

A HYBRID ALGORITHM FOR MOTIF DISCOVERY FROM DNA SEQUENCES

EDWARD WIJAYA, KANAGASABAI RAJARAMAN, MANISHA BRAHMACHARY,
VLADIMIR B. BAJIC * AND SUNG SAM YUAN †

We propose a new hybrid algorithm that combines the strengths of probabilistic and deterministic approaches, for motif discovery from DNA sequences. This algorithm consists of a sequence filtering component that uses a probabilistic strategy, and a graph-theoretic motif finding component that utilizes a deterministic algorithm. We show that the algorithm can correctly find the target motif(s) with high probability. We perform experiments on synthetic and real biological datasets and observe that the algorithm offers a promising method for motif discovery from biological sequences.

1. Introduction

One of the major challenges that biologists are facing today is elucidation of the mechanisms that govern the regulation of gene expression. Unravelling the regulatory regions of potentially co-regulated genes and regulatory motifs in them makes an important step in this process. Motif discovery from DNA sequences is a very active research topic and there is a large body of work done in developing motif discovery algorithms (See, for example [2,3]). These algorithms generally fall into two classes, namely deterministic and probabilistic. Deterministic algorithms are based on enumeration strategies for finding the motifs. Some of the well known deterministic algorithms include TEIRESIAS [10], WINNOWER [9], MULTIPROFILER [7], and the recently proposed Constraint Based Method [5]. On the other hand, probabilistic algorithms employ a probabilistic search to find the target motifs. Examples of this class include GibbsDNA [8, 14], MEME [1], Random Projection [4], and Dragon Motif Builder [15]. Each of these approaches has its own strengths and weaknesses and none of the methods is efficient with large datasets, while the quality of the detected motifs may vary from case to case.

In this paper, we take a hybrid approach that aims to incorporate the strengths of both the deterministic and probabilistic approaches. Our idea is to develop an algorithm that consists of a probabilistic component to handle the variabilities of biological data, and a deterministic component to ensure that close to globally optimal motifs are found. We consider the problem of discovering motifs from DNA sequences where only a fraction of the sequences may have motif instances. This is a more generalized problem than the one addressed by graph-theoretic methods such as WINNOWER and Constraint-Based Method which require all the sequences have at least one motif instance. We propose

*Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613. Email for correspondence: kanagasa@i2r.a-star.edu.sg

†School of Computing, National University of Singapore, Singapore 117543.

a hybrid algorithm that consists of deterministic and probabilistic components, to solve this problem. We perform experiments on synthetic datasets, as well as a collection of 27 promoters of β -defensin genes and their mouse-rat orthologs. We were able to show that the hybrid approach enables the algorithm to detect highly homogeneous sets of target motifs.

This paper is organized as follows. Section 2 provides the problem statement. In section 3, we present our hybrid motif discovery algorithm and describe the probabilistic and deterministic components. Section 4 describes the experiments done on synthetic and real biological datasets and presents the performance results of the algorithm. In section 5, we provide a discussion of the related work and conclude the paper.

2. Problem Statement

Let $S = \{s_1, \dots, s_T\}$ be a given set of DNA sequences. Let M be a motif (unknown) of length l . M may occur in a sequence as mutated instance with up to d substitutions. Let m_1, m_2, \dots, m_{T^*} , be the instances of M . The (l, d) motif finding problem is to find M when $T^* \leq T$.

We assume $T^* \geq 3$, to avoid finding trivial motifs.

Note that typically several target (l, d) motifs may exist, depending on T^* . In general, a computational algorithm cannot determine a priori the biological significance of a motif with certainty. Hence, the aim to find all these motifs.

3. Hybrid Motif Discovery(HMD) Algorithm

Our HMD algorithm uses a sequence filtering idea. Each sequence is viewed as either ‘planted’ or ‘corrupted’. A planted sequence is one that contains an instance of a motif. All other sequences are called corrupted. The algorithm consists of two components: *sequence filtering* and *motif finding from planted sequences*. Sequence filtering is implemented using a technique known as *locality sensitive hashing*. For motif finding, we employ the Constraint Based method (henceforth called, CMF algorithm). We describe the two components in detail below.

