted into three conflicting optimization problems of maximizing similarity, increasing motif length and support for candidate motifs. The implementation of MOGAMOD was based on a well known high performance multi-objective Genetic Algorithm called NSGA II(Non- dominated Sorting Genetic Algorithm) by [10, 16, 49].

The sensitivity of MOGAMOD is further enhanced by its flexibility in choice of similarity measures for finding motifs. The user can analyze the obtained optimal motifs, and makes decision on the tradeoff between the different objectives. A detailed comparison of this similarity measure and that used by other popular motif discovery algorithms was reported in [31].

In [15], an iterative approach was employed to increase the computational efficiency of motif searching by using parallel random search. A new operation was added to the 3 operations of GA to achieve the algorithm. The computational efficiency was increase in GARPS by reducing the search space through RPS (Random Projection Strategy). RPS was to find good starting positions in the input sequences so as to infer possible candidate motifs and the candidate motifs are set as the population of the GA to iteratively refine and identify the best motifs. So GARPS is a combination of 2 approaches for motif discovery.

d) Motif Discovery Tools Based On Combinatorial Approach

A number of motif discovery algorithms combine two or more approaches to get a hybrid approach, which inherits desired features of the various approaches. This concept was reported by [34] in their study on survey of motif discovery tools.

A popular motif discovery tool in this category is MUSA (Motif finder with UnSupervised Approach) based on a combination of machine learning and statistical technique [34]. MUSA used a bi-clustering algorithm that operates on a matrix of co-occurrences of simple motifs and computed the statistical significance using position weight matrix. MUSA successfully identified complex biologically significant motifs with a performance that was independent of the composite structure of the motifs being sought. MUSA could be used as a standalone tool or as a tool to determine the parameters required to run other motif discovery tools already available because of its effective statistical significance assessment method. MUSA was validated both with synthetic and real data from yeast, and it was able to discover new biologically significant motifs that had eluded searches performed using other motif finders such as MEME and AlignAce. [33].

BioProspector by [29] also combined Gibbs sampling statistical technique with a machine learning markov model. While APMotif by [48] used Affinity propagation (AP) clustering to find candidate motifs, and used EM to search for optimal motifs from the candidate motifs.

A common choice among researchers of motif discovery tools is a combination of pattern-driven and statistical-based

methods since this approach guarantees that the sensitivity of the statistical based method be complemented with the speed efficiency of pattern-driven techniques. An example of this is the STEME (Suffix Tree and Expectation Maximization for Motif Elicitation) algorithm by [44].

This notion also influenced the design methodology of STGEMS algorithm by [30]. It combined the suffix tree, a pattern-driven approach with Gene Enrichment Motif Searching, a statistical approach**.** The incorporation of the suffix tree improved the speed limitation of the statistical based method.

A list of the motif discovery tools reviewed in this study can be found in the appendix section.

3. Result

Our result shows a comparative analysis of some popular motif discovery algorithms in terms of their empirical runtime.

The algorithms used are MEME and GEMS (statistical based motif discovery tools), WEEDER, RISOTTO and EXMOTIF (pattern driven motif discovery tool), MOGAMOD (a machine learning based motif discovery tool) and STGEMS (algorithm based on a combinatorial methodology).

The experiment was conducted using sets of genes from the intraerythrocytic development cycle of *P. falciparum* downloaded from PlasmoDB (An online database of P.*falciparum* genes maintained by National Center for Biotechnology Information (NCBI) http://www.ncbi.nlm.nih.gov).

Four different sizes of genes were used in the analysis i.e. 20,000, 40,000, 60,000 and 80,000bp, this variation in gene sizes was chosen to enable a classification of the performance of the algorithms as a function of input size.

The empirical run time evaluation of the seven motif discovery tools are shown in figure 2.

