The Evolutionary Computation Approach to Motif Discovery in Biological Sequences [*]

Michael A. Lones and Andy M. Tyrrell
Intelligent Systems Research Group
Dept. Electronics, University of York
Heslington, York YO10 5DD, UK
mal503@ohm.york.ac.uk / amt@ohm.york.ac.uk

August 26, 2005

Abstract

Finding motifs — patterns of conserved residues — within nucleotide and protein sequences is a key part of understanding function and regulation within biological systems. This paper presents a review of current approaches to motif discovery, both evolutionary computation based and otherwise, and a speculative look at the advantages of the evolutionary computation approach and where it might lead us in the future. Particular attention is given to the problem of characterising regulatory DNA motifs and the value of expressive representations for providing accurate classification.

1 Introduction

A motif, in the context of biological sequence analysis, is a consensus pattern of DNA bases or amino acids which accurately captures a conserved feature common to a group of DNA or protein sequences. DNA motifs are sometimes termed signals: examples are regulatory sequences, scaffold attachment sites, and messenger RNA splice sites. Examples of protein motifs, which are also known as fingerprints, include enzyme active sites, structural domains, and cellular localisation tags. Motif discovery is the act of identifying and characterising motifs, and underlies a number of important biomedical activities. For example: the identification of regulatory signals has applications for gene finding in sequenced genomes, understanding of regulatory networks, and the design of drugs for regulating specific genes; and protein motifs are routinely used to identify the function of newly-sequenced genes and to understand the basis of a protein's cellular function.

Evolutionary computation (EC) has certain advantages for motif discovery. Unlike many algorithms used in bioinformatics, evolutionary algorithms (EA) carry out global search and have relatively low sensitivity to initial conditions. Evolutionary algorithms are comparatively flexible in terms of how solutions are represented and evaluated and do not require knowledge about the problem to which they are being applied. Since motif discovery is typically an off-line activity, the relative low speed of evolutionary algorithms when compared to other bioinformatics algorithms is not a significant issue.

The remainder of this paper is divided into three sections. Section 2 presents the biological basis of motifs, describing the appearance and function of both DNA and protein motifs. Section 3 reviews existing approaches to motif discovery, discussing the advantages and disadvantages of conventional approaches, and summarising the various evolutionary computation approaches. Section 4 hosts a discussion of the relative advantages of evolutionary computation for motif discovery, highlights the issues of representation and evolvability, and suggests future avenues of research for EC-based approaches. Most of the material presented in this paper should be accessible to readers with a limited biological background.

[*]

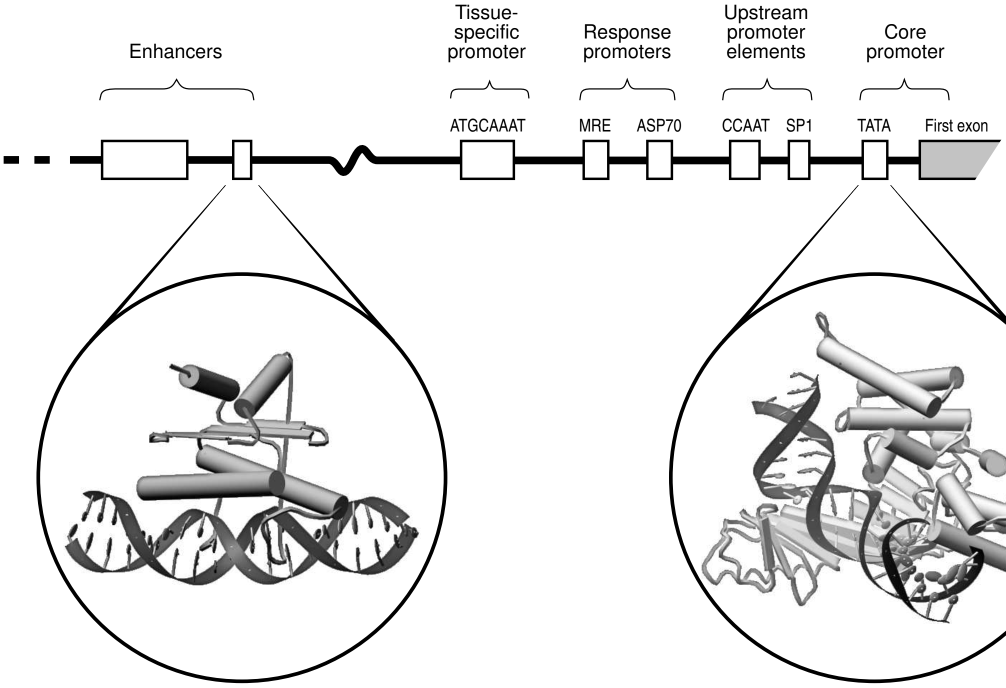

Figure 1: Typical organisation of eukaryotic promoter elements within the DNA sequence upstream of the coding region of a gene. Inset molecular visualisations show [left] a DNA-bending transcription factor bound to an enhancer (PDB ID: 1C7U [46]), and [right] a transcription factor complex bound to the TATA promoter element (PDB ID: 1D3U [35]).

2 The Biological Basis of Motifs

This discussion is limited to two kinds of biological sequence data: nucleotide sequences and protein sequences. A nucleotide sequence is a string of letters (A,C,G and T) representing the sequence of nucleotide bases (Adenine, Cytosine, Guanine and Tyrosine) present within DNA and RNA molecules. A protein sequence is a string of letters (A–Z, excluding B, J, O, U, X and Z, which represent ambiguity groups) representing the linear sequence of amino acids from which a protein is constructed.

A sequence motif can be seen as a pattern, present within one or more biological sequences, which alludes to the presence of a particular biological characteristic. In order to correctly characterise motifs, it is important to take into account their biological meaning. This section gives some biological background to the kind of motifs which occur within nucleotide and protein sequences.

2.1 Motifs in Nucleotide Sequences

In addition to the coding regions of genes, genomes contain a host of other signals which determine interactions between DNA, RNA transcripts, and the cellular machinery. Examples are:

Regulatory signals which determine interactions with the transcriptional machinery and therefore how and when genes are expressed.

Splicing signals which determine interactions with RNA editing proteins and thus serve to delineate the coding and non-coding components of genes.

Localisation signals which determine interactions of messenger RNA transcripts with the cellular localisation machinery and consequently the location of mature proteins within cells.

Cell cycle signals which determine interactions with the cell cycle machinery, facilitating DNA replication and recombination during cell division.

