PDC: Pattern Discovery with Confidence in DNA Sequences

Yi Lu[1], Shiyong Lu[1], Farshad Fotouhi[1], Yan Sun[1] and Zijiang Yang[2]
Department of Computer Science, Wayne State University, Detroit, MI 48202
Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008

ABSTRACT

Pattern discovery in DNA sequences is one of the most challenging tasks in molecular biology and computer science. The main goal of pattern discovery in DNA sequences is to identify sequences of important biological function hidden in the huge amounts of genomic sequences. Several methods and techniques have been proposed and implemented in this field. However, in order to reduce computational time and complexity, most of them either focus on finding short DNA patterns or require explicit specification of pattern lengths in advance. Scientists need to find longer patterns without specifying pattern lengths in advance and still have good performance.

In this paper, we propose a pattern discovery algorithm called Pattern Discovery with Confidence (PDC). Based on biological studies, we propose a new measurement system that can identify over-represented patterns inside DNA sequences. Using this measurement, PDC algorithm can narrow the search space by checking dependency along the pattern, thus extending the pattern as long as possible without the need to restrict or specify the length of a pattern in advance. Experimental tests demonstrate that this approach can find long, interesting patterns within a reasonable computation time.

KEY WORDS
Pattern Discovery, Confidence, DNA sequence

1. Introduction

Since Watson and Crick discovered the double-helix structure of DNA 50 years ago, scientists have been studying the function of DNA at many levels. One of the most interesting challenges is to discover sequences that are similar or identical between different genomic locations or between different genomes. Similar sequences may be present because they have been conserved or selected during evolution due to some mediating important biological functions. In particular, human and rodent genomes exhibit more than 5000 “ultraconserved” sequences, most of as yet unknown function, greater than 100 bp in length [1]. Despite obvious differences between humans and mice as organisms, similarities between human and mouse genomes may indicate comparable functional requirements for regulating basic biological mechanisms and developmental stages in various tissues.

Among the important repeated sequence motifs that have been identified are cis-regulatory (CR) elements. CR elements are DNA patterns that are generally located in the upstream promoter region of a gene. They provide the binding site for transcription factors which regulate the expression of the gene. Genes are simultaneously up-regulated or down-regulated often have similar CR elements in their promoter regions. Thus, genes that share common sequences in their promoter regions are candidates for examining whether their expressions are co-regulated. Given the importance of gene expression regulation in biological function and the large amount of genomic data now becoming available, scientists have been developing various computational approaches to discovering new conserved sequence motifs, including CR elements.

Mining of sequence pattern in an informational science context has been widely studied [2, 3]. However, the problem of finding patterns in biological sequences is slightly different from the general case. The biological pattern discovery should align the pattern with each occurrence while general data mining only consider the order of items in the pattern. A large number of literatures has been published in this area, and they basically can be divided into two categories: the statistical based [4-8] and the combinatorial based [9-11].

CONSENSUS [8] compares the information content of a large number of possible binding site alignments to arrive at a matrix representation of the binding site pattern. The specificity of the protein is represented as a matrix and a consensus sequence, allowing patterns that are typical of regulatory protein-binding sites to be identified. Another widely used statistical approach is MEME [4], which is an extension of expectation maximization (EM) technique [6]. It discovers one or more motifs in a collection of DNA sequences. The algorithm estimates how many times each motif occurs in each sequence in the data sets and outputs the alignment of the occurrences of the motif. Patterns with variable-length are split by MEME into two or more motifs. The GIBBS sampler [7] stochastically examines candidate alignments in order to find the best alignment as measured by the maximum a posteriori log-likelihood ratio. This algorithm allows the simultaneous detection and optimization of multiple patterns. While MEME may get stuck into a local optima and need to ask a user to specify the minimal and maximal lengths of search motifs, both GIBBS and CONSENSUS need the input of motif length.

One of combinatorial based approaches, WEEDER [10] extends the exhaustive search mechanism, and adds the feature of error ratio threshold. Search paths can be excluded if they exceed a given error ratio. However, when WEEDER is used to find the CR sequence patterns, it may face the dilemma of choosing between a higher error ratio threshold versus missing interesting patterns. TEIRESIAS [9] extends the idea of general mining technique proposed in [2, 3] to discover protein patterns, however, in DNA pattern mining it may suffer from generating enormous candidate patterns and thus the efficiency may be affected. While the statistical approach may be affected by background noise and get stuck into a local optima, combinatorial approach may suffer from the computation cost and thus can only be applied to find short patterns.