3.1. Sequence Filtering

Sequence filtering is implemented through two steps: *locality sensitive hashing* and *sequence scoring*.

Locality Sensitive Hashing Locality sensitive hashing (LSH) is a technique proposed by Indyk and Motwani [6] in the context of computational geometry. LSH is based on the idea that simple hashing functions can be used to map objects in a multidimensional space to buckets that have high probability of containing objects close to each other than those which are far apart. Buhler and Tompa [4] have exploited this idea in their Random Projection algorithm for obtaining good starting motif models that are later refined to identify the target motifs. Our approach is different in that we use LSH only to filter out corrupted sequences rather than motif discovery. The motif finding is handled by the CMF algorithm.

Let T sequences each of length N be given, and M be an (l, d) motif. Define a hash function $h(x)$ that hashes an l -mer to a k -mer by choosing k positions at random. If $k \leq l - d$ and k is not too small, then it is more likely that the motif instances will be hashed to the same bucket than random l -mers, because they must agree in all k chosen positions. This implies that instances from implanted sequences tend to co-occur than random l -mers.

Sequence Scoring Consider three sequences S_1, S_2 and S_3 of which S_1 and S_2 are planted with a motif but S_3 is not. Suppose that l -mers from these sequences are hashed using $h(x)$. By LSH property, l -mers from S_1 & S_2 are more probable to co-occur in a bucket than those from S_2 & S_3 or S_1 & S_3 . This observation enables us to identify the planted sequences. We perform hashing over many independent trials and compute co-occurrence statistics from the buckets to deduce the planted sequences with high probability. The complete sequence filtering algorithm is given in Algorithm 1.

In Steps 2-9, LSH is applied on the l -mers extracted from the sequences and the buckets collected. The buckets are accumulated to form a hashtable in Steps 10-14. This process is iterated over m iterations, in the for loop spanning Steps 1-15. The for loop in Steps 16-18 computes a count of the sequence pairs in each bucket. Step 19 sorts the sequences using the occurrence frequency and returns the top T sequences in Step 20.

The choice of k and s (the bucket threshold) is crucial. k is chosen large enough such that $4^k \gg t(n - l + 1)$, but smaller than $l - 2d$. s is chosen small enough so that enough buckets are considered and true motif instances are not missed. We choose m empirically to get a stable ranking in Step 19.

To find the target motifs, we pass the filtered sequences to the CMF algorithm, described next.

3.2. Constraint Based Motif Finding (CMF) Algorithm

The CMF algorithm is a novel combinatorial motif discovery method that is based on the concept of *constraint rules*. The constraint rules are defined through *center strings*. Given l -mers x and y , let $dist(x, y)$ define the hamming distance between x and y . Consider a set of l -mers $S = \{s_1, s_2, \dots, s_n\}$. Center c of S is defined as any l -mer that satisfies the inequality $dist(c, s_i) \leq d, \forall i = 1, \dots, n$. Note that centers are candidates for the target motif(s). The idea of the CR algorithm is to first find centers from the given sequences and then verify if they are true motifs.

3.2.1. Constraint Rules

A graph-theoretic method is used to analyze the given sequences (described below) and find the centers. This is the critical and combinatorially expensive step. Constraint rules are procedures employed to make this step efficient. These procedures do not need the verification of the hamming distance inequality but instead can derive the centers directly by observing that the sequence set have patterns under some special cases. The exact rules to handle these special cases and their correctness are provided in [5]. To handle all other cases, an algorithm called *Constraint Mechanism* is employed. Due to space limitations,

Algorithm 1 Sequence Filtering Algorithm**procedure** *SeqFilt*($l - mers, k, s, m, N, T'$)

- 1: **for** m iterations **do**
- 2: pick k indices i_1, \dots, i_k from $\{1, \dots, L\}$
- 3: generate LSH hash function $f(s) = (Si_1, \dots, Si_k)$
- 4: **for** each l-mer **do**
- 5: **for** $j = 1; j \leq k; j++$ **do**
- 6: k-mer[j] = l-mer[Si_j]
- 7: **end for**
- 8: hash l-mers into buckets such that
all l-mers in a bucket correspond to same k-mer
- 9: **end for**
- 10: **for** each bucket **do**
- 11: **if** the class size is greater than certain threshold s **then**
- 12: output all buckets, l-mers, seqIndex to a hashtable
- 13: **end if**
- 14: **end for**
- 15: **end for**
- 16: **for** each bucket **do**
- 17: count pairs(seqIndex-1, seqIndex-2) found in the bucket
- 18: **end for**
- 19: Sort each sequence in the top- N pairs based on number of occurrences
- 20: Return the top T' sequences

we omit this algorithm and refer the reader to [5].

