# Designing a Novel Hybrid Swarm Based Multiobjective Evolutionary Algorithm for Finding DNA Motifs

David L. González-Álvarez
Department of Technologies of Computers and Communications
ARCO Research Group
University of Extremadura (Spain)
dlga@unex.es

Miguel A. Vega-Rodríguez
Department of Technologies of Computers and Communications
ARCO Research Group
University of Extremadura (Spain)
mavega@unex.es

## ABSTRACT

In this paper we present a novel local search for improving the ability of multiobjective evolutionary algorithms when finding repeated patterns -motifs- in DNA sequences. In the metaheuristic design, two competing goals must be taken into account: exploration and exploitation. Exploration is needed to cover most of the optimization problem search space and provide a reliable estimation of the global optimum. In turn, exploitation is also important since normally the solutions refinement allows the achievement of better results. In this work we take advantage of both concepts by combining the exploration capabilities of a population-based evolutionary algorithm and the power of a local search, especially designed to optimize the Motif Discovery Problem (MDP). For doing this, we have implemented a new hybrid multiobjective metaheuristic based on Artificial Bee Colony (ABC). After analyzing the results achieved by this algorithm, named Hybrid-MOABC (H-MOABC), and comparing them with those achieved by three multiobjective evolutionary algorithms and thirteen well-known biological tools, we prove that the hybridization computes accurate biological predictions on real genetic instances in an optimum way. In fact, to the best of our knowledge, the results presented in this paper improve those presented in the literature.

## Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization*; J.3 [**Life and Medical Science**]: Biology and Genetics

## General Terms

Algorithms


*GECCO'13 Companion,* July 6–10, 2013, Amsterdam, The Netherlands.


## Keywords

Hybrid algorithm; artificial bee colony; multiobjective optimization; motif discovery; DNA.

## 1. INTRODUCTION

Transcriptional regulation, the primary genetic regulation, is performed by interactions (bindings) of regulatory elements. Although these mechanisms are not yet completely understood, numerous efforts are being invested in their understanding. What is known is that certain special proteins called Transcription Factors (TF), or transcriptional elements, bind to certain small substrings in DNA forming the Transcription Factor Binding Sites (TFBS) [19]. As a result of these unions, the gene expression process, which is the process whereby the genes are transcribed into RNA, is enabled or disabled. Identifying these TFBSs and other elements that control gene expression, in addition to the interactions between different TFs, may explain the origin of living organisms, providing important information about its complexity and evolution. The optimization problem addressed in this work, the Motif Discovery Problem (MDP), aims to discover small DNA patterns -motifs- with some biological significance as being TFBSs. For doing this, the MDP is formulated and modeled as a multiobjective optimization problem whose main objective is to find solutions in the midst of a huge amount of biological information in DNA sequences. MDP defines three conflicting objective functions to be maximized: motif length, support, and similarity and, for tackling it, we propose the use of a new hybrid

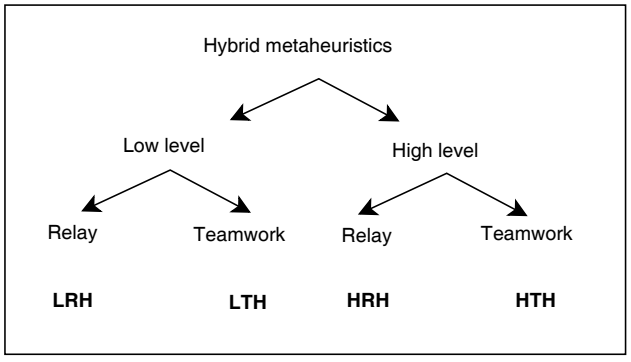



**Figure 1: Classification of hybrid metaheuristics in terms of design issues.**

multiobjective evolutionary algorithm named Hybrid Multiobjective Artificial Bee Colony (H-MOABC) based on the multiobjective metaheuristic presented in [6].

In recent years the interest on hybrid metaheuristics in the optimization field has increased. In fact, the best results obtained in many optimization problems are achieved by hybrid techniques [15]. There are numerous classifications defined to organize this kind of algorithms, but one of the best known is presented in [16]. This classification distinguishes one first level that includes low-level and high-level hybridizations. In the low-level hybridization a function of a given metaheuristic is replaced by another algorithm, an example might be to insert a local search that optimizes the solutions obtained in a generation of a population-based algorithm. In high-level hybrid algorithms there is no relationship between the internal working of the metaheuristics, but they collaborate internally maintaining its original operation, an example may be an algorithm that optimizes a solution generated by another technique. In turn, we can distinguish two kind of hybridizations: relay or teamwork, resulting in low-level relay or teamwork hybridizations (LRH and LTH) and high-level relay or teamwork hybridizations (HRH and HTH). In relay hybridization, a set of metaheuristics (or functions) are executed one after another by using the output of the previous technique as input. On the other hand, we can define a cooperative optimization model that evolve in a parallel way by considering the teamwork hybridization. In Figure 1 we show a graphical representation of this classification. In this work we focus on the first kind of hybridization (LRH), applying a local search at the end of each evolutionary step of a multiobjective population-based evolutionary algorithm. As we have already said, this algorithm is based on Artificial Bee Colony (ABC) [7]. To demonstrate the quality of the proposed method, the results achieved by the hybrid algorithm are evaluated and analyzed in broad comparative sections. First, we compare the results obtained by H-MOABC with those achieved by the corresponding non-hybridized version, MOABC, demonstrating the hybridization advantages. Then, we compare H-MOABC with Non-dominated Sorting Genetic Algorithm II (NSGA-II, [2]) and Strength Pareto Evolutionary Algorithm 2 (SPEA2, [20]). Finally, we also compare the predictions made by the hybrid with those predicted by thirteen well-known biological tools.