In this paper, we propose an approach that can be used to find long patterns without specifying the length of patterns in advance and without costing too much computational time. Main contributions of this paper are:

- A new measurement system based on some biological observations. This measurement system will help the algorithm narrow the search space in which patterns will be searched.
- A new algorithm, called Pattern Discovery with Confidence (PDC). PDC is proposed to discover long patterns efficiently without specifying pattern lengths in advance.
- Experimental demonstrations of PDC algorithm performance on both simulated and real data. Furthermore, the effects of parameters on algorithm performance are examined, in order to determine the parameter values that will help discover patterns at the most competitive time cost.

This paper is organized as follows: Section 2 defines the computational problem. The algorithm is presented in section 3. Section 4 describes the results of experiments. Finally, section 5 gives the conclusion and possible future work.

2. Problem Definition and Measurement

What kind of patterns are biologists interested in? Is a pattern more interesting when it occurs more frequently? In most approaches so far, the number of occurrence is a major measurement to determine whether a pattern is interesting or not. However, sometimes this measurement is not enough to discriminate a pattern from background noise, and it may spend time checking many patterns of no biological interest. By making some biological observations, we give our measurement system in this section.

Definition 2.1 (DNA characters and sequence): Given a DNA alphabet set Σ={A,C,G,T}, A,C,G and T are called DNA characters or bases. A DNA sequence S is represented as $c_1c_2 \ldots c_n$, where $c_i \in \Sigma (1 \leq i \leq n)$. $|S|$ denotes the length of sequence S. A sequence of length n is called an n-sequence.

Definition 2.2 (DNA sequence database): A DNA sequence database D is a set of DNA sequences $\{S_1,S_2,\ldots,S_m\}$. The sum of the lengths of sequences in D is denoted as $|D| = |S_1|+|S_2|+\ldots+|S_m|$.

Definition 2.3 (Pattern): A pattern is a sequence of characters drawn from $\Sigma^*=\Sigma\cup\{N\}=\{A,C,G,T,N\}$, where N denotes the do-not-care character. We say a pattern R matches a DNA sequence S if $|R|=|S|$ and for each $i\in\{1\ldots|R|\}$, either $R_i=c_{k+i}$ $(k+i \leq |S|)$ or $R_i=N$. An occurrence of pattern R in DNA sequence S is a subsequence of S that matches R.

We need to consider the following factors to identify frequent patterns.

- Support: from a biological point of view, a group of co-regulated genes share a common DNA binding site in their regulatory regions. Therefore, a sequence that occurs more frequently has a higher probability of being associated with a co-expressed gene. Thus, the number of occurrences of a pattern is one of important parameters.

- Confidence: experimental data might contain errors, if a pattern R occurs n times in DNA sequence S, then the average number of occurrences of Rx, where x is any of the 4 DNA characters that can extend sequence R, is 0.25n. As a result, if R is frequent, then Rx has a high probability to be frequent even when x is generated randomly. To differentiate real frequent patterns from false ones, the notion of confidence is introduced.

- Significance: number of occurrences of a pattern should be significant enough to be considered as a frequent pattern. The measurement of significance is evaluated based on a ratio of the real support of a pattern *R* to the expected support of *R*.

We define these notions formally in the following.

Definition 2.4 (Support): Given a pattern *R* and a sequence *S*, the number of occurrences of *R* in *S* is called the *support* of *R* in *S*, denoted by *Supp(R, S)*. We extend this notion to a DNA sequence database $D\{S_1,S_2,...,S_m\}$, the support of *R* in *D* is defined as $Supp(R,D)=\sum_{i=1}^{m} Supp(R,Si)$.

Definition 2.5 (Confidence): Given a sequence $R=R_1R_2...R_n$ and DNA sequence database *D*, the confidence of R_1R_2 with respect to R_1 is defined as $Conf(R_1R_2,R_1)=Supp(R_1R_2,D)/Supp(R_1,D)$. And we extend the confidence *R* respect to *D* as

$$Conf(R,D)=\min\{Conf(R_1R_2,R_1),\cdots,Conf(R_1\ldots R_n,R_1\ldots R_{n-1})\}$$

ID	DNA sequence							
1	c	a	g	t	C	g	c	c
2	g	c	t	a	C	t	g	t
3	c	t	c	g	A	c	t	g

Example (Confidence): Using the sample DNA sequence database shown in the table above, character A occurs 3 times in *D*, and AC occurs 2 times in *D*, then *Conf*(AC,A)=67%.

