International Journal of Systems Science and Applied Mathematics
2018; 3(4): 67-73
http://www.sciencepublishinggroup.com/j/ijssam
doi: 10.11648/j.ijssam.20180304.11
ISSN: 2575-5838 (Print); ISSN: 2575-5803 (Online)

An In-depth Evaluation of an Optimized Algorithm for DNA Motive Discovery

Amannah Constance Izuchukwu[1], Ernest Chukwuka Ukwosah[2]

[1]Department of Computer Science, Ignatius Ajuru University of Education, Port Harcourt, Nigeria

[2]Department of Computer Science, Federal University, Wukari, Nigeria

Email address:

aftermymsc@yahoo.com (A. C. Izuchukwu)

To cite this article:

Amannah Constance Izuchukwu, Ernest Chukwuka Ukwosah. An In-depth Evaluation of an Optimized Algorithm for DNA Motive Discovery. *International Journal of Systems Science and Applied Mathematics.* Vol. 3, No. 4, 2018, pp. 67-73.
doi: 10.11648/j.ijssam.20180304.11

Received: May 29, 2018; **Accepted**: June 27, 2018; **Published**: March 7, 2019

Abstract: Motive Finding is the process of locating the meaningful patterns in the sequence of DNA, RNA or Proteins. There are many widely used algorithms in practice to solve the motive finding problem and these methods are local search methods. Different search algorithms were discussed which are Gibbs sampling, projection, pattern branching, and profile branching. The limitations surrounding them gave an advantage for the selection of the best algorithm in producing an optimized algorithm for the DNA discovery.

Keywords: Motive, Optimization, Algorithm, DNA, RNA, Discovery

1. Introduction

Motive Finding is the process of locating the meaningful patterns in the sequence of DNA, RNA or Proteins. The patterns are not exact copies due to biological reasons. So, the motive finding problem tends to be an NP-Complete Problem. It is one of the key areas of interest for a number of researchers. There are different types of motives in literature namely: Sequential Motives, Gapped motives, Structured Motives, Planted Motives and Network Motives. A number of methods, algorithms and tools have been developed in recent years to solve these problems. Gibbs Sampler and MEME are most widely used in practice to solve the motive finding problem and these methods are local search methods. For planted motives, Random Projections and Pattern Branching got better results compared to others. All these methods suffered from the problem of local optima. In the recent days many Evolutionary Computational Techniques/Evolutionary Algorithms (EAs) are being tried with different coding schemes and different objective functions to eliminate local optima.

Among these, Genetic Algorithm (GA) is one of the most widely used algorithms to find motives. Though GA's help overcome the problem of local optima, it is only to some extent and it is possible only at the cost of exercising more operators

In Bioinformatics, Motive finding is one of the most popular problems, which has many applications. Generally, it is to locate recurring patterns in the sequence of nucleotides or amino acids. As we can't expect the pattern to be exact matching copies owing to biological mutations. By approximating the same in different aspects, scientists have provided many solutions in literature.

The most of the algorithms suffer with local optima. Particle swarm optimization (PSO) is a new global optimization technique which has wide applications. It finds the global best solution by simply adjusting the trajectory of each individual towards its own best location and towards the best particle of the swarm at each generation.

Statement of the Problem

The critical issue is DNA motive finding remains a complex challenge for biologists and computer scientists. Performance comparison of different motive finding tools and identification of the best tools have proven to be a difficult task because tools are designed based on algorithms and motive models that are diverse and complex and our incomplete understanding of the biology of regulatory mechanism does not always provide adequate evaluation of underlying algorithms over motive models. Thus, a more

realistic motive evaluation model giving appropriate consideration to the domain knowledge could provide better motive predictions.

Aim and Objectives of the Study

The aim of this research was to design an optimized algorithm for DNA motive discovery. The following objectives were taken to achieve the aim of the study.

i. to ascertain the DNA motive prediction algorithms;
ii. to ascertain the operation procedure of the DNA prediction algorithm.
iii. to design a template for DNA motive prediction algorithm.

Scope of the Study

The study focuses on motive finding using an optimized algorithmic method. a large number of motive finding algorithms are available, therefore, users may like to have some guidance in choosing the best tools for their motive finding endeavour.