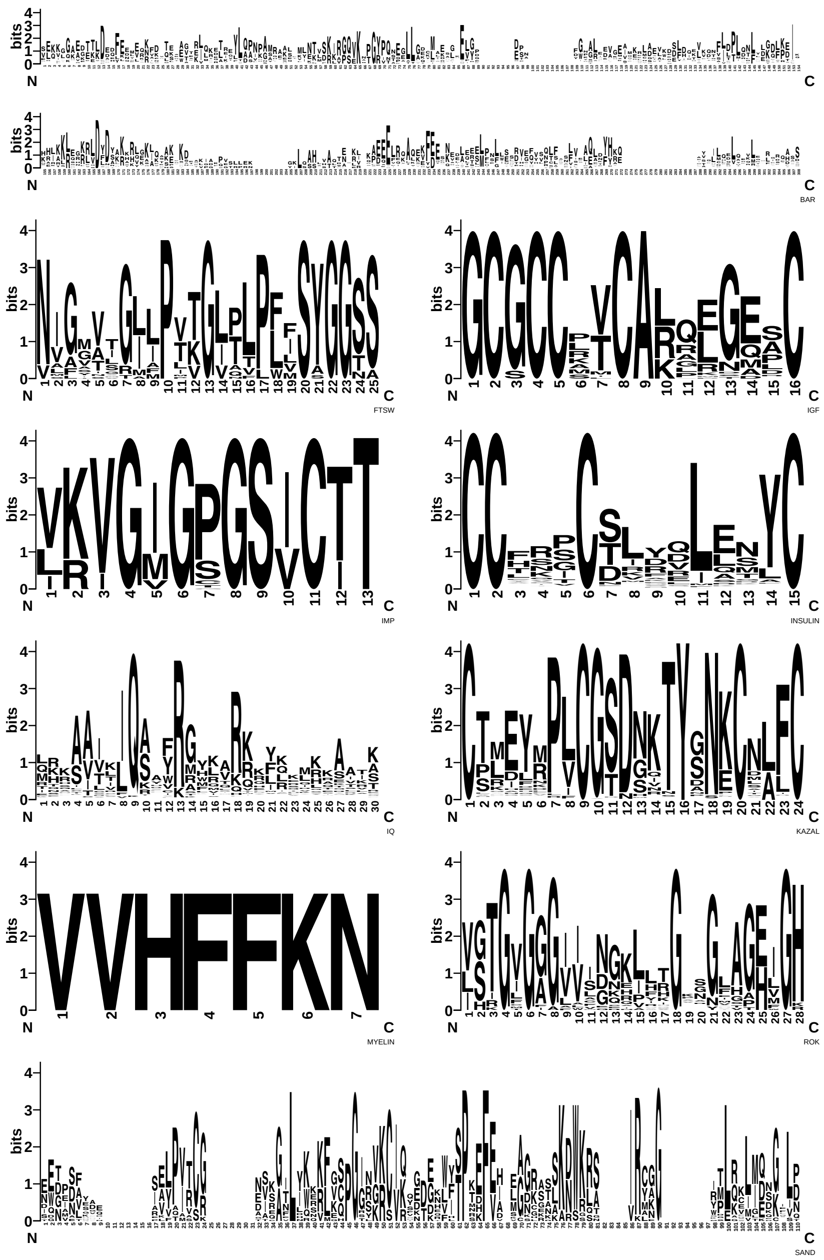

Figure 2: Sequence logos for SwissProt entries matching PROSITE motifs, indicating the relative probability of particular amino acid residues occurring at each position in a sequence. Positions with taller letter stacks are more significant. Left to right then top to bottom: BAR (PROSITE ID PS51021, covers two lines), FTSW_RODA_SPOVE (PS00428), IGF_BINDING (PS00222), IMP_DH_GMP_RED (PS00487), INSULIN (PS00262), IQ (PS50096), KAZAL (PS00282), MYELIN_MBP (PS00569), ROK (PS01125) and SAND (PS50864).

In each of these cases, the corresponding motifs typically describe protein binding sites present within DNA (or RNA) sequences. A better understanding of DNA motifs can be gained by looking at an example. Figure 1 presents a conceptual diagram of the regulatory elements upstream of a gene's coding region. In order for a gene to be transcribed, it is necessary for the RNA polymerase enzyme to bind to the start of the gene. In eukaryotes, there is little direct binding affinity between the gene start site and RNA polymerase. Consequently, transcription requires the presence of 'helper' proteins which stabilise the interaction between RNA polymerase and the DNA molecule, forming a transcription complex. These *transcription factors* function either by directly stabilising the interaction between DNA and RNA polymerase or indirectly by, for instance, stabilising the interaction between DNA and another transcription factor. This is achieved by transcription factors either acting as 'glue' — binding one thing to another by binding to both of them — or by changing the shape of other elements of the transcription complex, making them fit together better.

Many of these transcription factors bind to the DNA sequence upstream of a gene's start site. As figure 1 illustrates, this behaviour is reflected by a pattern of DNA regulatory elements upstream of the first exon. Broadly speaking, there are two kinds of regulatory element: promoters and enhancers. *Promoters* are located within the region up to about 300 base pairs upstream of the first exon, providing binding sites to transcription factors which have fairly direct roles in forming the transcription complex. *Enhancers*, by comparison, can be found at distances of many kilobases upstream or downstream of the first exon, and serve to 'enhance' the binding of transcription factors to promoter elements. Promoters can be divided into several categories. The core promoter is the site where RNA polymerase initially binds to the DNA, and covers the area just upstream of and a few bases into the first exon. The core promoter often (but not always) includes an element known as the TATA box. The core promoter is necessary, but not normally sufficient, to support transcription. Upstream promoter elements are usually found just upstream of the TATA box. These bind transcription factors which are key to the formation of a stable transcription complex, and are usually required for efficient transcription to take place. The CCATT box and the Sp1 box (which has consensus sequence 'GGGCGG') are common examples of upstream promoter elements. Response promoters and tissue-specific promoters are also found upstream of the TATA box. Both bind transcription factors which mitigate gene expression depending upon the current state of the cell or its environment. Examples of response promoters are HSP70, which causes expression when heat shock signals are present in a cell and MRE (metal response element) which induces expression in the presence of heavy metals. An example of a tissue-specific promoter is the sequence 'ATGCAAAT', which promotes the expression of immunoglobin genes in B cells.

Enhancers are generally harder to characterise than promoters, not least due to the magnitude and variance of their distance from the core promoter. Enhancers can appear as one or more copies of a particular sequence on either DNA strand and often this sequence resembles that of the promoter elements with which they interact. Given their distal location, it is not always obvious how enhancers interact with the transcription complex. However, many enhancers are known to bind factors which bend DNA, forming large loops which bring enhancer-bound factors into association with promoter-bound factors. Enhancers can be general or specific, enhancing promoter activity in all or only certain cell types. Enhancers may also act to repress transcription, by binding disruptive transcription factors. These are known as silencers.