Definition 2.6 (Pattern Probability): The Pattern Probability of *R* in sequence *S* reflects the probability that an arbitrary subsequence *XS* if *S* with length |*R*| will match *R*. Given a pattern $R=R_1R_2...R_n$ and a DNA sequence *S*, the pattern probability of *R* in *S* is defined as $\Pr(R,S)=\prod_{i=1}^{n}\Pr(R_i,S)$,where *Pr(N)=1* and *Pr*(*c*) = *Supp(c,D)* / |*D*|, for each $c\in\Sigma$. We extend the notion of pattern probability to DNA sequence database *D*. Given a pattern $R=R_1\ldots R_n$ and a DNA sequence database *D*, the pattern probability of *R* in *D* is defined as $\Pr(R,D)=\prod_{i=1}^{n}\Pr(Ri,D)$.

For example, using the same sequences in the example above the pattern probability of pattern ACTG is Pr(ACTG, *D*) = Pr(A,*D*) * Pr(C,*D*) * Pr(T,*D*) * Pr(G,*D*) = 0.125 * 0.25 * 0.25 * 0.375= $2.9*10^{-3}$.

Definition 2.7 (Expected Support):Given the statistics of character occurrences in a DNA sequence database, the expected support estimates the expected number of occurrence of a pattern *R* in *D*. Assuming |*D*|>>|*R*|, then the expected support of pattern *R* is defined as *Exp*(R,*D*)=Pr(R,*D*)*|*D*|, where Pr(R,*D*) is the pattern probability.

For example, the pattern probability of ACTG is $2.9*10^{-3}$. Then *Exp*(ACTG, *D*) = Pr(ACTG,*D*)*|*D*| = $2.9*10^{-3}$ * 24 = 0.07.

Definition 2.8 (Significance): Given a pattern *R* and a DNA sequence database *D*, the significance of pattern *R* is defined as *Sig(R,D)= Supp(R,D) /Exp(R,D).*

Example (Significance): Using the example above, if we have the support 2, then the significance is *Supp*(ACTG, *D*) /*Exp*(ACTG, *D*) = 2/0.07 = 300.

When we extend the patterns, we introduce N, the do-not-care character, into the pattern to allow some mutation or variation in the pattern. However, too much variation in the pattern will very likely disrupt the biological function and make computation more costly. Thus, it is necessary to define a maximum number of do-not-care characters that can be involved in the pattern to avoid spending too much time on non-functional patterns.

Definition 2.9 (Max Gap): Given a pattern *R*, the number of do-not-care characters, N, is called max gap of *R*, denoted as *DNC(R)*.

Then our problem can defined as following based on the notion introduced above: Given a set of DNA sequence database *D*, minimum support threshold *ST*, minimum confidence threshold *CT*, minimum significance threshold *FT*, and max gap threshold *GT*, find all patterns *R* that

- *Supp(R, D) ≥ ST*
- *Conf(R, D) ≥ CT*
- *Sig(R, D) ≥ FT*
- *DNC(R) ≤ GT*

3. PDC algorithm

The starting point of the PDC algorithm is to identify the core patterns which are composed of a list of consecutive characters without any do-not-care characters. The idea is inspired from the observation of biology nature. Most *cis*-Regulatory Elements (CREs) have consensus core sequences. The overall pattern of such kind of CREs includes gap between conserved base pair. This property enables us to develop a solution which start from an *l*-sequence instead of 1-sequence (*l*≥1), and then extend to the longer pattern. The algorithm will only find patterns that have at least one core pattern which has length more than *l*. However, we can set *l* to a small value which will not affect the result a lot. Here, we set *l* to 4.

A useful property to use in this process is the idea that (*n+1*)-pattern $R_1...R_{n+1}$ is a solution if and only if *n*-pattern $R_1R_2...R_n$ is a solution. However, exhaustive search is time consuming and can only be used to find short patterns. By using confidence, support and max gap threshold in our measurement system, we can eliminate the patterns that are unlikely to be extended as patterns, and thus gain performance efficiency.