The rest of the paper is organized as follows. In the following section we define the MDP, including a brief review of the problem state-of-the-art. In Section 3 we detail the proposed function, explaining its operation and the adjustments made in MOABC for its incorporation. The experimental methodology and results are included in Section 4. Then, in Section 5 we compare the predictions made by H-MOABC with those predicted by other thirteen biological tools. Finally, some conclusions and future lines are discussed in Section 6.

# 2. MOTIF DISCOVERY PROBLEM

In this section we include a review of the most important tools and algorithms used for discovering motifs in DNA sequences, explaining the motivations that have driven us to use a multiobjective formulation. To do this, we discuss the advantages and limitations presented by these techniques. Then, we define the MDP multiobjective formulation in mathematical terms, and we solve an artificial MDP to better understanding the explained concepts.

## 2.1 Related Work

There are many proposals based on evolutionary techniques for discovering motifs in DNA sequences. Some examples are FMGA (Finding Motifs by Genetic Algorithm) [9], a Genetic Algorithm (GA) based on the SAGA operators; St-GA (Structured Genetic Algorithm) [14]; MDGA (Motif Discovery using a Genetic Algorithm) [1]; or GAME [18], a GA used for detecting cis-regulatory elements. Although there are other proposals such as TS-BFO [11], EDA/DE proposed by the same authors in [12], or PCEA [10]; if we focus on the proposals presented for discovering motifs, we can note how many of them are based on GAs. Furthermore, almost all listed proposals employ a single objective to discover motifs, the motif length is given beforehand, and try to find motifs in all sequences. Furthermore, [5] and [4] propose a new multi-term fitness function, remedying some of the previously mentioned limitations. However, from our point of view, the best way to address the MDP is using a multiobjective approach. In [8], the author propose a multiobjective GA based method named MOGAMOD for discovering motifs, demonstrating the advantages of using this multiobjective methodology. Due to the advantages of this kind of optimization, we have also adopted it in our problem definition. Unfortunately, we have not been able to compare the results obtained by our algorithms with those obtained by MOGAMOD, due to changes that we have made in our problem definition.

With respect to non-evolutionary techniques, we can find a lot of biological tools in the literature. There are different types of tools, one of the best known classifications distinguish two kind of tools: string-based and probabilistic biological tools. String-based biological tools guarantee global optimality and they are appropriate for finding totally constrained motifs, some examples are Oligo/Dyad-Analysis, MITRA (Mismatch Tree Algorithm), YMF, QuickScore, and Weeder. Among the probabilistic tools, which usually imply representation of the motifs by a position weight matrix, we highlight Consensus, MEME (Multiple EM for Motif Elicitation), Improbizer, AlignACE (Aligns Nucleic Acid Conserved Elements), ANN_Spec (Artificial Neural Network with a GS method to define the Specificity), MotifSampler, GLAM (Gapless Local Alignment of Multiple sequences), and SeSiMCMC (Sequence Similarities by Markov Chain Monte-Carlo). Thanks to the work [17], we can compare the results obtained by our algorithms with those obtained by these thirteen listed biological tools.

## 2.2 Mathematical Formulation

To solve the MDP we have to optimize three conflicting objective functions: motif length, support, and similarity; as well as satisfy a set of constraints. Given a set of sequences $S = \{S_i | i = 1, 2, ..., D\}$ of nucleotides defined on the alphabet $B = \{A, C, G, T\}$. $S_i = \{S_i^j | j = 1, 2, ..., w_i\}$ is a sequence of nucleotides, where $w_i$ is the sequence width. The set of all the subsequences contained in $S$ is $\{s_i^{j_i} | i = 1, 2, ..., D, j_i = 1, 2, ..., w_i - l + 1\}$, where $j_i$ is the binding site of a possible motif instance $s_i^j$ on sequence $S_i$, and $l$ is the motif length, the first objective to be maximized. To obtain the values of the other two objectives we have to build the Position Indicator Matrix (PIM) $A = \{A_i | i = 1, 2, ..., D\}$

of the motif, where $A_i = \{A_i^j | j = 1, 2, ..., w_i\}$ is the indicator row vector with respect to a sequence $S_i$. $A_i^j$ is 1 if the position $j$ in $S_i$ is a starting position of a binding site, and 0 otherwise. We refer to the number of motif instances as $|A| = \sum_{i=1}^{D} \sum_{j=1}^{w_i} A_i^j$. We also require to find the consensus motif, which is a string abstraction of the motif instances. Only those sequences that achieve a motif instance of certain quality with respect to the consensus motif are taken into account when we build the final motif. This is indicated by the second objective to be maximized, the support. Furthermore, $S(A) = \{S(A)_1, S(A)_2, ..., S(A)_{|A|}\}$ is a set of $|A|$ motif instances, where $S(A)_i = S(A)_i^1 S(A)_i^2 ... S(A)_i^l$ is the $i$th motif instance in $|A|$. $S(A)$ can also be expanded as $(S(A)^1, S(A)^2, ..., S(A)^l)$, where $S(A)^j = S(A)_1^j S(A)_2^j ... S(A)_{|A|}^j$ is the list of nucleotides on the $j$th position in the motif instances. Then, we build the Position Count Matrix (PCM) $N(A)$ with the different nucleotide bases on each position of the candidate motifs ($A$) which have passed the threshold marked by the support. $N(A) = \{N(A)^1, N(A)^2, ..., N(A)^l\}$, and $N(A)^j = \{N(A)_b^j | b \in B\}$, where $N(A)_b^j = |\{S(A)_i^j | S(A)_i^j = b\}|$. The dominant nucleotides of each position are normalized in the Position Frequency Matrix (PFM) $\widehat{N} = \frac{N(A)}{|A|}$. Finally, we calculate the third objective value, the similarity, by averaging all the dominance values of each PFM column, as is indicated in the following expression:

$$Similarity(Motif) = \frac{\sum_{i=1}^{l} max_b\{f(b,i)\}}{l} \quad (1)$$

where $f(b,i)$ is the score of nucleotide $b$ in column $i$ in the PFM and $max_b\{f(b,i)\}$ is the value of the dominant nucleotide in column $i$.