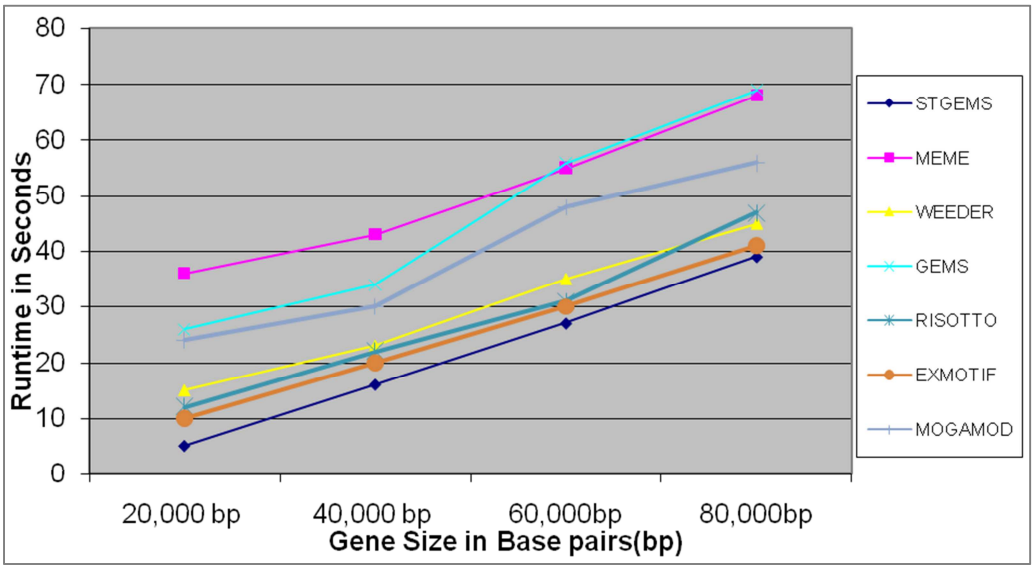

Figure 2. *Empirical Runtime Comparison of some popular Motif Discovery Algorithms.*

The empirical runtime of the different algorithms was obtained by including a time stamp in the algorithm so that its output displays the execution time. From figure 2 above, it is clear that the empirical run time of all the algorithms tested increased as the size of input increased. The run time of the MEME algorithm, which is a statistical based motif discovery tool, was higher than all other algorithms that is MEME had the lowest performance among all the other tools compared since its running time over the set of the selected input was the highest. This is followed by GEMS, which is also based on a statistical method. MOGAMOD, a machine learning tool has an average run time performance. The pattern-driven methods WEEDER, RISOTTO and EXMOTIF are much faster than the statistical based tools; this speed performance gain is attributed to the fact that they all used the suffix tree data structure, which is known to enhance searching speed. EXMOTIF performed better than RISOTTO and WEEDER because it incorporated the use of suffix links in its implementation of the suffix tree data structure. STGEMS outperformed all the six algorithms. This is because the framework of STGEMS was built on an implementation of the suffix tree using linked list and hash table data structures unlike EXMOTIF, RISOTTO and WEEDER algorithms that did not incorporate these combined features.

4. Conclusion

Motif discovery is the process of identifying and extracting patterns believed to have biological importance and therefore necessary for understanding the complex biological mechanisms of living organisms. The variety of techniques adopted in the design paradigm of the various motif inference tools shows an increasing effort of researchers to develop efficient algorithms for genomic functions predictions. The efficiency of these algorithms is measured in terms of their time complexities, which in turn is affected by the choice of data structures used in the design paradigm.

From this large number of available tools for motif finding,

users would like to have guidance in choosing the best tool. However, the assessment of the performance of the tools in identifying biologically motivated patterns is still a difficult task. This is mainly because we do not have a clear understanding of the biology of regulatory mechanisms; therefore, we lack an absolute standard against which to measure correctness of tools. Nevertheless, most of the algorithms compare the motifs identified using *in-silico* methods with those extracted using wet-lab methods. A high correlation between these two outputs gives a good performance measure to some extent.

We agree with Tompa [49], that biologists should use a few complementary tools in combination rather than relying on a single one and pursue the top few predicted motifs of each rather than the single most significant motif. In the same vein, STGEMS which is based on a combinatorial approach of Pattern-driven and statistical method had a remarkable performance. This combinatorial approach guaranteed the incorporation of the speed efficiency of pattern-driven method with the improved predictive ability of the statistical based methods.