2. Related Literature

A gene is a segment of DNA that is the blueprint for protein. Basically, the control of gene regulation is determined by the chemical reactions which are, in turn, controlled by the shape and electrostatic charges of the molecules involved. In 1950, Francois Jacob and Jacques Monod first discovered regulation genes, in Paris. These genes provide the instructions for creating proteins to control the expression of the other structural genes and play a key role in gene expression [1].

In order to regulate the gene expression process, a molecule called transcription factor will bind to a short substring in the promoter region of the gene. We call this substring as a binding site of the transcription factor. A single transcription factor can be bound to multiple binding sites. We refer to these binding sites as "Motives".

Motives are fundamental functional elements in proteins. These patterns are vital for understanding gene function, human disease, and may serve as therapeutic drug target. Motives can be used to determine evolutionary and functional relationships of the genes.

Motives vary in lengths, positions, redundancy, orientation and bases. Finding these short sequences (motives or signals) is a fundamental problem in molecular biology and computer science with important applications such as knowledge-based drug design, forensic DNA analysis, and agricultural biotechnology [2].

A DNA motive is defined as a nucleic acid sequence pattern that has some biological significance such as being DNA binding sites for a regulatory protein, i.e., a transcription factor. Normally, the pattern is fairly short (5 to 20 base-pairs (bp) long) and is known to recur in different genes or several times within a gene [3].

Motives are patterns in biological sequences which can indicate the presence of certain biological characteristics. In general, these could represent patterns in any kind of biological sequences such as DNA sequences, RNA sequences, protein sequences etc. Some of the features of motives are:

i. They are patterns of length, 10 to 25 bases, and are repeated over many sequences.
ii. They are statistically over-represented in regulatory regions.
iii. They are small, have constant size, and are repeated very often.

Identification of motives is becoming very important because they represent conserved sequences which can be biologically meaningful. Some of the areas where motive discovery can be useful include finding binding sites in amino acids, finding regulatory information within either DNA or RNA sequences, searching for splicing information, and protein domains. The motives can represent patterns which activate or inhibit the transcription process and are responsible for regulating gene expression. Motive identification can be thought of as finding the best local multiple alignments for the sequences under consideration [4].

DNA motives are often associated with structural motives found in proteins. Motives can occur on both strands of DNA. Transcription factors indeed bind directly on the double-stranded DNA. Sequences could have zero, one, or multiple copies of a motive. In addition to the common forms of DNA motives, two special types of DNA motives are recognized: palindromic motives and spaced dyad (gapped) motives. A palindromic motive is a subsequence that is exactly the same as its own reverse complement, e.g., CACGTG. A spaced dyad motive consists of two smaller conserved sites separated by a spacer (gap). The spacer occurs in the middle of the motive because the transcription factors bind as a dimer. This means that the transcription factor is made out of two subunits that have two separate contact points with the DNA sequence. The parts where the transcription factor binds to the DNA are conserved but are typically rather small (3–5 bp). These two contact points are separated by a non-conserved spacer. This spacer is mostly of fixed length but might be slightly variable.

Given a set of DNA sequences (promoter region), the motive finding problem is the task of detecting overrepresented motives as well as conserved motives from orthologous sequences that are good candidates for being transcription factor binding sites. A large number of algorithms for finding DNA motives have been developed. Most of these algorithms are designed to deduce motives by considering the regulatory region (promoter) of several co-regulated genes from a single genome. It is assumed that co-expression of genes arises mainly from transcriptional co-regulation. As co-regulated genes are known to share some similarities in their regulatory mechanism, possibly at transcriptional level, their promoter regions might contain some common motives that are binding sites for transcription factors. A sensible approach to detecting these regulatory elements is to search for statistically overrepresented motives in the promoter region of such a set of co-expressed genes. A statistically overrepresented motive means a motive that

occurs more often than one would expect by chance. Therefore, these algorithms search for overrepresented motives in this collection of promoter sequences. However, most of these motive finding algorithms have been shown to work successfully in yeast and other lower organisms, but perform significantly worse in higher organisms. To overcome this difficulty recent motive finding algorithms are taking advantages of cross-species genome comparison or phylogenetic foot printing [5]. The simple premise underlying phylogenetic foot printing is that selective pressure causes functional elements to evolve at a slower rate than non-functional sequences. This means that usually well conserved sites among a set of orthologous promoter regions are excellent candidates for functional regulatory elements or motives. Several motive finding algorithms have been developed based on phylogenetic foot printing [6].