To summarize, motif length objective function indicates the number of nucleotides that compose the solution, support represents the number of sequences used to build the final solution (those that share at least a 50% of nucleotides with the consensus motif), and similarity measures the similarity among the substrings that have exceeded the previously mentioned threshold value of support.

As far as constraints is concerned, considering that motifs are usually formed by a few nucleotides [3], we have restricted the motif length to the range [7,64]. We have also restricted the minimum support value to 2 in the data sets composed by 4 or less sequences, and to 3 for the other ones (more than 4 sequences). In this way, we ensure that at least a 20% of the instance sequences support the quality of the final solution. Finally, we have applied the complexity concept detailed in [5] and extended in [4] by using the following expression:

**Table 1: An artificial motif discovery problem.**

| Organism | Seq. | Start | Sites | Concordance | |
|---|---|---|---|---|---|
| Human | 0 | -365 | GTGATATTCC | 6/10 | √ |
| Human | 1 | -87 | GGAAACTCCG | 8/10 | √ |
| Human | 2 | -403 | TGAGACTGCC | 6/10 | √ |
| Human | 3 | -199 | GTTGAATAAG | 4/10 | X |
| Human | 4 | -214 | GGGAAATCCC | 9/10 | √ |
| Human | 5 | -257 | GGAATTTCCC | 8/10 | √ |

(a) Consensus motif.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A: | 0 | 0 | 3 | 4 | 3 | 2 | 0 | 0 | 0 | 0 |
| C: | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 5 | 4 |
| G: | 4 | 4 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| T: | 1 | 1 | 0 | 0 | 2 | 1 | 5 | 1 | 0 | 0 |

(b) Position count matrix.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A: | 0.0 | 0.0 | 0.6 | 0.8 | 0.6 | 0.4 | 0.0 | 0.0 | 0.0 | 0.0 |
| C: | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.4 | 0.0 | 0.6 | 1.0 | 0.8 |
| G: | 0.8 | 0.8 | 0.4 | 0.2 | 0.0 | 0.0 | 0.0 | 0.2 | 0.0 | 0.2 |
| T: | 0.2 | 0.2 | 0.0 | 0.0 | 0.4 | 0.2 | 1.0 | 0.2 | 0.0 | 0.0 |

(c) Position frequency matrix.

(d) Resulting motif.

**Figure 2: Consensus motif, position count matrix, position frequency matrix, and resulting motif of the MDP example included in Table 1.**

$$Complexity = \log_N \frac{l!}{\prod(n_i!)} \quad (2)$$

where $N = 4$ for DNA sequences, $l$ is the motif length, and $n_i$ is the number of nucleotides of type $i \in \{A, C, G, T\}$. This constraint ensures a minimum number of base changes (A, C, G, and T) in the discovered DNA strings. For instance, if we consider the motif 'AAAA' ($n_A = 4$, $n_T = 0$, $n_G = 0$, and $n_C = 0$) we will obtain a minimum complexity since we get the highest value in $\prod(n_i!)$. Otherwise, if we have, for example, the 'ACGT' motif ($n_A = 1$, $n_T = 1$, $n_G = 1$, and $n_C = 1$) we will obtain the highest complexity for motif length equal to 4. In our algorithms we have established a minimum complexity of 50%. These constraints must be met by all the generated solutions, i.e., if a solution does not meet all the defined constraints, it will be discarded and will not be part of the population.

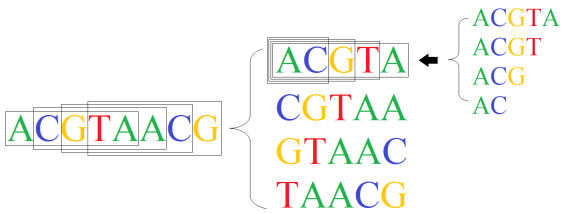
## 2.3 MDP Example

In Table 1 and Figure 2 we represent the resolution of an artificial MDP of length 10 ($motif\ length = 10$) to facilitate the understanding of the mathematical concepts described in the previous subsection. In Table 1 we indicate the organism to which each sequence belongs, the sequence identifier, the starting location of each candidate motif, the corresponding sites, and the achieved concordance rate. This latter value is obtained by comparing each candidate motif with the consensus motif (Figure 2(a)) at a nucleotide level. As we have already explained, those candidates that reach a minimum concordance rate of 50% will be taken into account in the second objective function (support) and, consequently, will

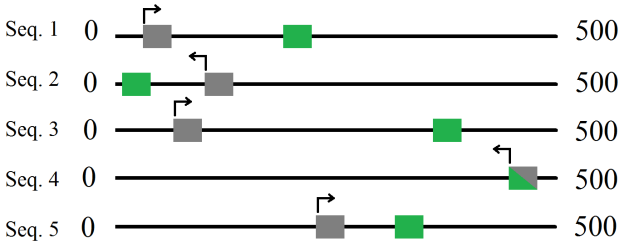
be used to compose the final solution (Figure 2(d)). In this example we have *support* = 5. To calculate the similarity value we have to build the PCM and PFM (Figures 2(b) and 2(c)) and apply the equation 1 by using the dominant values of each position. In this case we obtain a *similarity* of a 74%.