Appendix

Table 1. *List of Motif Discovery Algorithms.*

S/N	Algorithm	Category	Operating Principle	Strengths	Weakness	Reference
1.	By Hert et al	SBA	Greedy Algorithm	Simple to implement	It is not time efficient	Hertz and Stormo(1990)
2.	MEME	SBA	Expectation maximization	Prior knowledge of the sequence is not required	It cannot run large data set at once	Bailey and Elkan (1995)
3.	AlignACE	SBA	Gibbs Sampling	Displays frequency of non site sequence at a glance	Not time efficient	*Roth* et al, (1998)
4.	CONSENSUS	SBA	Weight Matrix	Detects evolutionary relationship	Building the evolution tree takes time	Hertz and Stormo(1999)
5.	PhyME	SBA	EM	Shows evolutionary relationship at a glance	Extra time to construct the evolution tree	Sinha et al., (2004)
6.	Oligo-Analysis	PDA	Enumeration	Easy to implement	It cannot handle motifs with mutation	Van Helden *et al*. (1998).
7.	WEEDER	PDA	Suffix Tree	Allow flexible parameter specification	It can only return simple motif	Pavesi(2001)
8.	By Sagot	PDA	Suffix Tree	Improved speed	It can only return simple motifs	Sagot(1998)
9.	By Tompa	PDA	Enumeration	Good at discriminating randomly occurring motif	Cannot handle motifs with mutations	Tompa (1999)
10.	Verbumculus	PDA	Suffix tree	Improved speed of execution	It can only return simple motifs	Apostolico *et al*. (2001)
11.	SMILE	PDA	Suffix Tree	It can identify complex structured motifs	Space inefficient	Marsan and Sagot(2000)
12.	YMF	PDA	Enumeration	Allow flexible parameter specification	It can only return simple motif	Sinha and Tompa(2000)
13.	BioProspector	SBA & MLA	Gibbs Sampling and hidden markcov	Allows Multiple optimal motif detection	Very slow with large data set	Liu et al.,(2001)
14.	DRIM	SBA	Hyper geometric Framework	Added feature of Ranking motifs	Too slow especially for large data set	Eden et al (2007)
15.	GEMS	SBA	Gene Enrichment	Identified simple motifs in the malaria parasite	Cannot identify structure motifs	Young et al (2008)
16.	MITRA	PDA	PrefixTree/Mismatch tree and Graph	Allow preprocessing of sequences	Space inefficient	Eskinand Pevzner(2002)
17.	PSMILE	PDA	Suffix Tree	Partitioning of search space that can run on parallel systems	Extra cost of space due to the partitioning	Carvalho et al (2004)
18.	RISO	PDA	Box links and suffix tree	Additional speed gain due to boxlinks	Additional Space requirement for the box link	Carvalho et al (2005)
19.	RISOTTO	PDA	Box links and suffix tree	Good for long complex motifs	Extra space need to store Extensibility information	Pisanti et al., (2006)
20	EXMOTIF	PDA	Inverted index of symbols and hash table	actual occurrences of the structured motifs instead of the relative frequency	Additional space requirement for storing the symbols	Zang and Zaki (2006)
21.	FMGA	MLA	Genetic Algorithm	Can handle difficult search space	Time consuming	Liu et al. (2004)
22.	GAME	MLA	Genetic Algorithm	Return high fitness motif	Inconsistent in multiple runs	Wei and Jensen (2006)
23.	MUSA	MLA & SBA	Biclustering and PSSM	No need to specify parameter and can be used to determine the parameter needed for other algorithms	The speed is unacceptable especially for large data set	Mendes et al(2006)
24.	MOGAMOD	MLA	Multi Objective	Handles multiple optimal motifs	It is time consuming	Mehmet Kaya (2007)