The Significance parameter in the measurement system can be used to rank patterns after they have been identified. Patterns with higher Significance are likely to have greater biological relevance.

```
Algorithm PDC
Input: database $D$, support threshold $ST$, confidence threshold $CT$, significance threshold $FT$, and max gap threshold $GT$
Output: All patterns R where Supp(R,D) $\geq$ ST, $Conf\ C(R,D) \geq CT$, $Sig(R,D) \geq FT$, $DNC(R) \leq GT$
Begin
  Initialization
  Generate all core $l$-patterns, push into stack s
  While (!s.isEmpty) Do
      Pop up candidate pattern $R$ from stack s
      $R_I$=$R$+ "A"/ "C"/ "G"/ "T"/ "N"
      For i = 1 to 5 Do
          If IsExtendable ($R_i$) Then
              s.push ($R_i$)
          Else
              If IsSolution($R_i$) Then
                  Output the pattern $R_i$
          End If
      End For
  End While
End
```

Figure 1 Pseudo-code of PDC algorithm

Figure 1 shows our PDC algorithm. The algorithm starts with initialization. Basically it scans the database and finds all occurrences for each 4-sequence, and keep a record of each occurrence as a pair (Sequence ID, Location in Sequence). At the same time, it calculates the character probabilities for A,C,G and T. Calculation of character probabilities for A,C,G and T is straightforward. It can be obtained by adding all occurrences of A,C,G,T, and dividing by the total number of characters in the database. In order to keep all occurrences of 256 possible 4-sequences from AAAA to TTTT, an array of 256 elements will be used to store the occurrences.

In each element of the array, we keep 6 parameters:

- pattern: a list of DNA characters, may include do-not-care character N for n-pattern (n>4).
- Support: the support of current pattern in D, as introduced in definition 2.4.
- PP: pattern probability of the current pattern.
- Significance: significance of the current pattern.
- LocList: a linked list, in which each location is recorded as a pair *(i,j),* while i is the sequence ID, and j is the location number in sequence S_i.
- gap: the total do-not-care characters in the pattern, at the beginning, it is 0 for core 4-sequences.

The core and costly step in PDC algorithm is the pattern extension step. The pattern searching process is just like a tree expanding process. After the initialization step, we have already reached the 4th level, and the next step would be to extend to level 5 patterns with 5 characters each.

The extension could extend each possible 5-way path with possible 5 characters, A, C, T, G and N. The 5-way extension would generate an exponentially increasing number of candidate patterns, making the search space extremely large and the computation extremely costly. To avoid checking every candidate pattern that can be generated by this 5-way extension, The function *IsExtendable* in the algorithm checks whether a candidate pattern can be pruned or not based on three thresholds introduced in section 2. By setting a confidence threshold, we can eliminate most of the extension paths. At the same time, the support and gap thresholds will eliminate most extension paths with little likelihood of yielding interesting patterns. If the candidate pattern can not be extended, then it will be checked whether it meets the minimum significance threshold defined in section 2. All patterns with higher significance may be of greatest interest. The significance threshold is used to filter out patterns with small Significance value. Patterns are therefore sorted and output based on the significance value.

4. Experimental Results

Our experiments were conducted on a Dell Dimension 8200 PC machine with 2.4G Hz CPU and 512MB RAM. The algorithm is implemented using Java SDE 1.4 and tested with Windows XP operating system. Three categories of experiments were conducted. First, the effectiveness of the algorithm was evaluated. Second, the algorithm efficiency was examined. Finally, the significance ranking system was examined.

4.1 Effectiveness Study

Two experiments were conducted in this study. First, we simulated a DNA sequence database by randomly generating DNA sequences composed of A,C,G and T. Each sequence data set had m sequences with equal length of n base pairs. Simulated genomic sequences contain a highly-represented pattern. In this case, we use perfect estrogen response element (pERE: GGTCANNNTGACC) as the pattern. The pERE was implanted randomly into the simulated random DNA sequences. We controlled the number of pEREs implanted to test the sensitivity of the algorithm. The lowest number of pERE in our test set was 10 in 20 DNA sequences with length of 1000 bp. We set the minimum support to 3, the max gap to 10, minimum confidence ranges from 0.1 to 0.5. The pERE is found in all data sets with highest Significance value.