3.2.2. Graph-Theoretic Motif Finding Algorithm

Let $I = \{I_1, \dots, I_t\}$ be the set of implanted motif instances. The motif discovery problem can be simplified as finding a subset of I which can be converted into centers (using constraint rules or constraint mechanism). This is achieved by the use of clique finding techniques. First of all we construct a graph $G(S, l, d)$, where $S = \{s_1, \dots, s_t\}$ is the given set of t sequences, l is the motif length, and d is the maximum number of mismatches/mutations. For every position p in the sequence s_i , construct a vertex $s_{i,p}$ representing the length- l substring starting at position p in s_i . Connect vertex $s_{i,p}$ with $s_{j,q}$ by an edge if $i \neq j$ and the Hamming distance between them does not exceed $2d$.

Suppose that every sequence in S contains a motif instance of M . Let $p = \{p_1, \dots, p_t\}$ be the set of positions where p_i is the position of the motif instance in sequence s_i . Let $V = \{s_{1p_1}, s_{2p_2}, \dots, s_{tp_t}\}$ be the subset of vertices $G(S, l, d)$ representing the t motif instances. Every pair of vertices in V should have an edge in $G(S, l, d)$ graph, therefore the set V should correspond to a clique of size t in $G(S, l, d)$. In contrast to other similar algorithms, this method does not aim to find large cliques. It only suffices to find a *convertible clique*

that has 3 vertices and meets requirements of a constraint rule (or constraint mechanism). The whole procedure is illustrated in Algorithm 2.

Algorithm CMF begins with finding a set Q consisting of all the 2-cliques (cliques of size 2) in the first two sequences s_1 and s_2 . Then, we iteratively enlarge the cliques in Q with the remaining sequences in S . For the enlarged cliques which are convertible, we immediately convert them into centers. The remaining enlarged cliques are inserted into Q . The process is iterated until Q is empty or size of cliques in Q reach certain limit f . If some f -cliques are still remained in Q after the whole process, we compute the centers for these f -cliques by constraint mechanism. Note that, we set a clique size limit to avoid the huge extension cost of large cliques. The center testing is performed through an algorithm that makes use of the Pigeon Hole principle [12].

Algorithm 2 Constraint Based Method (sketch)

```

1:  $Q \leftarrow \emptyset$  and  $C \leftarrow \emptyset$ ;
2: for each  $s_{1i}$  and  $s_{2j}$  ( $1 \leq i, j \leq n - l$ ) do
3:   if  $dist(s_{1i}, s_{2j}) \leq 2d$  then
4:     check if  $\{s_{1i}, s_{2j}\}$  is 2-clique in  $G(S, l, d)$  and Insert  $\{s_{1i}, s_{2j}\}$  to  $Q$ ;
5:   end if
6: end for
7: for each  $m = 3 \dots f$  ( $f$  is clique size limit) do
8:   for each  $(m - 1)$ -clique  $cli$  in  $Q$  and each vertex  $s_{mj}$  do
9:     Remove  $cli$  from  $Q$ ;
10:    if  $cli = cli \cup \{s_{mj}\}$  is a  $m$ -clique in  $G(S, l, d)$  then
11:      if  $cli$  is a convertible clique then
12:        Convert  $cli$  into centers, and insert these centers into the set  $C$ ;
13:      else
14:        Insert  $cli$  into  $Q$ ;
15:      end if
16:    end if
17:  end for
18: end for
19: for each remaining  $f$ -clique  $cli$  in  $Q$  do
20:   Convert  $cli$  using Constraint Mechanism and insert derived centers into the set  $C$ ;
21: end for
22: for each center  $c$  in the set  $C$  do
23:   Call CenterTest to verify if it is an actual motif.
24: end for

```

Theorem 3.1. *Let M be a motif that has at least one instance in every sequence in S . Then, Algorithm CMF finds M .*

The theorem follows from the results of [5].