Most recently, algorithms that integrate DNA sequence data from co-regulated genes and phylogenetic foot printing have significantly improved motive finding from genomic sequences [7]. Efforts have also focused toward developing algorithms that incorporate parameters that are useful for motive finding in higher organisms [8]. An excellent history of development and application of computer algorithms for DNA motive finding was presented in [9]. Since then a remarkably rapid development has occurred in DNA motive finding algorithms and a large number of DNA motive finding algorithms have been developed and published. In this survey, we review the recent developments in DNA motive finding algorithms.

Based on the type of DNA sequence information employed by the algorithm to deduce the motives, we classify available motive finding algorithms into three major classes:

i. those that use promoter sequences from co-regulated genes from a single genome,
ii. those that use orthologous promoter sequences of a single gene from multiple species (i.e., phylogenetic foot printing) and
iii. those that use promoter sequences of co-regulated genes as well as phylogenetic foot printing [10].

However, most of the earlier literature categorized motive finding algorithms into two major groups based on the combinatorial approach used in their design:

1. Word-based (string-based) methods that mostly rely on exhaustive enumeration, i.e., counting and comparing oligonucleotide frequencies and
2. Probabilistic sequence models where the model parameters are estimated using maximum-likelihood principle or Bayesian inference [11].

The word-based enumerative methods guarantee global optimality and they are appropriate for short motives and are therefore useful for motive finding in eukaryotic genomes where motives are generally shorter than prokaryotes. The word-based methods can also be very fast when implemented with optimized data structures such as suffix trees and are a good choice for finding totally constrained motives, i.e., all instances are identical. However, for typical transcription factor motives that often have several weakly constrained positions, word-based methods can be problematic and the result often needs to be post-processed with some clustering system [12]. Word-based methods also suffer from the problem of producing too many spurious motives. The probabilistic approach involves representation of the motive model by a position weight matrix. Position weight matrices are often visualized as a pictogram in which each position is represented by a stack of letters whose height is proportional to the information content of that position.

Probabilistic methods have the advantage of requiring few search parameters but rely on probabilistic models of the regulatory regions, which can be very sensitive with respect to small changes in the input data. Many of the algorithms developed from the probabilistic approach are designed to find longer or more general motives than are required for transcription factor binding sites. Therefore, they are more appropriate for motive finding in prokaryotes, where the motives are generally longer than eukaryotes. However, these algorithms are not guaranteed in finding globally optimal solutions, 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.

3. Methodology and System Analysis

The methodology adopted in this research was the V-model research methodology. The V-Model demonstrates the relationships between each of the DNA Motive Discovery algorithms. The horizontal and vertical axes represents accuracy, efficiency (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively. There are many applications used in motive finding, as we can't expect the patterns to be exact, different algorithms are used in finding motive.

The Algorithms for DNA motive prediction

1. Gibbs Sampling
2. The Projection Algorithm
3. Pattern Branching
4. Profile Branching

Gibbs sampling and Projection are methods that search in the space of starting positions. Pattern Branching and Profile Branching are examples of methods that search in the space of possible motives.

Gibbs Sampling

Gibbs sampling is a well-known method for finding motives in DNA sequences [13].

Given t sequences $s_1,..., s_t$, each of length n, and an integer l, the goal is to find an l-mer in each of the sequences such that the "similarity" between these l-mers is maximized.

Let $(a_1,..., a_t)$ be a list of l-mers contained in $s_1,..., s_t$. This forms a $t \times l$ alignment matrix.

Let $X(a) = (x_{ij})$ denote the corresponding $4 \times l$ profile, where x_{ij} denotes the frequency with which we observe nucleotide i at position j. Usually, we add pseudo counts to ensure that X does not contain any zeros (Laplace correction).