The second experiment was conducted on real data sets that are collections of sequences obtained from biological experiments. The data contains upstream regulatory regions of 277 genes known to be responsive to estrogen in gene expression microarrays. Three data sets were used, each of them composed of 277 sequences with lengths of 1000, 2000, and 5000 bases respectively. The minimum support was set to 10; max gap was set to 10; and minimum confidence was set to 0.3. Several significant pERE and longer patterns were found during the experiments, one of them as long as 80 bp. While some of these patterns had previously been noted in the biological literature, others may be novel and should be carefully examined by biologists.

4.2 Performance Study

Three experiments have been conducted to check the impact of three thresholds defined in section 2.

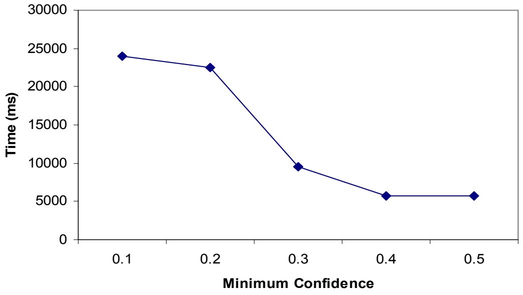

Figure 2 impact of Minimum Confidence threshold

1. Minimum Confidence threshold. The impact of changing minimum confidence threshold was studied on randomly chosen data set of 100 sequences/1000 bases. The minimum support is set to 10 and max gap set to 10. The experiment studied the impacts of varying minimum confidence from 0.1 to 0.5. Figure 2, showing the algorithm performance with different values for minimum confidence, illustrates a non-linear effect. The greatest change along the confidence axis is from 0.2 to 0.3, above which not much improvement results from further changes in the minimum confidence threshold. In most cases, the characters are distributed evenly. That means the A, C, G and T occur almost at the same ratio in the data set as the frequency of each character, which is approximately 25%. Thus, if the minimum confidence is set to 0.3, most random patterns will be filtered out, and real patterns, occurring more frequently than 25% of the time will survive and be extended.

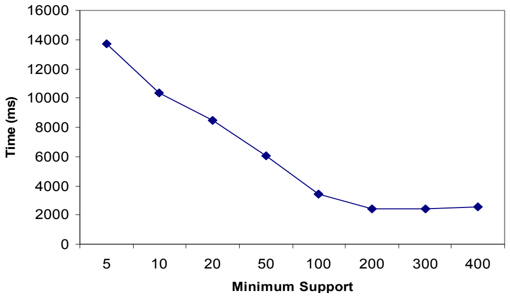

Figure 3 impact of Minimum Support threshold

2. Minimum Support. The impact of the minimum support was similarly examined in a randomly chosen test data set having 100 sequences each 1000 bases in length. The minimum confidence was set to 0.3 and the max gap was set to 10. We studied the effect of changing minimum support from 5 to 400. As shown in Figure 3, the running time decreases as minimum support is increased. In the minimum support range of 5 to 200, the relationship between running time and minimum support is approximately linear. However, for values of minimum support greater than 200, there was no influence on the running time. This is because most of time, when the minimum support is greater than 200, the extension procedure of the algorithm will not be running.

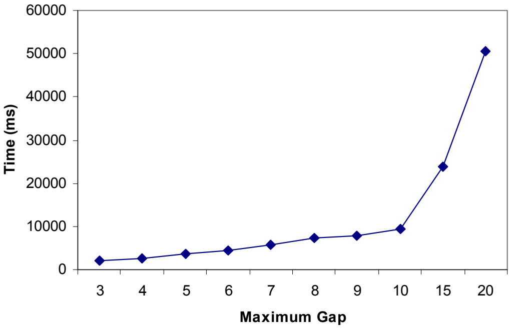

Figure 2 impact of Max Gap threshold

3. Max Gap. Finally, the effect of max gap was examined. Using again the 100 sequence/1000 bases data set combinations, the minimum confidence set to 0.3, and minimum support fixed at 10, the impact of max gap in the range of 3 to 20 was investigated. As shown in Figure 4, when the max gap becomes larger, the running time increase accordingly. Obviously, the algorithm needs to check more possible patterns when the max gap is larger.

In summary, these 3 experimental results demonstrate the effects of minimum support, max gap and minimum confidence levels on the performance of the PDC algorithm. While minimum support has linear effects on the running time, max gap and minimum confidence thresholds have non-linear impacts. Thus, defining appropriate thresholds for these thresholds may improve performance. We conclude from the experimental study that our measurement system has the potential to improve the pattern discovery process. The confidence threshold particularly serves an important role in improving performance, giving three- to five- fold faster searches when set to 0.3 or greater.