The LSH property enables that sequence filtering can identify implanted sequences with high probability, provided the number of iterations is large enough. Together with Theorem 3.1, this implies that HMD algorithm can find the target motif(s) even if only a subset of the sequences may be implanted. Noting that this result is only partially proven, we conduct experiments to verify it empirically.

3.2.3. Motif Scoring

Typically, even after center testing (Step 23 in Algorithm 2), there may be a lot of motifs returned. We apply a scoring function to rank the motifs and retain only those above a threshold. The information content of M is used as the scoring function. Suppose a motif s of length k has approximate occurrences in a subset of S input sequence. The information content of this motif is defined to be:

$$IC(M) = \sum_{j=1}^k \sum_{c \in \Sigma} p_{c,j} \log_2 \frac{p_{c,j}}{b_c}$$

where $p_{c,j}$ is the probability with which base c occurs in position j among the motif occurrences in S , and b_c is the background frequency of the character c in the DNA sequences. The background distribution is assumed to be uniform. Information content provides a measure of how well a motif is conserved and how likely a motif is with respect to the background distribution.

4. Experiments

4.1. Synthetic Data

We consider a (15,2)-motif problem where sequences are of length $n = 600$. The sequences are generated uniformly at random and implanted with chosen (15,2) motif. We consider two cases, a) $T^*=15$ and $T=20$, b) $T^*=10$ and $T=20$. For each case, we generate five different datasets and compute the average performance.

The sequence filtering component and the complete HMD algorithm are evaluated separately.

4.1.1. Evaluation of Sequence Filtering

The ranked sequences output by sequence filtering are evaluated using precision and recall. Let *true positives (TP)* be the number of returned sequences that are implanted and *false positives (FP)* be the number of returned sequences that not implanted. Then, precision = $TP/(TP+FP)$ and recall = TP/T^* . The results are presented as P-R curves in Figure 1.

We observe that our method achieved perfect performance for Case *a*. In Case *b*, it performed at over 95%.

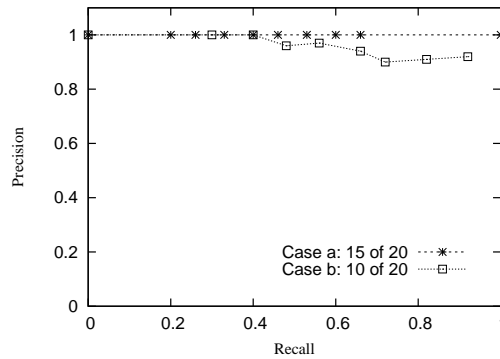


Figure 1. Performance of Sequence Filtering.

4.1.2. Evaluation of HMD Algorithm

Here we evaluate the motif discovering performance of the HMD algorithm. We consider two tasks: 1) finding motif(s) when l and d are known, and 2) when l and d are not known.

Task 1 - (l, d) known In this task, the HMD algorithm returns (15,2) motifs. The results are provided in Table 1 and 2. We notice that HMD correctly found the correct motif in both cases. In the Case b , the precision was affected because the algorithm returned noisy motifs. In a later investigation, we found that the noisy motifs had more than 9 positions common to the correct motif.

Table 1. Performance of HMD algorithm for $l = 15$ and $d = 2$.

Datasets	k	s	m	T'	Avg No. of Motifs	P	R	Time (min)
Case a	9	4	200	7	1	1.00	1.00	8.0
Case b	9	4	500	5	3	0.54	1.00	16.9

Task 2 - (l, d) unknown The previous task assumed that l and d are known. However in practical scenarios, they are not known in general. Hence, we attempt to set l and d to values different from the true values and evaluate how the HMD algorithm performs. Specifically, we try to find a (9,1)-motif. We restricted the problem to Case a and compared the performance with that of the Random Projection (RP) algorithm.