3.1. Gibbs Sampling Algorithm

Gibbs sampling operates as follows:

1. Randomly select an l-mer a_i, in each input sequence s_i.
2. Randomly select one input sequence s_h.
3. Build a 4 × l profile X from $a_1,\ldots, a_{h-1}, a_{h+1},\ldots, a_t$
4. Compute background frequencies Q from input sequences $s_1,\ldots, s_{h-1}, s_{h+1},\ldots, s_t$.
5. For each l-mer a $\in s_h$, compute w (a) = P(alX)/P(alQ)
6. Set a_h = a, for some a ∈sh chosen randomly with probability W (a)/∑a'€shw(a'
7. Repeat until "convergence occurs

Gibbs sampling is a well-known method that often works well in practice. However, it has difficulties finding subtle motives.

Also, its performance degrades if the input sequences are skewed, that is, if some nucleotides occur much more often than others. The algorithm may be attracted to low complexity regions like AAAAAAA....

To address this problem, the algorithm can be modified to use "relative entropies" rather than frequencies.

Another modification is the use of "phase shifts": The algorithm can get trapped in local minima that are shifted up or down a few positions from the strongest pattern. To address this, in every M^{th} iteration the algorithm tries shifting some a_i up or down a few positions.

3.2. The Projection Algorithm

The key idea of this method is to choose k of l positions at random, then to use the k selected positions of each l-mer x as a hash function h(x). With a sufficient number of l-mers hash to the same bucket, it is likely to be enriched for the planted motive

Like many probabilistic algorithms, the Projection algorithm performs a number of independent trials of a basic iteration. In each such trial, it chooses a random projection h and hashes each l-mer x in the input sequences to its bucket h(x). Any hash bucket with sufficiently many entries is explored as a source of the planted motive, using a series of refinement steps.

Random Projections

Choose k of the l positions at random, without replacement. For an l-mer x, the hash function h(x) is obtained by concatenating the selected k residues of x. Viewing x as a point in l-dimensional Hamming space, h(x) is the projection of x onto a k-dimensional subspace.

If M is the (unknown) motive, then we call the bucket with hash value h (M) the planted bucket. The key idea is that, if k < l − d, then there is a good chance that some of the t planted instances of M will be hashed to the planted bucket, namely all planted instances for which the k hash positions and d substituted positions are disjoint. So, there is a good chance that the planted bucket will be enriched for the planted motive, and will contain more entries than an average bucket. The first part of the Projection algorithm is a heuristic for finding promising sets of l-mers in the sequence. It must be followed by a refinement step that attempts to generate a motive from each of such set.

The algorithm has three main parameters:

1. The projection size k,
2. The bucket (inspection) threshold s, and
3 And the number of independent trails m.

The following, highlights how to choose each of these parameters.

Projection size: Ideally, the algorithm should hash a significant number of instances of the motive into the planted bucket, while avoiding contamination of the planted bucket by random background l-mers.

To minimize the contamination of the planted bucket, we must choose k large enough. The size of k must be such that the average bucket will contain less than 1 random l-mer.

Since we are hashing t (n − l + 1) l-mers into 4 k buckets, if we choose k such that

$$4^k > t(n - l + 1)$$

then the average bucket will contain less than one random l-mer.

Bucket threshold: In the Challenge Problem, a bucket size of s = 3 or 4 is practical, as we should not expect too many instances to hash to the same bucket in a reasonable number of trails.

If the total amount of sequence is very large, then it may be that one cannot choose k to satisfy both k < l − d and 4^k> t (n − l + 1). In this case, set k = l − d − 1, as large as possible, and set the bucket threshold s to twice the average bucket size t (n − l + 1)/4^k

Number of independent trails: We want to choose m so that the probability is at least q = 0.95 that the planted bucket contains s or more planted motive instances in at least one of the m trails. Let p(l, d, k) be the probability that a given planted motive instance hashes to the planted bucket.