Whether or not, and how much, a gene is expressed can be the result of interactions between a wide range of transcription factors. Each regulatory element contributes either positively or negatively to expression and groups of regulatory elements carry out analogues of Boolean functions. For these reasons, gene regulatory regions have been compared to both logic circuits and McCulloch-Pitts neurons, e.g. [38]. Whilst understanding of regulatory motifs is important for predicting when genes will be expressed and consequently what they might do, regulation is not determined by these factors alone. Other factors which need to be taken into account include: the expression levels of transcription factors, which requires understanding of genome-wide regulatory networks or access to transcriptome data; the presence of transcription factors which do not bind to DNA but which effect whether or not a transcription complex can be assembled; the presence of insulators (DNA motifs which limit the action of enhancers to certain regions of DNA); chromatin state, the presence of chromatin remodelling factors, and histone acetylation — all of which affect the accessibility of DNA to transcription factors; and post-translational localisation and modification of messenger RNA transcripts and proteins [15]. For more information about regulatory motifs see, for instance, [43, 54, 69].

2.2 Motifs in Protein Sequences

A protein motif is usually considered to be a pattern of amino acid residues which are conserved between members of a protein family. The presence or absence of a motif can be used to determine whether a particular protein belongs to a family and therefore determine whether or not it is likely to carry out a particular function. Like DNA motifs, protein motifs often correspond to binding sites, since these tend to be well-conserved regions of protein structure. Protein motifs may also correspond to structural domains, especially in protein families with a structural rather than enzymatic role. Many proteins belong to multiple protein families and hence contain sequences matching multiple motifs.

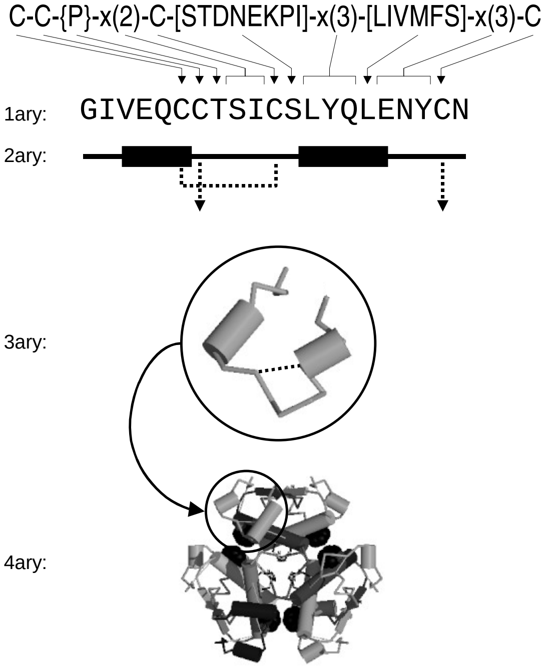

Figure 3: Correspondence between the insulin PROSITE motif and the structure of the human insulin hexamer molecule. Dashed lines indicate disulfide bonds between cysteine residues.

There are numerous databases of protein motifs, e.g. Blocks [30], PFam [6], PRINTS [2], ProDom [11], PROSITE [22], and the integrated resource InterPro [48]. A selection of motifs from PROSITE are depicted in Figure 2 as sequence logos. These represent motifs from a diverse range of protein families and show how protein motifs vary widely in terms of both length and conservation. Figure 3 illustrates how the insulin PROSITE motif — a regular expression — captures structural features of insulin, including the cysteine residues which form key disulfide bonds holding together both the tertiary and quaternary structure of the protein. This also highlights the fact that protein motifs attempt to capture facets of three dimensional structure within a one dimensional sequence. For more information about protein structure and composition see, for instance, [45].

3 Approaches to Motif Discovery

The most common approach to locating and characterising conserved regions in groups of biological sequences is multiple sequence alignment. A multiple sequence alignment, or MSA, is a group of sequences vertically aligned such that regions of similarity in each of the sequences occur in the same columns of the alignment. MSAs are typically found using dynamic programming approaches. Dynamic programming is a divide-and-conquer algorithm which can efficiently find the optimal alignment of two sequences. For larger numbers of sequences, dynamic programming can not be used directly to find optimal alignments, since the computational complexity of the approach rises exponentially in respect to the number of sequences in the alignment. Instead, alignments are built up in a piece-wise fashion from alignments between smaller groups of sequences. Clustering techniques are used to determine the order in which sequences should be aligned, based upon the heuristic that similar sequences should be aligned before less-similar sequences (e.g. [62]). This process does not guarantee optimal alignments, but is widely used by biologists to determine sequence homology. The advantage of MSAs for motif discovery is that they do not lose any of the information contained in each sequence. However, they have several disadvantages: they do not generalise the sequence data, and therefore are of limited help for biological understanding; they can not be directly used to classify other sequences; and they can be very large.

A more common way of representing motifs is to use consensus sequences. ‘TATAAT’, for instance, captures the most likely form of the bacterial version of the TATA box [55]. However, many regulatory motifs are not well conserved and can not be adequately described using a deterministic expression such

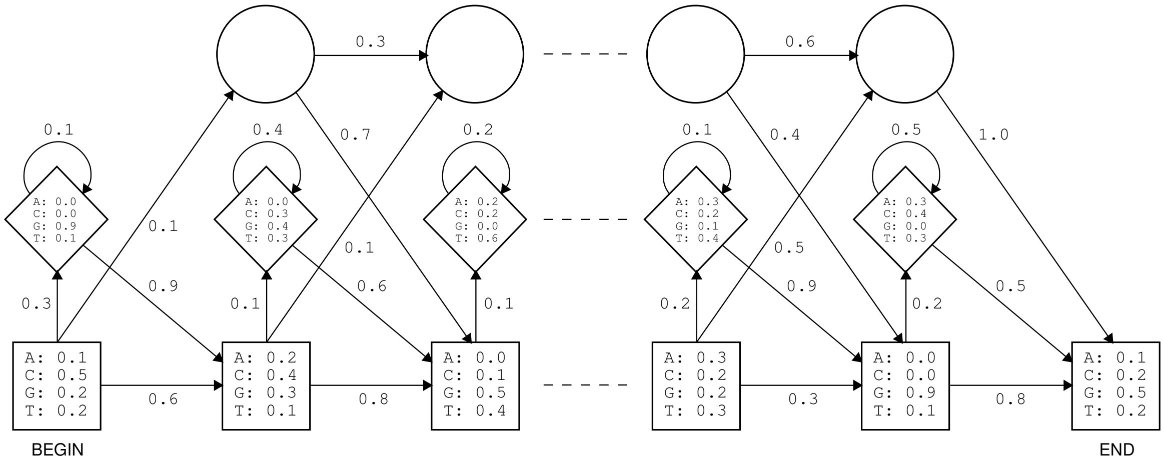

Figure 4: A profile hidden Markov model. Box states represent conserved sites. Diamond-shaped and circular states represent insertions and deletions, respectively. Numerical values indicate symbol emission and state transition probabilities.