## 3. LOCAL SEARCH OPERATION

The local search presented in this paper aims to improve the process of discovering motifs in DNA sequences. As we will see in the following sections, its operation is simple as well as effective for improving the quality of the solutions discovered by a given metaheuristic. The implemented local search defines three important parameters: the window size ($WS$), the search direction ($DIR$), and the reference string ($REF$). The window size defines the substring length that we will attempt to find in the corresponding DNA sequences. We have conducted experiments with window sizes ranging from $WS = 1$ to $WS = 7$ since, according to the defined problem constraints, the minimum motif length is 7. The search direction is the direction that we must take to find the selected substring. $DIR = 0$ indicates that we begin the search from the beginning of the sequence, i.e., from the nucleotide 0; $DIR = 1$ that we have to choose a random direction (right or left) from the starting positions of the corresponding candidate motif; and $DIR = 2$ that we have to check the quality of the solutions resulting from searching in both directions (right and left from the starting position) and select the one which produces the best solution. Finally, the $REF$ parameter indicates which motif (among all candidates and the consensus motif) is used as reference. With $REF = 0$ we select the candidate motif of the first sequence, with $REF = 1$ we choose the candidate of a randomly selected sequence, with $REF = 2$ we use the consensus motif, and with $REF = 3$ we consider the candidate closer to the consensus motif at a nucleotide level. By using the example included in Figure 3, the following steps explain the operation of the defined local search:




(a) Considering $WS = 5$, possible strings to be searched (arrow indicates the selected substring).




(b) Considering $DIR = 1$, changes made in the starting candidate motif positions (in gray the old locations and in green the new ones).

**Figure 3: Graphical representation of the local search operation.**

**Step 1:** We set the value of the $WS$, $DIR$, and $REF$ parameters. - *Example: We use $WS = 5$, $DIR = 1$, and $REF = 2$.*

**Step 2:** We load the reference string considering the $REF$ parameter. - *Example: As $REF = 2$, we use the consensus motif as reference string, in this case: ACGTAACG.*

**Step 3:** We repeat the following steps on all sequences where we have found any candidate motif. - *Example: We repeat the following steps on the 5 sequences that compose the solution.*

**Step 3.1:** We select a random substring among the possible windows. - *Example: As $WS = 5$ and the consensus motif is ACGTAACG, we randomly choose one among the four possible substrings: ACGTA, CGTAA, GTAAC, or TAACG (see Figure 3(a)). In this case we select the first substring: ACGTA.*

**Step 3.2:** We search the substring of size $WS$ in the corresponding sequence following the indications of $DIR$. - *Example: With $DIR = 1$ we search the substring in the corresponding sequence following a random direction, right or left of the candidate motif starting position.*

**Step 3.3:** If we find the substring we update the starting location of the processed candidate motif.

**Step 3.4:** If $WS > 1$ and we do not find the substring, we return to the step 3.2, reducing the value of $WS$ by one. - *Example: If we do not find the ACGTA substring in the sequence and $WS > 1$, we reduce the $WS$ value in one unit ($WS = WS - 1$) and we will search, in this case, the ACGT substring.*

**Step 4:** We evaluate the resulting solution and if it is able to dominate the previous one, we exchange them.

## 4. EXPERIMENTAL RESULTS

To analyze the performance of the proposed local search, we have embedded it in a multiobjective evolutionary algorithm specially dedicated to discovering motifs in DNA

**Table 2: Individual representation.**

| | Seq. 0 | Seq. 1 | Seq. 2 | | Seq. n-1 |
|---|---|---|---|---|---|
| Motif Length | $S_0$ | $S_1$ | $S_2$ | ... | $S_{n-1}$ |

**Table 3: Instance properties.**

| | Number of Sequences | Sequence Size | Number of Nucleotides | Established Runtimes (s) |
|---|---|---|---|---|
| dm01g | 4 | 1500 | 6000 | 20 |
| dm04g | 4 | 2000 | 8000 | 20 |
| dm05g | 5 | 2500 | 12500 | 20 |
| hm03r | 10 | 1500 | 15000 | 30 |
| hm04m | 13 | 2000 | 26000 | 30 |
| hm16g | 7 | 3000 | 21000 | 20 |
| mus02r | 9 | 1000 | 9000 | 20 |
| mus03g | 4 | 1500 | 6000 | 20 |
| mus07g | 12 | 500 | 6000 | 30 |
| yst03m | 8 | 500 | 4000 | 20 |
| yst04r | 7 | 1000 | 7000 | 20 |
| yst08r | 11 | 1000 | 11000 | 30 |