Projection Algorithm

Algorithm Projection

Input: sequences $s_1,\ldots, s_t$, parameters k, s and m

Output: best guess motive for i = 1 to m do choose k different positions

$I_k \subset \{1, 2,\ldots, l\}$ for each l-mer x ∈ $s_1,\ldots, s_t$ do compute hash value $h_{Ik}(x)$

Store x in hash bucket for each bucket with ≥ s elements do refine bucket using EM algorithm return consensus pattern of best refined bucket.

3.3. Pattern Branching Algorithm

Let M be an unknown motive of length l, and let A_0 be an occurrence of M in the sample with exactly k substitutions. Given A_0, how do we determine M? Since the Hamming distance d(M, A0) = k, we have M ∈ D=k (A_0), defined as the set of patterns of distance exactly k from A_0. The idea of the Pattern Branching algorithm is to construct a path of patterns

$$A_0 \longrightarrow A_1 \longrightarrow \ldots \longrightarrow A_k,$$

in each step, moving to the "best neighbour" in D=1(A_i). The pattern A_k is scored as a guess for M. Given a pattern A of

length l, two questions must be addressed:

1. How do we score A?
2. How do we determine the "best neighbour" of A?

First, we score A using its total distance from the sample. For each sequence s_i in the sample $S = \{s_1,...,s_t\}$, let $d(A, s_i) = \min\{d(A, P) \mid P \in s_i\}$
where P denotes an l-mer contained in s_i. Then the total distance of A from the sample is

$$d(a, S) = \sum d(A, s_i)\ s_i \in S$$

Second, we define a best neighbour of A to be any pattern $B \in D=1(A)$ with lowest total distanced (B, S).

The resulting algorithm is very straight-forward:

Pattern Branching Algorithm

Input: Sequences S, motive length l, number of substitutions k

Output: best guess motive M

Init: M ← arbitrary motive pattern

for each l-mer $A_0 \in S$ do

for j ← 0 to k do

if $d(A_j, S) < d(M, S)$ then M ← A_j

$$A_j+1 \leftarrow \text{Best Neighbour } (A_j)$$

Output M

To conduct a more thorough search of D=k

(A_0), one can keep a set A of r patterns at each iteration instead of a single pattern, defining Best Neighbours (A) to be the set of r patterns $B \in D_{=1}(A)$ with lowest total distance d(B, S).

Letting $A_0 = \{A_0\}$, we thus have $|A_0| = 1$ and $|A_j| = r$ for $j > 0$

The algorithm returns the motive that has the smallest total distance to all input strings.

3.4. Profile Branching Algorithm

The Profile Branching algorithm is similar to the Pattern Branching algorithm. However, the search is in the space of motive profiles, instead of motive patterns. The algorithm is obtained from the Pattern Branching algorithm by making the following changes:

a. convert each sample string $A_0$to a profile $X(A_0)$,
b. generalize the scoring method to score profiles,
c. modify the branching method to apply to profiles, and
d. use the top-scoring profile found as a seed for the EM algorithm.

Profile of the Branching Algorithm

Input: Sequences S, motive length l, number of substitutions k

Output: best guess motive profile X

Init: X^*← arbitrary motive profile

for each l-mer $A_0 \in S$ do

$X^0 \leftarrow X(A_0$

for j ← 0 to k do

if $e(Xj, S) < e(X^*, S)$ then $X^* \leftarrow X_j$

$$X_{j+1} \leftarrow \text{Best Neighbour } (X_j)$$

Run EM algorithm with X^* as seed and return result.

This algorithm runs about 5 times slower than the Pattern Branching algorithm. The Pattern Branching algorithm clearly outperforms the Profile Branching algorithm on Challenging problems. However, pattern-based algorithms have difficulty finding motives with many degenerate positions.

Limitations of the Branching Algorithm

Recently, it has been realized that current motive discovery algorithms are far from perfect. To improve the prediction accuracy, researchers incorporated other sources of information to complement the sequence information, such as phylogenetic trees and gene expression patterns [14].

On the other hand, despite the availability of dozens of motive discovery algorithms, there are few systematic comparative benchmarking that work to independently evaluate the prediction performance of existing motive discovery algorithms [15].