as this. Consequently, regulatory motifs are normally described using probabilistic expressions such as weight matrices which give the relative likelihood of each symbol appearing at each position in a sequence. Consensus sequences, whether deterministic or probabilistic, have the advantage that they can easily be used to classify new sequences by measuring the distance from a consensus sequence. For deterministic expressions, this distance is usually calculated using a substitution matrix which captures the relative likelihood of one symbol mutating into another during the course of evolution. Consensus sequences are commonly derived using statistical techniques (see [61] for a review). An example is Expectation-Maximisation (EM), a hill-climbing algorithm which iteratively derives a weight matrix using measures of information content and entropy. Whilst efficient, this approach does not guarantee optimality and is sensitive to initial parameter settings; although improved performance can be achieved through the use of Gibbs sampling. A recent statistical approach to regulatory motif discovery is described in [4].

More generally, motifs can be described using regular expressions. Rather than describing a single consensus sequence, these are able to capture a set of related sequences. For instance, the expression 'C-C-{P}-x(2)-C-[STDNEKPI]-x(3)-[LIVMFS]-x(3)-C' (the insulin motif from Figure 3) captures all amino acid sequences beginning with two cysteine residues, followed by a residue that isn't proline, two arbitrary residues, another cysteine residue, a residue from a specified set, three arbitrary residues, another residue from a set, three more arbitrary residues, and a final cysteine residue. There are various kinds of regular expression. Each of these has the same expressive power, but differences in syntax lead to differences in how easily certain patterns can be expressed. Sometimes it is also desirable to use more constrained forms of expression: for instance, only allowing expressions of the form 'R-x(n_0, n_1)-R-x(n_0, n_1)-. . . ' i.e. a simple consensus sequence with optional variable-sized gaps between each pair of specified residues. A classification of regular expression languages for biosequences can be found in [10]. All regular expressions can be represented by equivalent finite state automata, which can be used to efficiently search for matches within sequences. Regular expressions can be derived using either a sequence-led or a pattern-led approach. Sequence-led approaches attempt to generalise patterns from sequence data, whereas pattern-led approaches generate-and-test all patterns up to a certain length, checking for correspondence with the data. The latter approach is obviously intractable for motifs with long defining lengths.

For motifs with low levels of conservation, probabilistic regular expressions are more appropriate. Hidden Markov Models (HMMs) have proved particularly useful for describing conserved sequences, being both flexible and better able to identify distant homologues than other approaches [21]. Figure 4 illustrates a profile HMM, the form of HMM which is used most often to represent conserved sequences. This model, which has a fixed architecture, improves upon weight matrix expressions by explicitly identifying the likelihood of insertions and deletions within a conserved sequence. The emission and transition probabilities of HMMs are typically learnt from sequence data using the Baum-Welch algorithm, a variant of dynamic programming. Whilst efficient, this procedure is sensitive to initial parameter values and does not guarantee an optimal model of the sequence data. See [23] for an example of an HMM-based approach to regulatory motif discovery.

Neural networks have also been used to recognise and classify patterns in biological sequences [8]. The benefits of a neural approach are best demonstrated by their application to protein secondary-structure prediction, where neural architectures are able to give a prediction accuracy of around 80% from the amino acid sequence data alone [5]. However, neural networks pose two significant problems when applied to motif discovery: (i) their relative inflexibility when mapping sequence data to inputs; and (ii) the difficulty of interpreting neural models, particularly when hidden layers are present within the network. The first of these problems can be somewhat overcome through the use of hybrid approaches; for instance, placing an HMM between the sequence data and the neural inputs [31]. For general overviews of motif discovery approaches see, for example, [10, 67, 70].

3.1 Evolutionary Computation Approaches

There have been numerous applications of evolutionary computation to consensus biosequence discovery. Most of these have been concerned with multiple sequence alignments [1, 12, 13, 17, 18, 25, 27, 32, 36, 37, 42, 44, 50–53, 57–59, 64, 66, 68, 73], although a number have targeted other representations [20, 28, 29, 33, 34, 39, 40, 56, 60, 63, 65, 71, 72]. Most MSA approaches can be divided into two classes: those which directly evolve alignments e.g. [52], and those which evolve the order in which sequences will be aligned using conventional techniques e.g. [44, 58]. Exceptions are [64], where an EA is used to post-process alignments generated by the popular alignment program CLUSTAL, and [53], in which an EA is used to generalise patterns extracted from MSAs. EA-based approaches to MSA have been shown to outperform conventional techniques in terms of alignment quality, though usually with a significant speed disadvantage. Recent reviews can be found in [59] and [18].

Of more relevance to this review are EC approaches in which structures more general than MSAs have been evolved. In [20], a GA is used to evolve consensus sequence strings and weight matrices, with promising results. Regular expressions have been evolved by both Hu [34] and Heddad et al. [29] to describe protein motifs, producing results competitive with conventional approaches, and in [56], the author describes how grammatical GP can be used to evolve probabilistic regular expressions which can effectively describe PROSITE motifs. Particle swarm optimisation (which is in many ways similar to EC) has also been used to design regular expressions, improving upon standard PROSITE motifs [14]. A number of studies have looked at how EAs may be used in the design of HMMs. In [72], a GA is used to evolve the topology and initial parameters of HMMs, which are then trained using Baum-Welch. This approach was shown to generate reliable models of the CCAAT and TATA boxes and other simple motifs. Thomsen [63] and Won et al. [71] have used similar approaches for other biosequence applications. EAs have also been used to evolve HMMs for non-biological sequence classification problems [16, 41, 60]. In [16], a GA is used to optimise HMM parameter values, and in [41] a GA is used to optimise both topology and parameter values — in both cases leading to more accurate models than those produced using Baum-Welch.

EC approaches have also been used to evolve a number of unconventional structures for representing and recognising sequence motifs. Howard and Benson [33] describe a classification architecture which they call GP-automata, consisting of a finite state automaton in which each state has an associated regular expression parse tree and each transition has a distance measure and a Boolean decision function. This architecture is designed to recognise regulatory sequences: using the distance measure to guide the movement of a reading head along the sequence, the regular expressions to recognise individual promoter motifs, and the Boolean functions to determine whether there is a suitable combination of promoters within the sequence to lead to gene expression. No direct comparison has yet been made against other promoter finding algorithms. A number of other researchers have also looked at how programmatic classifiers may be used for motif discovery [28, 39, 40, 65]. Handley [28] has evolved GP expressions consisting of continuous numerical functions and left/right relative movement commands to recognise promoter regions in the *E. coli* genome. This approach produced results competitive with contemporary approaches without (unlike the other approaches) requiring biological knowledge. However, the evolved expressions are not easy to understand. Koza et al. [39,40] have also used GP to evolve biological sequence classifiers, producing competitive results. An alternative approach has been demonstrated by Vallejo [65], who successfully evolved Turing machines to classify HIV sequences.