For the (15,2) motif finding problem, there exist up to 7 correct (9,1)-motifs. For example, given the actual implanted motif is *ATGTCTCTCTATACT*, the derived motifs with length 9 would be: *ATGTCTCTC*, *TGTCTCTCT*, *GTCTCTCTA*, *TCTCTCTAT*, *CTCTCTATA*, *TCTCTATAC*, and *CTCTATACT*. The performance is evaluated for the ability to identify all these motifs.

We compute the average precision and recall over 5 datasets and present the results in

Table 1.

Table 2. Comparison of Performance in finding a (9,1)-motif for a (15,2)-motif problem.

Algorithm	k	s	m	T'	Avg No. of Motifs	P	R	F meas.	Time(min)
RP	7	4	3	15	1	1.00	0.14	0.25	0.17
HMD	6	8	200	5	4	0.29	0.69	0.41	1.26
HMD	6	8	500	7	3	0.43	0.37	0.40	3.37

We observed that HMD outperformed RP in terms of F measure. Random Projection always identified only 1 motif *ATGTCTCTC*, and hence had 100% precision. However, it could not find the other 6 motifs. HMD was able to find them at close to 70% recall. HMD's precision was lower because it found many noisy motifs, which we later found were indeed very close to one of the correct motifs. By increasing T' to 7, we see that the number of noisy motifs can be reduced and precision improved, but at a loss in recall. Our contention is that, though precision is important, recall is critical because the biologist cannot afford to miss out motifs that may actually be biologically significant.

4.2. Biological Data

Here we applied our HMD algorithm on two real biological datasets; yeast-associated protein (YAP), and β -defensin Promoters.

4.2.1. Yeast-Associated Protein (YAP)

Buhler & Tompa [4] and Dong, et.al. [5] considered this dataset along with three other biological datasets namely: preproinsulin, dihydrofolate reductase (DHFR), and gametic lethal (GAL). Of these biological datasets we consider only YAP since others are small in size. YAP consists of 16 sequences with 800bp long each and has a published motif *TTACTAA* [13].

We attempt to find a (9,1)-motif from this dataset. Parameters we used for Sequence Filtering are: $k = 5$, $s = 4$, and $m = 100$. With $T' = 5$, the HMD algorithm was able to find the motif *tTTACTAAg* which contains the published motif. The algorithm took 25 seconds.

4.2.2. β -defensin Promoters

We have selected promoters of 27 human β -defensin genes and their mouse and rat orthologs. These promoters cover the region of [-1000, +500] relative to the transcription start site. We evaluate our HMD algorithm in finding three different types of (l,d)-motif. The results are summarized in Table 3.

Our results show that HMD algorithm can find motifs with high information content, for various (l, d) . For comparison, we applied RP algorithm on the same dataset.

Table 3. Motif Discovery by HMD from β -defensin Promoters.

Motif	k	s	m	No. of Motifs	IC	Time (min)
(9,1)	6	9	100	35	17.08	4.4
(12,1)	7	5	100	19	22.16	9.2
(15,2)	7	5	100	7	26.76	12.0

For a (9,1) problem, it returned *CCAGGTAAA* as the consensus. HMD algorithm indeed found this motif which had IC=15.2. In fact, it also found similar motifs *cctcCAGGT*, *ctcCAGGTt*, *aactCAGGT*, *acaCAGGTa*, and *ccCAGGTaa*, from which we are able to deduce that *CAGGT* is a highly conserved pattern for this dataset. We also analyzed the motifs found by the Dragon Motif Builder (DragonMB) [http://research.i2r.a-star.edu.sg/DRAGON/MOTIF_SEARCH]. After choosing threshold=0.85, average IC=1.5, and no. of motifs=10, DragonMB found motifs *CAGGatcaa* (IC=13.2) and *aaaaCAGGT* (IC=11.9) which contained similar patterns. The motifs found by DragonMB sometimes had more than 15 instances. HMD, though returned higher IC motifs, was able to identify motifs having upto 5 instances only.