One of the limitations is the inherent low signal/noise ratio in only-sequence-based motive discovery problems. Performance decreases significantly as the length of sequences increases. Several strategies have been proposed to increase the signal-to-noise ratio. An iterative refinement approach to this problem was proposed in [14]. Phylogenetic trees and structural information can be incorporated to increase signal-to-noise ratio [16].

The Gibbs Sampling can fail in two ways. The first is when there are islands of high-probability states, with no paths between them. For example, consider a probability distribution over 2-bit vectors, where the vectors (0,0) and (1,1) each have probability ½, but the other two vectors (0,1) and (1,0) have probability zero. Gibbs sampling will become trapped in one of the two high-probability vectors, and will never reach the other one. More generally, for any distribution over high-dimensional, real-valued vectors, if two particular elements of the vector are perfectly correlated (or perfectly anti-correlated), those two elements will become stuck, and Gibbs sampling will never be able to change them.

The second problem can happen even when all states have nonzero probability and there is only a single island of high-probability states. For example, consider a probability distribution over 100-bit vectors, where the all-zeros vector occurs with probability ½, and all other vectors are equally probable, and so have a probability of $½(2^{100}-1)$ each. If you want to estimate the probability of the zero vectors, it would be sufficient to take 100 or 1000 samples from the true distribution. That would very likely give an answer very close to ½. But you would probably have to take more thansamples from Gibbs sampling to get the same result. No computer could do this in a lifetime. This problem occurs no matter how long the burn-in period is. This is because in the true distribution, the zero vectors occurs half the time, and those occurrences are randomly mixed with the nonzero vectors.

4. Results and Discussion of Results

The Proposed Optimized Algorithm for DNA Motive

Discovery.

The Gibbs sampler is a technique for generating random variables from a (marginal) distribution indirectly, without having to calculate the density. Although straightforward to describe, the mechanism that drives this scheme may seem mysterious.

Suppose we are given a joint density f(x, y1,…, yp), and are interested in obtaining characteristics of the marginal density.

f(x) = ∫....∫ f(x, y1,…yp) dy1…. dyp Such as the mean or variance. Perhaps the most natural and straightforward approach would be to calculate f(x) and use it to obtain the desired characteristic. However, there are many cases where the integrations in the formular above are extremely difficult to perform, either analytically or numerically. In such cases the Gibbs sampler provides an alternative method for obtaining f(x).

Rather than compute or approximate f(x) directly, the Gibbs sampler allows us effectively to generate a sample X1,…, Xm ~ f(x). By simulating a large enough sample, the mean, variance, or any other characteristic of f(x) can be calculated to the desired degree of accuracy. It is important to realize that, in effect, the end results of any calculations, although based on simulations, are the population qualities.

Procedure of the Algorithm

1. Given n sequences, s_1, s_2,…,sn
2. Randomly initialize Position Weight Matrix (i.e. align)
3. For each sequence s_i, take it out from the PWM
   (a) score each segment of s_i with the rest of the sequences
   (b) put the sequence back

The important feature of the algorithm is convergence.

5. Conclusion

Motive Finding is the process of locating the meaningful patterns in the sequence of DNA, RNA or Proteins. A number of methods, algorithms and tools have been developed in recent years to solve these problems.

Five heuristics algorithms were mentioned and discussed with the limitations involved when each is used in motive finding. A proposed algorithm for the DNA motive discovery was selected for optimal results; this was the Gibbs Sampling algorithm.

Motive discovery in biological sequence analysis remains a challenge in computational biology. The Gibbs sampler algorithm is one of the most popular methods used in motive discovery. However, Gibbs heavily depends on initialization and suffers from local optima.

Biologists and computer scientists have been very interested in identifying computational tools for motive finding. With the advent of availability of large scale genome sequencing and high-throughput gene expression analysis techniques, a large number of motive finding tools have been designed and implemented over the past decade. This study survey shows that diverse approaches such as combinatorial enumeration, probabilistic modelling, mathematical programming, neural networks and genetic algorithms have been employed to develop motive finding tools. Earlier algorithms relied on co-expressed genes and searched for overrepresented motives. Recent algorithms take advantage of motive's overrepresentation and conservation among orthologous sequences. From this large number of available tools for motive finding, users would like to have guidance in choosing the generally best tool. However, assessment of performance of tools has still been 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. Most of the algorithms perform better in lower organisms including yeast as compared to higher organism. Recent algorithms that integrate the motive overrepresentation and cross-species conservation have proven to perform better in higher organism including human.