4 Discussion

It is clear from the literature referenced above that evolutionary computation techniques can be successfully applied to motif discovery. However, EC approaches remain a niche activity within this field, with practicing biologists preferring to use tried-and-tested approaches based around dynamic programming, statistical techniques and neural networks. Some of the reasons for this are quite evident. It is only recently that the EC community has begun to focus on biological applications, with special sessions and workshops becoming common-place at the larger conferences. Certainly much of the output in the area of motif discovery has been fragmentary, resulting from short-term projects and not well advertised in the biological community. For instance, most of the EC publications cited in this paper were published in evolutionary computation and artificial intelligence conference proceedings and journals: only a handful are available in publications accessible to biologists. There is also a lack of bioinformatics software (with a few exceptions) based around EC techniques. Relatively few biologists are programmers and many bioinformaticians are from a biological rather than computational background. Hence, most practitioners are unlikely to be aware of — let alone willing to implement — computational methods which are not currently in the bioinformatics mainstream.

4.1 Why use EC for Motif Discovery?

Whilst EC is certainly not a panacea, EC approaches do have particular advantages for motif discovery. One of the foremost criticisms of machine learning approaches to motif discovery is that often they do not produce biologically meaningful results. One reason for this is that many conventional bioinformatics algorithms (e.g. Baum-Welch and EM) carry out local search and thus tend to converge to solutions which are locally but not globally optimal. This is not only a problem for classification accuracy, but also for biological meaning — since many sub-optima will not reflect the biological truth more often associated with the global optima. EC approaches, by comparison, carry out a global search of the solution landscape; and do this without resorting to specific heuristics (which may skew the results) or exhaustive search (which, like dynamic programming, scale poorly). Whilst this does not guarantee optimal solutions, it does increase the likelihood of finding them.

EC approaches have two more advantages for generating biologically meaningful solutions: flexibility of scoring, and flexibility of representation. For conventional bioinformatics algorithms, there is often a close linkage between the way in which solutions are derived and the way in which they are scored; meaning that solutions can only be evaluated in regard to, for example, sum-of-distance between sequences. For EC, there is no such linkage: solutions can be evaluated by arbitrary fitness functions, and in principle these could take into account any information, biological or non-biological, functional or non-functional.

4.2 Representation

However, it is representational flexibility which is perhaps the most interesting quality of EC in respect to motif discovery. Motif representation is a key issue for any approach to motif discovery: since if a motif can not be accurately represented then it can not be learnt, regardless of the learning algorithm which is used. As Section 2 of this paper attempts to convey, motifs are not just arbitrary patterns within biological sequences. Motifs represent functional or structural characteristics of biological molecules. Where a motif represents a binding site, it must implicitly capture three-dimensional information: for instance, in the case of protein binding sites, the motif must capture the chemical and spatial interaction between the amino acid sequence, its ligand(s), and any other — possibly distant — parts of the amino acid sequence which are involved in the interaction. For regulatory DNA sequences, motifs must describe a one-dimensional image of the three-dimensional transcription complex assembled around the DNA molecule. A complete description of a gene's regulatory region must capture both the chemical bonding between the DNA sequence and the transcription factors and the interactions between transcription factors.

Many models of motifs used in standard approaches to motif discovery are fairly simplistic. There has been an understandable tendency to consider biological sequences as mere strings of letters, to be processed using standard computer science string processing algorithms. This has an obvious advantage: it encourages the use of efficient procedures such as dynamic programming and regular expression matching. Even the more biology-motivated representations, such as profile HMMs, still remain within the universe of regular expressions and dynamic programming-like algorithms. This status quo is due, at least in part, to the dependence of certain algorithms upon certain types of representation, and the difficulty of designing efficient algorithms for more expressive forms of representation. Nevertheless, regular expressions represent only a small part of the universe of languages with which classifiers can be expressed. Regular expressions have important limitations — notably their inability to describe long-range relationships within sequences — which limits the kind of biological patterns which they are able to accurately describe.

With evolutionary computation, there is no requirement for a certain form of representation. The mutate-and-test (or recombine-and-test) mechanism of EAs mean that, at least in principle, any solution representation can be used so long as it can be represented within the memory of a computer. Despite the plethora of MSA and regular expression-centric approaches, EC methods have already shown the potential benefits of non-standard forms of motif representation. Howard and Benson's GP-automata approach [33], for example, demonstrates a novel way of representing regulatory motifs and their interactions. However, more interesting are those approaches which break free of the regular expression mould, notably the use of Turing-complete classifiers by Koza et al. [39] and Vallejo [65]. In this context, a Turing-complete classifier can be interpreted as a programmatic model of the biological interactions which lead to a particular motif or set of motifs; so, in essence, the GP approach is analogous to reverse-engineering the algorithm underlying the biological system. Unlike simple representations such as regular expressions, Turing-complete classifiers could explicitly capture the constraints of chemical bonding, long-range interactions between parts of an amino acid chain, and 'hidden' variables such as direct interactions between the transcription factors in a transcription complex. Of course, it is not a simple matter to induce complex algorithms with GP. Other problems include choosing an appropriate function set, understanding the evolved algorithms, and whether or not it is possible to improve evolved algorithms with respect to newly acquired data.

4.3 Evolvability

Whilst there is no explicit requirement for a certain form of representation in EC, all representations are not equally appropriate for evolutionary search. Ideally the solution representation should be evolvable: it should engender "the ability to reach 'good' solutions via evolution" [49]. From a representation viewpoint, evolvability is a set of characteristics which encourage better solutions to be explored as a consequence of random changes enacted by variation operators. One of these characteristics is solution locality: that small changes made to a solution's representation should generally lead to small changes in the solution it represents. If this is not the case, then mutations are most likely to lead to unrelated solutions — most of which will have relatively low fitness, assuming that the solution being mutated has a high fitness relative to the search space average.