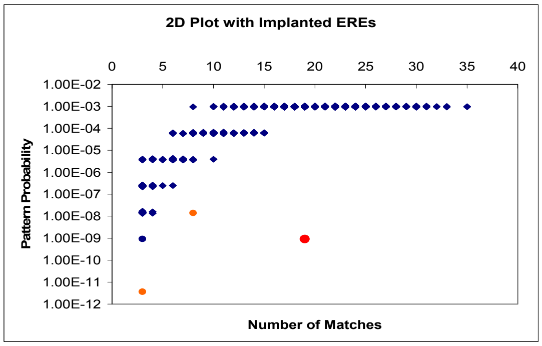

Figure 5 the Significance Ranking System

4.3 Significance Ranking System

To show the contribution of significance ranking system, the same data set used in section 4.2 is used here. In Figure 5, we demonstrate that our significance ranking system can discriminate the true pattern from the background sequences. We plot the result of random DNA sequences with implanted pERE sequences. Three significant patterns have been identified:

1: GGTCAnnnTGACC(support:19);

2: GTCAGNNTGACCANNNNNNT(support:3);

3: AGGTCNNNNNGACC(support:8).

These three patterns are all derived from the original pERE sequence. Pattern 1 indicates the exact pERE pattern implanted. Pattern 2 and 3 are part of the conserved sequence and the number of patterns found was only a subset of original patterns. Three round points with shallow color plotted in Figure 5 represents the sequences introduced above. We can separate them from other points even with visually observation. Using the significance ranking system introduced in section 2, the program can works decently to separate the real patterns and noise.

5 Conclusions and Future Work

In this paper, we proposed a new pattern discovery algorithm called PDC and a new measurement system to discover conserved patterns that might have biological function in DNA sequences. PDC algorithm inherits the advantage of combinatorial approach and thus can find patterns without inference of background noise in sequence database. Compared with the TEIRESIAS [9], our approach is more efficient in finding DNA sequence pattern by providing the pruning techniques. It has been confirmed by our experiments that our measurement system could find interesting patterns within a reasonable computation cost.

The significance measurement is a naive definition in this paper and is mainly based on character probability. It does not consider the impact of the order or position of the characters in the pattern. For future work, we may consider developing new measurement parameters that can better describe the sequences in a biological context. In addition, we would like to compare our work with the current most advanced algorithms on efficiency and effectiveness.

Reference

1. Bejerano, G., et al., *Ultraconserved elements in the human genome.* Science, 2004. 304(5675): p. 1321-5.
2. Agrawal, R. and R. Srikant. *Mining Sequential Patterns*. in *Proceeding of the 11th International Conference on Data Engineering*. 1995. Taiwan.
3. Srikant, R. and R. Agrawal. *Mining Sequential Patterns: Generalizations and Performance Improvements*. in *Fifth International Conference On Extending Database Technology (EDBT)*. 1996. Avignon, France.
4. Bailey, T.L. and C. Elkan, *Unsupervised learning of multiple motifs in biopolymers using EM.* Machine Learn, 1995. 21: p. 51-80.
5. Buhler, J. and M. Tompa, *Finding motifs using random projections.* J Comput Biol, 2002. 9(2): p. 225-42.
6. Lawrence, C.E. and A.A. Reilly, *An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences.* Proteins, 1990. 7(1): p. 41-51.
7. Lawrence, C.E., et al., *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment.* Science, 1993. 262(5131): p. 208-14.
8. Hertz, G.Z. and G.D. Stormo, *Identifying DNA and protein patterns with statistically significant alignments of multiple sequences.* Bioinformatics, 1999. 15(7-8): p. 563-77.
9. Rigoutsos, I. and A. Floratos, *Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm.* Bioinformatics, 1998. 14(1): p. 55-67.
10. Pavesi, G., G. Mauri, and G. Pesole, *An algorithm for finding signals of unknown length in DNA sequences.* Bioinformatics, 2001. 17(Suppl 1): p. S207-14.
11. Marsan, L. and M.F. Sagot, *Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification.* J Comput Biol, 2000. 7(3-4): p. 345-62.