6. Recommendations

Considering the relevance of locating recurring patterns in the sequence of nucleotides or amino acids, motive finding is becoming popular with different algorithms created to solve the problem, but most of the algorithms suffer with local optima.

The Gibbs Sampling method is faced with a difficulty in the algorithm implementation in setting an appropriate parameter value. It is recommended that they depend on the initial choice of parameters and/or an initial set of simple motives.

References

[1] Bucher, P. (1990). Weight matrix description for four eukaryotic RNA polymerase II promoter element derived from 502 unrelated promoter sequences. J Mol Biol. 1990; 212: 563–578.

[2] Cliften, P. F., Hillier L. W., Fulton L., Graves T., Miner T., Gish W. R., Waterston R. H., and Johnston M. (2001). Surveying Saccharomyces genomes to Identify functional elements by comparative DNA sequence analysis. Genome Res. 2001; 11: 1175–1186.

[3] Liu V. S., Brutlag D. J., and Liu L. S. (2002). An algorithm for finding protein DNA binding sites with applications to Chromatin-Immunoprecipitation microarray experiments Nat. Biotechnol. 20835-839.

[4] Pradhan L., Medha K. (2008). "Motive Discovery in Biological Sequences". Master's Projects. Paper 106. http://scholarworks.sjsu.edu/etd_projects/106

[5] Pevzner P. A., Sze S.-H. (2000). Combinatorial approaches to finding subtle signals in DNA sequences, ISBM 2000, 269–278.

[6] Sandelin A, and Wasserman W. W. (2004). Constrained binding site diversity within families of transcription factors enhances pattern discovery Bio informatics J. Mol. Biol. 338207-215.

[7] Tagle D., Koop B., Goodman M., Slightom J., Hess D., and Jones R. (2003). Embryonic ε and γ globin genes of a prosimian primate (Galago crassicaudatus): nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. J Mol Biol. 1988; 203: 439–455.

[8] Reddy S. U., Arock M., and Reddy A. V. (2010). Planted (l, d) – Motive Finding using Particle Swarm Optimization, JCA Special Issue on "Evolutionary Computation for Optimization Techniques" ECOT, 2010.

[9] Van Helden J., Andre B., and Collado-Vides J. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. J Mol Biol. 1998; 281: 827–842.

[10] Casati, R., and Smith, B. (1994, February 7). Naive Physics: An Essay in Ontology. Philosophical Psychology, pp. 7-8. Retrieved April 4, 2018.

[11] Cohen, P., and Levesque, H. (1990). Intention is choice with commitment, Artificial Intelligence (Vol. 42). Retrieved March 10, 2018.

[12] Davis, E., and Morgenstern, L. (2004). Introduction: Progress in formal commonsense. New York, U. S. A: Courant Institute, New York University, New York, NY 10012, USA.

[13] Davis, S., and Lessard, A. (2018). The A. I. Revolution Begins With Augumented Intelligence. Signafire.

[14] Gilev, S., Gorban, A., and Mirkes, E. (1991). Small experts and internal conflicts in learning neural networks, 320, (pp. 220-223). Retrieved March 12, 2018.

[15] Gilev, S., Gorban, A., Kochenov, D., Mirkes, Y., Golovenkin, S., Dogadin, S., Shulman, V. (1994). MultiNeuron neural simulator and its medical applications. International Conference On Neural Information Processing, ICONIP1994, 3, (pp. 1261-1264). Retrieved March 13, 2018.

[16] Gorban, A., Mirkes, E., and Tsaregorodtsev, V. (1999). Generation of explicit knowledge from empirical data through pruning of trainable neural Networks. IJCNN'99. International Joint Conference, 6, pp. 4393-4398. Retrieved March 12, 2018.

[17] Haugeland, J. (1978). The nature and plausibility of cognitivism. Behavioural and Brain Sciences (1), (pp. 215–226). Retrieved March 13, 2018.