Unfortunately many types of solution representation have poor solution locality, especially when crossover operators are used. Ideally crossover should produce new, fitter, solutions by recombining the components of existing fit solutions. In practice this rarely happens because most solutions are not

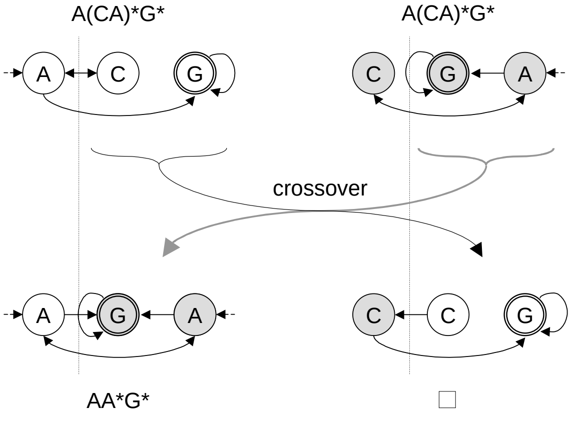

Figure 5: One-point crossover between equivalent finite state automata. Initial states are indicated by unconnected arrows and final states by double circles. The second child solution carries out no useful behaviour since it has no initial state.

compatible at the representation level. Figure 5 shows a typical example of this problem. Here, the same regular expression is represented by two finite state automata: each containing the same states, but in a different order. Because the states are in a different order, their contexts are not preserved when the two finite state automata are recombined using one-point crossover, resulting in child solutions with behaviour considerably different to their parents. Clearly if two equivalent solutions can not be meaningfully recombined, then it is unlikely that crossover will lead to meaningful behaviour in the situation where parents do not represent the same solution. This kind of problem is found in all representations where a component's context is determined by its position (such as sequences, matrices and programs), and is particularly a problem for variable-length representations [47].

A number of strategies have been looked at for improving the evolvability of solution representations in EC. Many of these are motivated by the organisation of biological genomes, which Conrad [19] has described as having three characteristics which reflect a balance between the need for phenotypic stability on the one hand and the pressure towards genetic exploration on the other: redundancy, compartmentalisation, and weak linkage. Of these, redundancy is the characteristic most often introduced to EC representations to promote evolvability (see [47] for a review). A number of authors have also tried to improve context preservation by removing positional-dependence from solution representations, e.g. [3,24,47]. Most of these ideas are applicable to the kind of representations commonly used to represent sequence motifs.

4.4 Future Directions

Existing approaches to motif discovery have made use of a fairly narrow selection of motif representations; a situation in large part due to limitations of 'traditional' approaches to sequence analysis. Considering its flexibility with regard to how solutions are represented and evaluated, EC appears to offer the opportunity to explore motif representations other than the standard sequence alignments, weight matrices, regular expressions and hidden Markov models of the bioinformatics world. Existing work in the area suggests the benefits of this approach, but there remains a lot of work still to be done. The main issues which need to be addressed include accuracy, interpretability, and evolvability.

The accuracy of a classifier depends upon its ability to capture all the constraints upon all appropriate variables (such as chemical bonds). For biological classification, the appropriate variables are often not known in advance, so it is desirable that the classifier can learn new variables as well as learning their constraints. EAs using expressive representations such as Turing-complete classifiers are able to build models of unknown variables so long as the necessary information is available. However, this is by no means a simple task and it would seem beneficial to introduce more knowledge into the process to help identification of appropriate variables. In GP, biological knowledge could be introduced via extra functions and terminals: for example, a function which uses an external program to calculate secondary structure for a region of a protein, removing the need for GP programs to rediscover the mapping from sequence to secondary structure (which would be a major task in itself). This mechanism could also be used to introduce non-sequence data such as X-ray models of protein tertiary structure, known post-translational modifications, metabolomics and transcriptome data, and details of epigenetic factors, such as local chromatin structure arrangement, for DNA sequences. In this way, EC approaches could offer a framework with which to relate sequence data to the increasing amounts of post-genomic data (data to which GP has already been applied [26]).

However, there is also scope for more disciplined representations. Where a biological model — or aspects of the model — is fairly well understood, it would seem more appropriate to represent the model explicitly rather than rely on evolution to find its gross structure. This could certainly help make evolved solutions more interpretable, and would still allow room for evolutionary exploration within the bounds of the model. A representation for genetic regulatory regions, for example, could represent explicit promoter and enhancer regions, their ordering, the degree of variance allowed in their relative positionings, and relate this to an evolved model of the relationships between corresponding and 'hidden' transcription factors. Individual promoter motifs could be represented using conventional motif representations or more exotic representations; and, as previous approaches have shown, hybrid representations and the use of local search algorithms as variation operators both provide attractive options.

Finally, there is the issue of evolvability. Evolvability is key to the effective behaviour of EAs yet, in application-led research especially, it is often over-looked as a significant consideration. There is a growing amount of general theory regarding how evolvability may be introduced to solution representations, much of which has only been applied within limited domains, often to toy problems. Since motif discovery is a difficult problem, applying this research to motif representations could be of benefit to both motif discovery and EC theory.

5 Acknowledgments

The authors wish to thank Finn Drabløs at NTNU for his contribution to the ideas developed in this paper. The molecular graphics images in Figure 1 were produced using the UCSF Chimera package from the Computer Graphics Laboratory, University of California, San Francisco (supported by NIH P41 RR-01081). Molecular model data was provided by the RCSB PDB website at http://www.pdb.org/ [7]. Sequence logos were created with WebLogo, a web-based application hosted at http://weblogo.berkeley.edu/, using data from SwissProt [9] and PROSITE [22].

References

[1] L. A. Anbarasu, V. Sundararajan, and N. Narayanasamy. Parallel genetic algorithm for performance-driven sequence alignment. In L. Spector, E. D. Goodman, A. Wu, et al., eds., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, p. 1015. Morgan Kaufmann, San Francisco, California, USA, 7-11 Jul. 2001.

[2] T. K. Attwood. The PRINTS database: a resource for identification of protein families. *Brief Bioinform*, 3(3):252–63, Sep 2002.

[3] R. M. A. Azad and C. Ryan. Structural emergence with order independent representations. In E. C.-P. et al., ed., *Proceedings of the 2003 Genetic and Evolutionary Computation Conference, GECCO 2003*, vol. 2724 of *Lecture Notes in Computer Science*, pp. 1626–1638. Springer-Verlag, 2003.

[4] T. L. Bailey and W. S. Noble. Searching for statistically significant regulatory modules. *Bioinformatics*, 19 Suppl 2:II16–II25, Oct 2003.

[5] P. Baldi and S. Brunak. *Bioinformatics : the machine learning approach.* MIT Press, 2 edn., 2001. Baldi.

[6] A. Bateman, L. Coin, R. Durbin, et al. The Pfam protein families database. *Nucleic Acids Res*, 32(Database issue):D138–41, Jan 2004.

[7] H. M. Berman, J. Westbrook, Z. Feng, et al. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[8] K. Blekas, D. I. Fotiadis, and A. Likas. Motif-based protein sequence classification using neural networks. *Journal of Computational Biology*, 12(1):64–82, February 2005.