**Table 4: Best results (median hypervolumes and interquartile ranges) achieved by MOABC, H-MOABC (after configuring each parameter), NSGA-II, and SPEA2.**

| Instance | MOABC | H-MOABC | NSGA-II | SPEA2 | |
|---|---|---|---|---|---|
| dm01g | 83.24% 0.84% | 86.29% 2.04% | 82.69% 0.74% | 82.87% 0.70% | + |
| dm04g | 83.77% 1.37% | 87.22% 2.00% | 82.24% 1.16% | 82.44% 1.06% | - |
| dm05g | 86.39% 1.01% | 90.84% 0.85% | 85.44% 0.86% | 86.27% 1.14% | - |
| hm03r | 60.18% 2.63% | 69.10% 3.45% | 47.83% 4.02% | 53.06% 1.56% | + |
| hm04m | 53.20% 2.35% | 62.49% 2.45% | 43.32% 3.40% | 46.39% 0.94% | + |
| hm16g | 81.55% 5.52% | 91.53% 2.00% | 70.14% 1.02% | 71.66% 1.70% | + |
| mus02r | 63.58% 2.42% | 72.50% 2.84% | 60.63% 1.20% | 59.60% 1.48% | + |
| mus03g | 79.84% 0.79% | 82.77% 1.23% | 77.48% 0.47% | 77.48% 0.53% | - |
| mus07g | 89.21% 2.93% | 88.34% 3.63% | 87.97% 1.95% | 89.35% 0.52% | - |
| yst03m | 69.68% 2.45% | 76.86% 1.35% | 65.16% 2.08% | 66.24% 1.13% | + |
| yst04r | 75.54% 1.16% | 79.93% 0.98% | 75.18% 0.50% | 71.64% 0.56% | + |
| yst08r | 62.91% 2.30% | 74.81% 2.96% | 66.25% 1.74% | 56.88% 1.12% | + |
| Mean | 74.09% | 80.22% | 70.36% | 70.32% | |
| vs H-MOABC | 11 | 1 | 12 | 11 | |

dm04g: SPEA2 versus NSGA-II

dm05g: MOABC versus SPEA2

mus03g: SPEA2 versus NSGA-II

mus07g: MOABC versus SPEA2, H-MOABC versus NSGA-II

sequences. This algorithm is known as Multiobjective Artificial Bee Colony (MOABC) and it is properly configured (see [6] for further information). Its hybridization have been named Hybrid MOABC (H-MOABC) and it applies the implemented local search instead of exploring the neighborhoods of the employed bee food sources. Thus, the local search is applied to all individuals of the population once for generation. The basic idea behind this hybridization is to maintain its evolutionary scheme and only improve the quality of the discovered solutions. With regard to the experiments, we have carried out 31 independent runs using g++ (GCC) 4.4.5 on a 2.3GHz Intel PC with 1GB RAM. To assess the quality of the obtained results we use the hypervolume indicator [21], establishing the reference point in the theoretical optimum point of each instance; and the set coverage [20]. The representation of the individuals is the same in all algorithms and it includes the motif length and the starting position of each candidate in each sequence (see Table 2). Finally, as benchmark we have used a set of twelve biological instances with genetic information belonging to four organisms: drosophila melanogaster (dm), homo sapiens (hm), mus musculus (mus), and saccharomyces cerevisiae (yst) obtained from [17]. Their properties are described in Table 3.

The first results are included in Table 4. In this table we show the hypervolumes achieved by the MOABC, NSGA-II, and SPEA2 algorithms. In addition, in the last column of the table we show the '+' symbol if the differences among the results of all algorithms are statistically significant, and '-' otherwise. Finally, at the bottom of the table, we indicate in which cases there are no statistically significant differences. We now will discuss these results by parts. First, we can observe how the non-hybridized multiobjective evolutionary algorithm (MOABC) is able to obtain quality results, surpassing those achieved by two standard multiobjective evolutionary algorithms such as NSGA-II and SPEA2. MOABC achieves a percentages equal to 74.09% versus the 70.36% and 70.32% presented by NSGA-II and SPEA2. Concerning hybrid algorithm, we include the partial results obtained after tuning each parameter. We have conducted experiments with $WS = \{1, 2, ..., 7\}$, $REF = \{0, 1, 2, 3\}$, and $DIR = \{0, 1, 2\}$, as it is explained in Section 3. The H-MOABC local search gets the best results with $WS = 6$, $REF = 1$, and $DIR = 2$. If we compare the results achieved by this new hybrid algorithm with those obtained by its non-hybridized version, we can note how the results have been considerably improved, increasing from 74.09% to 80.22% (MOABC versus H-MOABC). To better appreciate this improvement we have included additional information in the last rows of Table 4. We include the mean hypervolume achieved by each technique when they solve the twelve instances and, on the other hand, we also indicate, in the case of the H-MOABC algorithm, the number of instances where it obtains the higher hypervolume; and for the rest (MOABC, NSGA-II, and SPEA2), the number of instances where they get a better result than the H-MOABC algorithm. This information allows us to compare the hybrid algorithm with its non-hybridized version, and with the NSGA-II and SPEA2 algorithms. As we can see, the superiority of the hybrid is clear because H-MOABC achieves the best result in 11 of the 12 solved instances. Finally, we have followed the statistical methodology described in [13] for studying the differences among the results. For doing this, we first apply the Kolmogorov-Smirnov test for

**Table 5: Direct comparison of the outcomes achieved by the algorithms. Each cell gives the fraction of non-dominated solutions obtained by A covered by the non-dominated points from B.**

| A / B | MOABC | H-MOABC | NSGA-II | SPEA2 | Mean |
|---|---|---|---|---|---|
| MOABC | X | 26.14% | 93.87% | 93.33% | 71.11% |
| H-MOABC | 79.42% | X | 91.99% | 90.23% | **87.21%** |
| NSGA-II | 13.52% | 13.41% | X | 64.24% | 30.39% |
| SPEA2 | 16.38% | 14.59% | 43.44% | X | 24.80% |
| Mean | 36.44% | **18.05%** | 76.43% | 82.60% | |