S/N	Algorithm	Category	Operating Principle	Strengths	Weakness	Reference
			Genetic Algorithm	efficiently		
25.	By Mehmet Kaya	MLA	Multi-objective GA	Can identify structured motifs	It is time consuming	Mehmet Kaya (2009)
26	MOTIFST	PDA	Suffix Tree	Fast	It cannot identify motif in the malaria parasite genome	Zare-Mirakaba et al. (2009)
27.	STEME	PDA &SBA	Suffix tree, Expectation maximization	Fast and very sensitive	Can only identify simple motifs	Reid J. and Wernisch L (2011)
28	STGEMS	PDA & SBA	Suffix tree Gene Enrichment	Fast, has a high predictive ability, can identify simple and structured motifs		Makolo et al (2012)
29	DRIMust	PDA	Suffix tree	Handles large data set, ranked lists and fast. Timely interaction with the results. User's friendly. Fast.		Leibovich et al.(2013)
30	SLI-REST	PDA	Suffix tree	Linear time motif construction.	Cannot handle implicit suffix trees	Cazaux and Rivals(2014)
31	GARPS	MLA	Genetic Algorithm	Can handle difficult search space. Improves finding faint motifs.	Time consuming	Fan et al.(2013)
32	EXTREME	SBA	Expectation Maximization	Returns novel and infrequent motif. Handles big data. Online.	Time consuming. Noise affect consistency in efficiency	Quang and Xie (2014)
33	APMotif	MLA & SBA	Affinity Propagation Clustering & Expectation Maximization	High prediction accuracy. Identify weak motifs. Reduce effect of local optimum.	Cannot handle large data set.	Sun et al. (2015)
34	MISTU	SBA	Stochastic Expectation Maximization	Increase site-level sensitivity		Kilpatrick et al. (2014)
35	TRIPWDM	SBA	Gibb Sampling	Increased Sensitive		Wu et al. (2013)

Legend:
PDA- Pattern Driven Approach
SBA- Statistical Based Approach
MLA- Machine Learning Approach

References

[1] Adebiyi, E. F. (2011) CODE MALARIA: Eradication developments for the decade. *Covenant University 1st Inaugural lecture mini-book*. Covenant University Press.

[2] Apostolico, A., Parida, L. and Rombo, S.E. (2001). Efficient detection of unusual words J. Comput. Bio., 7(1/2), pp 71-94

[3] Apostolico, A., Parida, L. and Rombo, S.E. (2008). Motif patterns in 2D, *Theoretical Computer* Science 390(1), pp 40–55

[4] Bailey, T. L. and Elkan, C. (2000). MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Research*, 34, Web Server issue W369–W373.

[5] Bischo, E. and Vaquero, C.(2010) In silico and biological survey of transcription-associated proteins implicated in the transcriptional machinery during the erthrocyctic development of Plasmodium falciparum, *BMC Genomics*, 11:34.

[6] Carvalho, A.M., Freitas, A.T., Oliveira, A.L. and Sagot, M.F. (2004). A parallel algorithm for the extraction of structured motifs. *19th ACM Symposium on Applied Computing* pp. 147–153.

[7] Carvalho, A., Freitas, A., Oliveira, A. and Sagot, M. (2005) Efficient Extraction of Structured Motifs Using Box-links. *String Processing and Information Retrieval Conference* 2004. pp. 267–278.

[8] Cawley S., Wirth, A., Speed, T.(2001). PHAT: A gene finding program for Plasmodium falciparum. *Mol Biochem Parasitol.* 118, pp 167–174.

[9] Cazaux, B. and Rivals, E. (2014). Reverse Engineering of Compact Suffix Trees and Links: a Novel Algorithm. J. Discrete Algorithms, dx.doi.org/10.1016/j.jda.

[10] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2), pp182–97.

[11] Deitsch, K. et al. (2007) Mechanisms of gene regulation in Plasmodium *Am J. Trop. Med. Hyg*. 77(2), pp201-8

[12] Dyer, M.D, Murali, T.M. and Sobral, B.W. (2007) Computational prediction of host-pathogen protein interactions *BMC Bioinformatics*; Vol. 23; 159-166.

[13] Eden, E., Lipson, D., Yogev, S. and Yakhini, Z. (2007) Discovering motifs in ranked lists of DNA sequences. *PLoS Comput Biol* 3(3): 239-243

[14] Eskin, E. and Pevzner, P. (2001) Finding composite regulatory patterns in DNA sequences. *BMC Bioinformatics*, 18 Suppl 1:S354, 63-70.