[9] B. Boeckmann, A. Bairoch, R. Apweiler, et al. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res*, 31(1):365–70, Jan 2003.

[10] A. Brazma, I. Jonassen, I. Eidhammer, et al. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5(2):277–304, 1998.

[11] C. Bru, E. Courcelle, S. Carrre, et al. The ProDom database of protein domain families: more emphasis on 3D. *Nucleic Acids Res*, 33 Database Issue:D212–5, Jan 2005.

[12] L. Cai, D. Juedes, and E. Liakhovitch. Evolutionary computation techniques for multiple sequence alignment. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pp. 829–835. IEEE Press, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 Jul. 2000.

[13] R. Carr, W. Hart, N. Krasnogor, et al. Alignment of protein structures with a memetic evolutionary algorithm. In W. B. Langdon, E. Cantú-Paz, K. Mathias, et al., eds., *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1027–1034. Morgan Kaufmann Publishers, New York, 9-13 Jul. 2002.

[14] B. C. H. Chang, A. Ratnaweera, H. Halgamuge, et al. Particle swarm optimisation for protein motif discovery. *Genetic Programming and Evolvable Machines*, 5(2):203–214, 2004.

[15] K. E. Chapman and S. J. Higgins. *Regulation of Gene Expression*, vol. 37 of *Essays in Biochemistry.* Portland Press, 2001.

[16] C. Chau, S. Kwong, C. Diu, et al. Optimization of HMM by a genetic algorithm. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 3, pp. 1727 – 1730. 1997.

[17] K. Chellapilla and G. B. Fogel. Multiple sequence alignment using evolutionary programming. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, et al., eds., *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 445–452. IEEE Press, Mayflower Hotel, Washington D.C., USA, 6-9 Jul. 1999.

[18] R. Choudhury. Application of evolutionary computation for multiple sequence alignment. Project report, Stanford University, 2003.

[19] M. Conrad. The geometry of evolution. *BioSystems*, 24:61–81, 1990.

[20] D. Corne, A. Meade, and R. Sibly. Evolving core promoter signal motifs. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pp. 1162–1169. IEEE Press, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001.

[21] R. Durbin, S. R. Eddy, A. Krogh, et al. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1999.

[22] L. Falquet, M. Pagni, P. Bucher, et al. The PROSITE database, its status in 2002. *Nucleic Acids Research*, 30:235–238, 2002.

[23] M. Frith, U. Hansen, and Z. Weng. Detection of cis-element clusters in higher eukaryotic DNA. *Bioinformatics*, 17(10):878–89, Oct 2001.

[24] D. E. Goldberg, K. Deb, H. Kargupta, et al. Rapid accurate optimization of difficult problems using fast messy genetic algorithms. In S. Forrest, ed., *Proceedings of The Fifth International Conference On Genetic Algorithms.* Morgan Kaufmann, 1993.

[25] R. R. Gonzalez, C. M. Izquierdo, and J. Seijas. Multiple protein sequence comparison by genetic algorithms. In S. K. Rogers, D. B. Fogel, J. C. Bezdek, et al., eds., *Proceedings of SPIE*, vol. 3390 of *Applications and Science of Computational Intelligence*, pp. 99–102. 1998.

[26] R. Goodacre and D. B. Kell. Evolutionary computation for the interpretation of metabolome data. In G. G. Harrigan and R. Goodacre, eds., *Metabolic profiling: its role in biomarker discovery and gene function analysis*, pp. 239–256. Kluwer, Boston, 2003.

[27] K. Hanada, T. Yokoyama, and T. Shimizu. Multiple sequence alignment by genetic algorithm. *Genome Informatics*, 11:317–318, 2000.

[28] S. Handley. Predicting whether or not a nucleic acid sequence is an E. coli promoter region using genetic programming. In *Proceedings of the First International Symposium on Intelligence in Neural and Biological Systems INBS-95*, pp. 122–127. IEEE Computer Society Press, Herndon, Virginia, USA, 29-31 May 1995.

[29] A. Heddad, M. Brameier, and M. MacCallum. Evolving regular expression-based sequence classifiers for protein nuclear localisation. In G. R. Raidl, S. Cagnoni, J. Branke, et al., eds., *Applications of Evolutionary Computing, EvoWorkshops2004*, vol. 3005 of *LNCS*, pp. 31–40. Springer Verlag, Coimbra, Portugal, 5-7 Apr. 2004.

[30] S. Henikoff, J. Henikoff, and S. Pietrokovski. Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics*, 15(6):471–9, Jun 1999.

[31] L. S. Ho and J. Rajapakse. High sensitivity technique for translation initiation site detection. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 153– 159. 2004.

[32] J.-T. Horng, C.-M. Lin, B.-J. Liu, et al. Using genetic algorithms to solve multiple sequence alignments. In D. Whitley, D. Goldberg, E. Cantu-Paz, et al., eds., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pp. 883–890. Morgan Kaufmann, Las Vegas, Nevada, USA, 10-12 Jul. 2000.

[33] D. Howard and K. Benson. Evolutionary computation method for pattern recognition of cis-acting sites. *Biosystems*, 72(1-2):19–27, Nov. 2003. Special Issue on Computational Intelligence in Bioinformatics.

[34] Y.-J. Hu. Biopattern discovery by genetic programming. In J. K. et al, ed., *Proceedings Genetic Programming 1998*, pp. 152–157. Morgan Kaufmann, 1998.

[35] K. Huang, J. Louis, L. Donaldson, et al. Solution structure of the MEF2A-DNA complex: structural basis for the modulation of DNA bending and specificity by MADS-box transcription factors. *EMBO J.*, 19:2615–2628, 2000.

[36] M. Ishikawa, T. Toya, Y. Totoki, et al. Parallel iterative aligner with genetic algorithm. In *Genome Informatics Workshop IV*, pp. 13–22. 1993.

[37] M. Isokawa, M. Wayama, and T. Shimizu. Multiple sequence alignment using a genetic algorithm. *Genome Informatics*, 7:176–177, 1996.

[38] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

[39] J. Koza, F. Bennett, and D. Andre. Using programmatic motifs and genetic programming to classify protein sequences as to extracellular and membrane cellular location. In V. W. Porto, N. Saravanan, D. Waagen, et al., eds., *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, vol. 1447 of *LNCS*. Springer-Verlag, Mission Valley Marriott, San Diego, California, USA, 25-27 Mar. 1998.

[40] J. R. Koza and D. Andre. Automatic discovery of protein motifs using genetic programming. In X. Yao, ed., *Evolutionary Computation: Theory and Applications*. World Scientific, Singapore, 1999.

[41] S. Kwong, C. W. Chaua, K. F. Manb, et al. Optimisation of HMM topology and its model parameters by genetic algorithms. *Pattern Recognition*, 34(2):509–522, 2001.