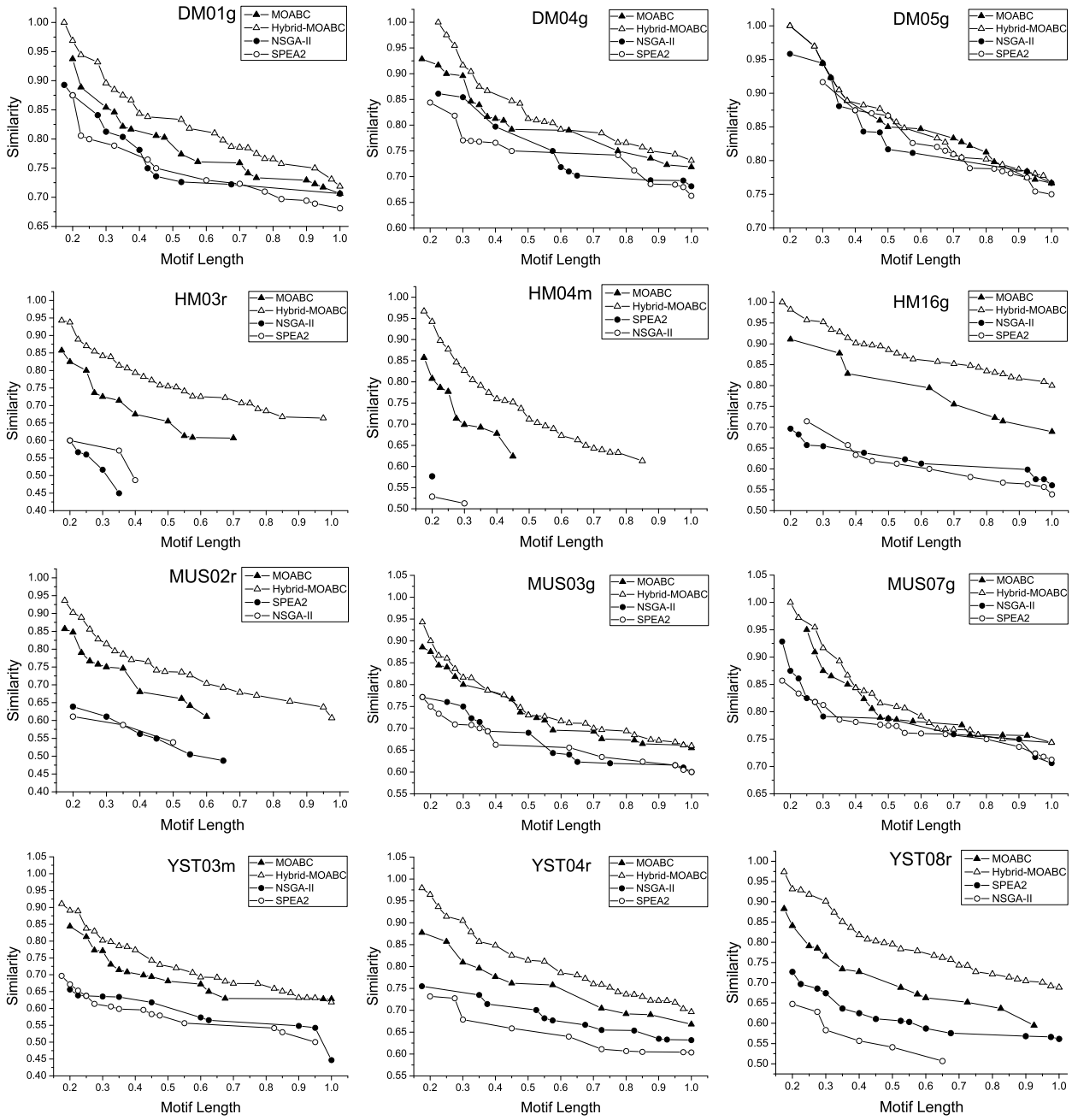



**Figure 4: Comparison among the non-dominated solutions with maximum support discovered by the algorithms.**

analyzing the sample distributions and the Levene test for examining the variance homogeneities. If both tests are positives, we conduct a parametric ANOVA test and, otherwise, we conduct a non-parametric Kruskal-Wallis test, for studying the result differences, always considering a confidence level of 95%. At the foot of the table we indicate in which cases the differences among algorithms are not statistically significant.

In Table 5 we include the results obtained by using the set coverage metric. As we can see, the drawn conclusions reaffirm those achieved in the first comparison. The hybrid algorithm (H-MOABC) is the technique that covers a greater percentage of solutions: 87.21%. In addition, it is also the less covered algorithm: 18.05%. This means that not only the overall quality of the Pareto front is good (demonstrated by the hypervolume metric), but also the solutions that compose them.

Finally, in Figure 4 we show some of the solutions discovered by the compared algorithms (MOABC, H-MOABC, NSGA-II, and SPEA2). Thus, we provide a graphical representation of the quality improvement. More specifically, we have represented the non-dominated solutions with max-

imum support discovered by the algorithms in the solved instances: dm01g, dm04g, dm05g, hm03r, hm04m, hm16g, mus02r, mus03g, mus07g, yst03m, yst04r, and yst08r. Observing these graphs, we can see how the hybrid algorithm fronts achieve greater height (remember that the objective functions have to be maximized), width, and spread. Although this information is limited to the maximum support Pareto fronts, it allows us to give an idea of the improvement achieved by using the presented local search.