5. Discussion and Conclusion

This paper has proposed a new hybrid algorithm for motif discovery from DNA sequences. It was motivated by the idea of combining the strengths of probabilistic and deterministic approaches. The algorithm consisted of a sequence filtering component (that was based a probabilistic strategy) and a graph-theoretic motif finding component (that utilized a deterministic algorithm). Through experimental studies, we observed that the hybrid approach enables the algorithm effectively find target motif(s) even though only a fraction of the sequences may be implanted.

Our work generalizes the work reported in [5]. As noted in Section 3.2.2, the CMF algorithm is not guaranteed to find the correct motif even if one of the sequences is not implanted. For such cases, they proposed a strategy of choosing 3 sequences iteratively at random to increase the chances of finding a motif. This requires C_3^T runs of the CMF algorithm and hence potentially inefficient. Also, this strategy may miss out some important motifs. In contrast, our sequence filtering approach can provide a ranked list of implanted sequences directly and these sequences will contain all the important motifs with high probability (provided LSH is run enough number of times). Our method is also more efficient because the LSH runs are much simpler than running CMF iteratively.

However, both the approaches have the limitation of resulting in motifs derived from small number of instances (usually 3-5). This is because the CMF approach does not look for maximal cliques as, for example, done by WINNOWER [9]. This limitation can be addressed by generalizing the constraint rules and constraint mechanism to handle more than 3 sequences. We are currently investigating this direction. In addition, we have not considered domains where the motifs may include spacers [11] and motif instances contain insertions and deletions. Extending our work to handle these domains will form part of our

10

future work.

6. Acknowledgements

We would like to thank Dong Xiaolan for his insights, and Jeremy Buhler for providing the Random Projection implementation.

References

1. T. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21:51–80, 1995.
2. A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5:279–305, 1998.
3. B. Brejova, C. Patten, C. Dimarco, G. Holguin, S.R. Hidalgo, and T. Vinar. Finding patterns in biological sequences. Technical Report CS-2000-22, University of Waterloo, December 2000.
4. J. Buhler and M. Tompa. Finding motifs using random projections. In *Proceedings of the Fifth Annual International Conference on Research in Computational Molecular Biology*, RECOMB-01, pages 69–76, Montreal, Canada, April 2001.
5. X. Dong, S.Y. Sung, W.K. Sung, and C.L. Tan. Constrained based method for finding motif in DNA sequences. In *Proc. of IEEE 4th Symp. on Bioinformatics and Bioengineering (BIBE)*, 2004.
6. P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In ACM, editor, *Proceedings of the thirtieth annual ACM Symposium on Theory of Computing: Dallas, Texas, May 23–26, 1998*, pages 604–613, New York, NY, USA, 1998. ACM Press.
7. U. Keich and P. Pevzner. Finding motifs in the twilight zone. In *Proceedings of the Sixth Annual International Conference on Research in Computational Molecular Biology*, RECOMB, pages 195–204, 2002.
8. C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald, and J. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.
9. P. Pevzner and S. H. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 269–278, 2000.
10. I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences. *Bioinformatics*, 14:55–67, 1998.
11. E. Rocke and M. Tompa. An algorithm for finding novel gapped motifs in DNA sequences. In Sorin Istrail, Pavel Pevzner, and Michael Waterman, editors, *Proceedings of the 2nd Annual International Conference on Computational Molecular Biology (RECOMB-98)*, pages 228–233, New York, 1998. ACM Press.
12. W. K. Sung and W. H. Lee. Fast and accurate probe selection algorithm for large genomes. In *Proceedings of IEEE Computer Society Bioinformatics Conference (CSB)*, Stanford, CA, 2003.
13. J. van Helden, B. Andre, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281(5):827–842, 1998.
14. X. Wu, J. Cheng, C. Song, and B. Wang. A combined model and a varied gibbs sampling algorithm used for motif discovery. In Yi-Ping Phoebe Chen, editor, *2nd Asia-Pacific Bioinformatics Conference*, volume 29 of *CRPIT*, pages 99–104, Dunedin, New Zealand, 2004. ACS.
15. L. Yang, E. Huang, and V.B. Bajic. Some implementation issues of heuristic methods for motif extraction from dna sequences. *Int.J.Comp.Syst.Signals (To Appear)*, 2004.