[42] C.-C. Lai and S.-W. Chung. Multiple DNA sequences alignment by means of genetic algorithm. In *Design and application of hybrid intelligent systems*, pp. 224–232. IOS Press, 2003.

[43] D. Latchman. *Eukaryotic Transcription Factors*. Academic Press, 3 edn., 1999.

[44] S. Leopold. *An alignment graph based evolutionary algorithm for the multiple sequence alignment problem*. Master's thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, 2004.

[45] A. M. Lesk. *Introduction to Protein Architecture*. Oxford University Press, 2000.

[46] O. Littlefield, Y. Korkhin, and P. Sigler. The structural basis for the oriented assembly of a TBP/TFB/promoter complex. *Proc. Natl. Acad. Sci. USA*, 96:13668–13673, 1999.

[47] M. A. Lones. *Enzyme Genetic Programming: Modelling Biological Evolvability in Genetic Programming*. Ph.D. thesis, Department of Electronics, University of York, 2003.

[48] N. J. Mulder, R. Apweiler, T. K. Attwood, et al. InterPro, progress and status in 2005. *Nucleic Acids Res*, 33 Database Issue:D201–5, Jan 2005.

[49] C. L. Nehaniv. Editorial for special issue on evolvability. *BioSystems*, 69:77–81, 2003.

[50] H. D. Nguyen, I. Yoshihara, K. Yamamori, et al. A parallel hybrid genetic algorithm for multiple protein sequence alignment. In D. B. Fogel, M. A. El-Sharkawi, X. Yao, et al., eds., *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pp. 309–314. IEEE Press, 2002.

[51] H. D. Nguyen, I. Yoshihara, Y. Yamamori, et al. Improved GA-based method for multiple protein sequence alignment. In R. Sarker, R. Reynolds, H. Abbass, et al., eds., *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, pp. 1826–1832. IEEE Press, Canberra, 8-12 Dec. 2003.

[52] C. Notredame and D. G. Higgins. SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res.*, 24(8):1515–1524, 1996.

[53] B. Olsson. Using evolutionary algorithms in the design of protein fingerprints. In W. Banzhaf, J. Daida, A. E. Eiben, et al., eds., *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, pp. 1636–1642. Morgan Kaufmann, Orlando, Florida, USA, 13-17 Jul. 1999.

[54] A. Pedersen, P. Baldi, Y. Chauvin, et al. The biology of eukaryotic promoter prediction—a review. *Comput Chem*, 23(3-4):191–207, Jun 1999.

[55] D. Pribnow. Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. *Proc. Natl. Acad. Sci. USA*, 72:784–788, 1975.

[56] B. J. Ross. The evolution of stochastic regular motifs for protein sequences. *New Generation Computing*, 20(2):187–213, Feb. 2002.

[57] L. Sheneman and J. A. Foster. Evolving better multiple sequence alignments. In K. Deb, R. Poli, W. Banzhaf, et al., eds., *Genetic and Evolutionary Computation – GECCO-2004, Part I*, vol. 3102 of *Lecture Notes in Computer Science*, pp. 449–460. Springer-Verlag, Seattle, WA, USA, 26-30 Jun. 2004.

[58] L. J. Sheneman and J. A. Foster. Evolving guide trees in progressive multiple sequence alignment. In A. M. Barry, ed., *GECCO 2003: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, pp. 89–91. AAAI, Chigaco, 11 Jul. 2003.

[59] C. Shyu, L. Sheneman, and J. A. Foster. Multiple sequence alignment with evolutionary computation. *Genetic Programming and Evolvable Machines*, 5(2):121–144, 2004. Prepublication Date: 03/23/2004.

[60] M. Slimane, G. Venturini, J. P. A. de Beauville, et al. Optimizing hidden Markov models with a genetic algorithm. In J.-M. Alliot, E. Lutton, E. M. A. Ronald, et al., eds., *Artificial Evolution*, vol. 1063 of *Lecture Notes in Computer Science*, pp. 384–396. Springer, 1995.

[61] G. Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, January 2000.

[62] J. Thompson, D. Higgins, and T. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–80, Nov 1994.

[63] R. Thomsen. Evolving the topology of hidden markov models using evolutionary algorithms. In J. J. M. Guervós, P. Adamidis, H.-G. Beyer, et al., eds., *Parallel Problem Solving from Nature - PPSN VII*, vol. 2439 of *Lecture Notes in Computer Science*, pp. 861–870. Springer, 2002.

[64] R. Thomsen, G. B. Fogel, and T. Krink. A clustal alignment improver using evolutionary algorithms. In D. B. Fogel, X. Yao, G. Greenwood, et al., eds., *Proceedings of the Fourth Congress on Evolutionary Computation (CEC-2002)*, vol. 1, pp. 121–126. 2002.

[65] E. E. Vallejo and F. Ramos. Evolving Turing machines for biosequences recognition and analysis. In J. F. Miller, M. Tomassini, P. L. Lanzi, et al., eds., *Genetic Programming, Proceedings of EuroGP'2001*, vol. 2038 of *LNCS*, pp. 192–203. Springer-Verlag, Lake Como, Italy, 18-20 Apr. 2001.

[66] E. E. Vallejo and F. Ramos. Evolutionary two-dimensional DNA sequence alignment. In E. Cantú-Paz, J. A. Foster, K. Deb, et al., eds., *Genetic and Evolutionary Computation – GECCO-2003*, vol. 2723 of *LNCS*, pp. 429–430. Springer-Verlag, 12-16 Jul. 2003.

[67] A. Vanet, L. Marsan, and M.-F. Sagot. Promoter sequences and algorithmical methods for identifying them. *Res. Microbiol.*, 150:779–799, 1999.

[68] M. Wayama, K. Takahashi, and T. Shimizu. An approach to amino acid sequence alignment using a genetic algorithm. *Genome Informatics*, 6:122–123, 1995.

[69] T. Werner. Models for prediction and recognition of eukaryotic promoters. *Mamm Genome*, 10(2):168–75, Feb 1999.

[70] T. Werner. The state of the art of mammalian promoter recognition. *Brief Bioinform*, 4(1):22–30, Mar 2003.

[71] K.-J. Won, A. Prügel-Bennett, and A. Krogh. Training HMM structure with genetic algorithm for biological sequence analysis. *Bioinformatics*, 20(18):3613–9, Dec 2004.

[72] T. Yada, M. Ishikawa, H. Tanaka, et al. Extraction of hidden Markov model representations of signal patterns in DNA sequences. In *Pacific Symposium on Biocomputing*, pp. 686–96. 1996.

[73] C. Zhang and A. K. Wong. A genetic algorithm for multiple molecular sequence alignment. *Comput Appl Biosci.*, 13(6):565–581, 1997.