## 5. COMPARISON WITH OTHER AUTHORS

To demonstrate the quality of the solutions discovered by the designed hybrid algorithm, we have compared the disovered predictions with those made by thirteen well-known biological tools such as Consensus, MEME, AlignACE, ANN_Spec, Improbizer, MotifSampler, GLAM, SeSiMCMC, Oligo/Dyad-Analysis, MITRA, YMF, QuickScore, and Weeder. To perform this comparison we define a set of four biological indicators named Sensitivity ($nSn$), Positive Predictive Value ($nPPV$), Performance Coefficient ($nPC$), and Correlation Coefficient ($nCC$). These biological indicators measure the correction of the motifs discovered by our algorithm by comparing their positions with the real binding site locations. This comparison is conducted at a nucleotide level by using the following statistical parameters: $TP$ (True-Positives), $TN$ (True-Negatives), $FP$ (False-Positives), and $FN$ (False-Negatives). The methodology followed to perform this comparison, all results presented by the thirteen compared biological tools, and the biological indicator expressions are broadly described in [17].

In Table 6 we include the results of this comparison. Since we are comparing the results obtained by fourteen techniques (thirteen biological tools and our hybrid algorithm) by using four biological indicators in a total of twelve instances, we can imagine the volume of biological data that we have to manage. To organize all this information and make this comparison more understandable, we have followed the described methodology. First, we look for the tool that gets the best result for each instance and each biological indicator (information included in the first and second column of the tables). It is important to note that the biological indicators are in the range [-1,1], where -1 indicates perfect anti-correlation and 1 indicates perfect correlation. After this, we have to select the best multiobjective solution discovered by our algorithm. For doing this, we select, among all non-dominated solutions predicted by our hybrid algorithm, the one with the best combined score (result from adding the four biological indicator values). Thus, the information included in the tables are obtained by a single motif. Once described the information of Table 6, we can analyze its contents. Observing the results, we can see how the predictions made by H-MOABC achieve better results than those predicted by the best biological tools in most cases. Being also important to note that, while many tools are specialized in solving instances of a given organism, our hybrid algorithm is able to obtain good results in all instances, regardless of the organism to which it belongs.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper we propose a new local search to improve the ability of multiobjective metaheuristics when discovering motifs in DNA sequences. The proposed local search

**Table 6: Comparison between the solutions of H-MOABC and the results predicted by thirteen well-known biological tools ("-" when no tool is able to find solutions).**

(a) Sensitivity (nSn).

| Instance | Best tool | Result | H-MOABC |
|---|---|---|---|
| dm01g | SeSiMCMC | 0.344000 | 0.472000 |
| dm04g | MotifSampler | 0.022222 | 0.355556 |
| dm05g | MEME | 0.037500 | 0.181250 |
| hm03r | MEME | 0.063726 | 0.161765 |
| hm04m | AlignACE | 0.005952 | 0.160714 |
| hm16g | - | 0.000000 | 0.341463 |
| mus02r | MEME | 0.094828 | 0.172414 |
| mus03g | AlignACE | 0.281690 | 0.669014 |
| mus07g | ANN_Spec | 0.040000 | 0.520000 |
| yst03m | Improbizer | 0.340136 | 0.176871 |
| yst04r | Consensus | 0.335878 | 0.343511 |
| yst08r | AlignACE | 0.387097 | 0.555556 |

(b) Positive Predictive Value (nPPV).

| Instance | Best tool | Result | H-MOABC |
|---|---|---|---|
| dm01g | SeSiMCMC | 0.344000 | 0.737500 |
| dm04g | MotifSampler | 0.032967 | 0.761905 |
| dm05g | MEME | 0.026667 | 0.966667 |
| hm03r | MEME | 0.108333 | 1.000000 |
| hm04m | AlignACE | 0.006061 | 0.450000 |
| hm16g | - | 0.000000 | 0.565657 |
| mus02r | MEME | 0.142857 | 1.000000 |
| mus03g | AlignACE | 0.256410 | 0.678571 |
| mus07g | ANN_Spec | 0.020942 | 0.722222 |
| yst03m | YMF | 0.700000 | 0.928571 |
| yst04r | MITRA | 0.357143 | 0.865385 |
| yst08r | MotifSampler | 0.786408 | 0.782828 |

(c) Performance Coefficient (nPC).

| Instance | Best tool | Result | H-MOABC |
|---|---|---|---|
| dm01g | SeSiMCMC | 0.207730 | 0.404110 |
| dm04g | MotifSampler | 0.013453 | 0.320000 |
| dm05g | MEME | 0.015831 | 0.180124 |
| hm03r | MEME | 0.041801 | 0.161765 |
| hm04m | AlignACE | 0.003012 | 0.134328 |
| hm16g | - | 0.000000 | 0.270531 |
| mus02r | MEME | 0.060440 | 0.172414 |
| mus03g | AlignACE | 0.155039 | 0.508021 |
| mus07g | ANN_Spec | 0.013937 | 0.433333 |
| yst03m | oligodyad | 0.261905 | 0.174497 |
| yst04r | Consensus | 0.202765 | 0.326087 |
| yst08r | MotifSampler | 0.269103 | 0.481366 |

(d) Correlation Coefficient (nCC).