[15] Fan, Y., Wu, W., Liu, R. and Yang, W. (2013). An Iterative Algorithm for Motif Discovery. Procedia Computer Science 24, 25 – 29.

[16] Hon L.S. and Jain A.N.: A deterministic motif finding algorithm with application to the human genome. *Bioinformatics* 2006, 22:1047-1054.

[17] Hu J, Li B, Kihara D: Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Res* 2005, 33:4899-4913.

[18] Huo, H., Zhao, Z., Stojkovic, V. and Liu, L. (2010). Optimizing Genetic Algorithm for Motif Discovery. Mathematical and Computer Modelling, 52, 2011–2020.

[19] Jiang, D., Pei, J., Zhang, A. (2003) DHC: A Density-based Hierachical Clustering Method for Time-series Gene Expression Data. *In Proceeding of BIBE2003: 3*PRDP *IEE International Symposium on Bioinformatics and Bioengineering*, 10-12.

[20] Kaya, M. (2009) MOGAMOD: Multi-Objective Genetic Algorithm for Motif Discovery, *Expert Systems with Applications*, 36 (2): 1039-1947.

[21] Keich, U. and Pevzner, P. A. (2002). Finding motifs in the twilight zone *BMC Bioinformatics*, 18(10):1374-1381.

[22] Kellis, M., Patterson, N., Endrizzi, M., Birren, B., and Lander, E.S. 2004. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* 423, 241–254.

[23] Kilpatrick, A.M., Ward, B. and Aitken, S. (2014). Stochastic EM-based TFBS motif discovery with MITSU. Vol.30, pg. 310–318, doi:10.1093/bioinformatics/btu286.

[24] Knowles, J.D. and Corne, D.W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evol Comput*; 8(2):149–72.

[25] Konak, A., David, W., Coitb, A., Smith, E. (2006). TMulti-objective optimization using genetic algorithms: A tutorial. T *Reliability Engineering and System Safety* 91 pp 992–1007.

[26] Leibovich, L. and Yakhini, Z. (2012). Efficient Motif Search in Ranked Lists and Applications to Variable Gap Motifs. Nucleic Acids Research, 1–16, doi:10.1093/nar/gks206.

[27] Leibovich, L., Paz, I., Yakhini, Z. and Mandel-Gutfreund, Y. (2013). DRIMust: A Web Server for Discovering Rank Imbalanced Motifs Using Suffix Trees. Nucleic Acids Research, Vol. 41, Web Server issue, doi:10.1093/nar/gkt407.

[28] Leung HCM, Chin FYL (2006): Finding motifs from all sequences with and without binding sites. *Bioinformatics*, 22:2217-2223.

[29] Liu, F.M., Tsai, J.J., Chen, R.M., Chen, S.N. and Shih, S.H. (2004). TFMGA: finding motifs by genetic algorithm. T TFourth IEEE Symposium on Bioinformatics and Bioengineering T, 459.

[30] Makolo, A., Ezekiel, A. and Osofisan, A. (2011). Mining Structured Motifs with Gene Enrichment Motif Searching on Suffix tree. *Journal of Computer Science and its Applications,* 18(1), pp 71-78.

[31] Makolo, A., Ezekiel, A. and Osofisan, A. (2012). Comparative Analysis of Similarity Check Mechanism for Motif Extraction. *IEEE African Journal of Computing & ICT*.

[32] Marsan, L. and Sagot, M. (2000). Algorithm for extraction of structured motif using a suffix tree with an application to promoter and regulatory site consensus identification. Journal of Computational Biology, 7(3) pp 45-362.

[33] Mendes, N.D., Casimiro, A.C., Santos, P.M., Correia, I.S, Oliveira, A.L., Freitas, A.T. (2006). MUSA: A Parameter Free Algorithm For the Identification of Biologically Significant Motifs. *Bioinformatics (Oxford Journals)* 22(24), pp 2996-3002.

[34] Modan K. D. and Ho-Kwok, D. (2007). A survey of DNA motif finding algorithms. *Proceedings of the Fourth Annual MCBIOS Conference*. Computational Frontiers in Biomedicine.