| Instance | Best tool | Result | H-MOABC |
|---|---|---|---|
| dm01g | SeSiMCMC | 0.330043 | 0.583302 |
| dm04g | MotifSampler | 0.013401 | 0.515335 |
| dm05g | MEME | 0.006491 | 0.414609 |
| hm03r | MEME | 0.063601 | 0.397568 |
| hm04m | AlignACE | -0.000400 | 0.266233 |
| hm16g | MEME | -0.005204 | 0.436154 |
| mus02r | MEME | 0.097480 | 0.410754 |
| mus03g | AlignACE | 0.222480 | 0.654279 |
| mus07g | ANN_Spec | 0.006056 | 0.607391 |
| yst03m | oligodyad | 0.437304 | 0.397973 |
| yst04r | Consensus | 0.322430 | 0.540508 |
| yst08r | MotifSampler | 0.470596 | 0.652253 |

is specially designed to optimize the resolution of this biological optimization problem, the Motif Discovery Problem (MDP). It is simple and easy to be understood and configured (it only defines three parameters: $WS$, $REF$, and $DIR$). Moreover, it has also demonstrated a high effectiveness in improving the quality of the solutions found by other techniques. To analyze its performance we have embedded it in a multiobjective evolutionary algorithm resulting in a novel technique named Hybrid Multiobjective Artificial Bee Colony (H-MOABC). In different sections we have compared the achieved results, demonstrating that the hybrid algorithm, which incorporates the defined local search, is able to discover better solutions. Finally, we have also compared the predictions made by H-MOABC with those predicted by thirteen well-known biological tools among which we highlight AlignACE, MEME, and Weeder. The obtained results demonstrate the good biological quality of the solutions predicted by our hybrid algorithm.

As future work we will intend to apply our algorithm to solve more complex instances by using, if necessary, parallelism techniques.

## Acknowledgements

This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the ERDF (European Regional Development Fund), under the contract TIN2012-30685 (BIO project). Thanks also to the Fundación Valhondo for the economic support offered to David L. González-Álvarez.

## 7. REFERENCES

[1] Che, D. Song, Y., and Rashedd, K. MDGA: Motif discovery using a genetic algorithm. *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO'05)* (2005), 447–452.

[2] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6*, 2 (2002), 182–197.

[3] D'haeseleer, P. What are DNA sequence motifs? *Nature Biotechnology 24*, 4 (2006), 423–425.

[4] Fogel, G. B., Porto, V. W., Varga, G., Dow, E. R., Crave, A. M., Powers, D. M., Harlow, H. B., Su, E. W., Onyia, J. E., and Su, C. Evolutionary computation for discovery of composite transcription factor binding sites. *Nucleic Acids Research 36*, 21 (2008), e142, 1–14.

[5] Fogel, G. B., Weekes, D. G., Varga, G., Dow, E. R., Harlow, H. B., Onyia, J. E., and Su, C. Discovery of sequence motifs related to coexpression of genes using evolutionary computation. *Nucleic Acids Research 32*, 13 (2004), 3826–3835.

[6] González-Álvarez, D. L., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., and Sánchez-Pérez, J. M. Comparing multiobjective swarm intelligence metaheuristics for DNA motif discovery. *Engineering Applications of Artificial Intelligence 26*, 1 (2012), 341–326.

[7] Karaboga, D. An idea based on honey bee swarm for numerical optimization. *Technical report-tr06, Erciyes University, Turkey* (2005).

[8] Kaya, M. MOGAMOD: Multi-objective genetic algorithm for motif discovery. *Expert Systems with Applications 36*, 2 (2009), 1039–1047.

[9] Liu, F. F. M., Tsai, J. J. P., Chen, R. M., Chen, S. N., and Shih, S. H. FMGA: finding motifs by genetic algorithm. *Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04)* (2004), 459–466.

[10] Lones, M. A., and Tyrrell, A. M. Regulatory motif discovery using a population clustering evolutionary algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 4*, 3 (2007), 403–414.

[11] Shao, L., and Chen, Y. Bacterial foraging optimization algorithm integrating tabu search for motif discovery. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM'09)* (2009), 415–418.

[12] Shao, L., Chen, Y., and Abraham, A. Motif discovery using evolutionary algorithms. *International Conference of Soft Computing and Pattern Recognition (SOCPAR'09)* (2009), 420–425.

[13] Sheskin, D. J. Handbook of parametric and nonparametric statistical procedures. *4th ed. New York: Chapman & Hall/CRC Press* (2007).

[14] Stine, M., Dasgupta, D., and Mukatira, S. Motif discovery in upstream sequences of coordinately expressed genes. *The 2003 Congress on Evolutionary Computation (CEC'03)* (2003), 1596–1603.

[15] Talbi, E.-G. A taxonomy of hybrid metaheuristics. *Journal of Heuristics 8* (2002), 541–564.

[16] Talbi, E.-G. Metaheuristics: From Design to Implementation. *John Wiley & Sons* (2009).

[17] Tompa, M., Li, N., Bailey, T. L., Church, G. M., De Moor, B., Eskin, E., Favorov, A. V., Frith, M. C., Fu, Y., Kent, W. J., Makeev, V. J., Mironov, A. A., Noble, W. S., Pavesi, G., Pesole, G., Régnier, M., Simonis, N., Sinha, S. Thijs, G., Van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., and Zhu, Z. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology 23*, 1 (2005), 137–144.

[18] Wei, Z., and Jensen, S. GAME: detecting cis-regulatory elements using a genetic algorithm. *Bioinformatics 22*, 13 (2006), 1577–1584.

[19] Zare-Mirakabad, F., Ahrabian, H., Sadeghi, M., Hashemifar, S., Nowzari-Dalini, A., and Goliaei, B. Genetic algorithm for dyad pattern finding in DNA sequences. *Genes & Genetic Systems 84*, 1 (2009), 81–93.

[20] Zitzler, E., Laumanns, M., and Thiele, L. SPEA2: Improving the strength pareto evolutionary algorithm. *Technical report tik-report 103, Swiss Federal Institute of Technology Zurich, Switzeland* (2001).

[21] Zitzler, E., and Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation 3*, 4 (1999), 257–271.