[35] Ortet, P. and Bastien, O. (2010). Where Does the Alignment Score Distribution Shape Come from?. *Evolutionary Bioinformatics* 6: 159-187

[36] Pavesi, G., Thomas, M. and Martin V. (2001). An algorithm for finding sequence of unknown length. *Bioinformatics*(Oxford Journals)17(2), pp S207-S214.

[37] Pevzner P, Sze S. (2000): Combinatorial approaches to finding subtle signals in DNA sequences. *In Proceedings of the Eighth International Conference on Intelligent Systems on Molecular Biology*, San Diego, CA, 269-278.

[38] Pisanti, N., Carvalho, A., Marsan, L. and Sagot, M.F. (2006). RISOTTO: Fast extraction of motifs with mismatches. In seventh latin American theoretical information symposium.

[39] Pizzi, C., Rastas, P. and Ukkonen, E. (2011). Motif Discovery with Compact Approaches -Design and Applications, *IEEE/ACM Trans. Comput. Biology Bioinform.* 8(1), pp 69–79.

[40] Prakash A, Blanchette M, Sinha S, Tompa M (2004) Motif discovery in heterogeneous sequence data. *Proceedings of the Ninth Pacific Symposium on Biocomputing* 2004, 348-359.

[41] Pont N. et al.(2010). Nucleosome occupancy at transcription start sites in the human malaria parasite A hard-wired evolution of virulence? *Infect Genet Evol*, 10:1016.

[42] Quang, D. and Xie, X. (2014). EXTREME: An Online EM Algorithm for Motif Discovery. BIOINFORMATICS, Vol. 30, no. 12, pages 1667–1673, doi:10.1093/bioinformatics/btu093

[43] Redhead, E., Bailey, T.L. *(2007)* Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC Bioinformatics. 8, 385.*

[44] Reid, J. and Wernisch, L. (2011). STEME: efficient EM to find motifs in large data sets *Bioinformatics (Oxford Journals)*. 10(2), pp S93-S107.

[45] Sagot. Marie-France (1998). Spelling approximate repeated or common Motifs, PLoS Computational Biology.

[46] Siddharthan R, Siggia ED, van Nimwegen E: PhyloGibbs: A Gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput Biol* 2005, 1:534-556

[47] Sinha, S., Blanchette, M. and Tompa, M. (2004) PhyME: A probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics*, 5,170.

[48] Sun, C., Huo, H., Yu, Q., Guo, H. and Sun, Z. (2015). An Affinity Propagation-Based DNA Motif Discovery Algorithm. BioMed Research International, 853461, dx.doi.org/10.1155/2015/853461

[49] Tompa, M., Li, N., Bailey, T., Church, G., De Moor, B., Eskin, E., Favorov, A., Frith, M., Fu, Y., Kent, W., Makeev, V., Mironov, A., Noble, W., Pavesi, G., Pesole, G., Regnier, M., Simonis, N., Sinha, S., Thijs, G., Van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C. and Zhu, Z. (2005). T Assessing computational tools for the discovery of transcription factor binding sites. TTNat Biotechnol T, T23:T137-144.

[50] Wei, Z. and Jensen, S.T. (2006) GAME: detecting cis-regulatory elements using a genetic algorithm. Bioinformatics, 22, 1577–1584.

[51] Wu, H., Wong, P.W.H., Caddick, M.X. and Sibthorp, C. (2013). Finding DNA Regulatory Motifs with Position-dependent Models. Journal of Medical and Bioengineering Vol. 2, No. 2.

[52] Young, J., Johnson, J., Benner, C., Yan, F., Chen, K., Roch, K., Zhou, Y. and Winzeler, E. (2008). In silico discovery of transcription regulatory elements in Plasmodium falciparum. *BMC Genomics*, 9, 70.

[53] Yuda, M., Iwanaga, S., Shigenubu, S., Mair, G., Janse, C., Waters, A., Kato, T. and Kaneko, I. (2009) Identification of a transcription factor in the mosquito-invasive stage of malaria parasite. *Molecular Microbilogy*, 71, 1402-1414

[54] Zhang. Y. and Zaki, M. (2006). EXMOTIF: Efficient structured motif extraction. *Algorithms for Molecular Biology*, 1, 21.