Artificial Intelligence in Medicine 50 (2010) 193–201

Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Multi-marker tagging single nucleotide polymorphism selection using estimation of distribution algorithms

Roberto Santana [a,*], Alexander Mendiburu [b], Noah Zaitlen [c], Eleazar Eskin [c], Jose A. Lozano [b]

[a] Faculty of Informatics, Universidad Politécnica de Madrid, R. 3306, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain
[b] Intelligent Systems Group, University of the Basque Country, Paseo Manuel de Lardizábal 1, 20018 San Sebastian - Donostia, Spain
[c] Computer Science and Human Genetics Group, University of California 1596, 3532-J Boelter Hall, Los Angeles, CA 90095-1596, USA

ARTICLE INFO

Article history:
Received 5 August 2009
Received in revised form 27 May 2010
Accepted 30 May 2010

Keywords:
Estimation of distribution algorithms
Tagging single nucleotide polymorphism selection
Multi-marker selection
HapMap

ABSTRACT

Objectives: This paper presents an optimization algorithm for the automatic selection of a minimal subset of tagging single nucleotide polymorphisms (SNPs).
Methods and materials: The determination of the set of minimal tagging SNPs is approached as an optimization problem in which each tagged SNP can be covered by a single tagging SNP or by a pair of tagging SNPs. The problem is solved using an estimation of distribution algorithm (EDA) which takes advantage of the underlying topological structure defined by the SNP correlations to model the problem interactions. The EDA stochastically searches the constrained space of feasible solutions. It is evaluated across HapMap reference panel data sets.
Results: The EDA was compared with a SAT solver, able to find the single-marker minimal tagging sets, and with the Tagger program. The percentage of reduction ranged from 10% to 43% in the number of tagging SNPs of the minimal multi-marker tagging set found by the EDA with respect to the other algorithms.
Conclusions: The introduced algorithm is effective for the identification of minimal multi-marker SNP sets, which considerably reduce the dimension of the tagging SNP set in comparison with single-marker sets. Other variants of the SNP problem can be treated following the same approach.

1. Introduction

Disease-gene association consists of the identification of DNA variations which are highly associated with a known disease. The task can be accomplished by statistical genetic variation analysis of single nucleotide polymorphisms (SNPs). The study of complex disease in association studies requires the analysis of more than one locus because single locus methods cannot be used to identify complex patterns. They miss the genetic contribution to the disease of the interactions between loci [1,2]. Therefore, the analysis of multiple sites is required for better disease-gene association studies. Usually, this type of analysis involves genome wide association studies, where the whole genome is searched for the identification of genetic associations with observable traits [3–5].

Nevertheless, genotyping is complicated and very costly when a large number of candidate SNPs is considered. A possible remedy for this problem is the identification of a subset of representative SNPs or tagging SNPs that allows to reduce the genotyping overhead. In this way, frequency differences between case and control populations do not need to be measured in all SNPs but only in the subset of tagging SNPs. To this end, more precise mapping of the patterns of linkage disequilibrium is needed. Improved haplotype mapping of the human genome is an important step in this direction [4,5]. The other requirement is the conception of efficient procedures for appropriate selection of tagging SNPs.

The problem of determining a subset of SNPs to genotype from which to recover not genotyped SNPs, involves two different issues. The first one is the problem of selecting (tagging) SNPs. The second one is the problem of predicting the value of unknown or not genotyped SNPs from the ones available. The difference between these two problems has been previously emphasized [6]. In this work we focus on the first problem which is usually formulated as the objective of selecting the lowest number of tagging SNPs so that the remaining (tagged) SNPs are "covered". Covering is defined by some statistical criterion (e.g. a high correlation between tagging and tagged SNPs, informativeness measures, etc.). There are two main variants of this problem: When *single marker* SNPs are used, each tagged SNP can be covered by a single tagging SNP. When *multi-marker* tags are used, each SNP can be covered by a single SNP or by a subset of tagging SNPs. Multi-

* Corresponding author. Tel.: +34 913363675; fax: +34 943219306.
E-mail address: roberto.santana@upm.es (R. Santana).

0933-3657/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2010.05.010

marker tags can significantly outperform tagging efficiency with respect to single-marker approaches [7]. However, in the general case, the single and multi-marker SNP tagging problems are NP-complete [8].

Minimal tagging SNP selection has been mainly focused on single-marker tagging sets [8–11].

In multi-marker tagging set, some work has been reported: Bakker's Tagger tag SNP selection algorithm [7], available in Haploview [12], combines the simplicity of pairwise tagging methods with the efficiency benefits of multimarker haplotype approaches. The search is carried out trying to replace each tag of the original solution with a specific multi-marker predictor (on the basis of the remaining tags) to improve efficiency. The result of this greedy approach will depend mostly on the closeness of the initial single-marker tagging set to the optimal multi-marker set. Therefore, the algorithm is likely to get stuck in local optimal solutions.

Choi et al. [9] approach the minimal single-maker tagging SNP selection problem as an instance of the satisfiability (SAT) problem [13]. The optimal tagging set is obtained by enumerating the solutions to the SAT problem. Although the SAT approach allows to obtain optimal solutions for the single-marker tag problem, the number of SAT clauses exponentially increases for the multi-marker tag problem and the satisfiability approach does not seem to be applicable in this case.

Probabilistic graphical models, and in particular Bayesian networks, have been previously applied to haplotype block partitioning [14] and haplotype phasing [15]. There is also an application of Bayesian networks to the problem of tagging SNP selection [10], but the authors approach the problem from a different point of view.

In this paper we approach the search for a set of minimal multi-marker SNPs as an optimization problem. We focus on the problem of devising efficient methods to search the optimal solutions given a predefined quality measure. To address the problem, an estimation of distribution algorithm (EDA) [16–19] is employed. EDAs are evolutionary algorithms similar to genetic algorithms (GAs) [20,21] but where probabilistic modeling is used instead of genetic operators. EDAs allow to naturally incorporate a priori information about the problem. This information can dramatically improve the accuracy and efficiency of the search for optimal solutions. EDAs have been applied with excellent results to practical problems from several domains, including bioinformatics and biomedical problems [22,23].

The rest of the paper is organized as follows: In the next section, a number of basic biological concepts are introduced and the minimal tagging SNP set problem is presented. Section 3 introduces EDAs and explains the EDA approach to the minimal tagging SNP set problem. The experimental framework to evaluate our proposal is presented in Section 5, where the numerical results are analyzed. The conclusions of the paper and ideas for future work are presented in Section 6.

2. Motivation and description of the SNP tagging problem

In the human genome there are about 10 million sites where individuals differ by a single nucleotide. These sites are called single nucleotide polymorphisms (*SNPs*). An *allele* is an alternative form of a gene or SNP, or another type of variant. Most SNPs are *biallelic*, i.e. they appear as having only two possible nucleotides. A *haplotype* is a combination of alleles at multiple linked sites on a single chromosome, all of which are transmitted together. A *haplotype block* is a region containing strongly associated SNPs.

A chromosome carrying a particular allele of a given SNP has a high probability of carrying a particular allele of another SNP close to the first one. Thus, an allele frequency difference in the second SNP can manifest itself as an allele frequency difference in the first SNP. The non-random association of alleles at two or more sites on the same chromosome is called *linkage disequilibrium* (LD) and this relationship is often measured by the correlation coefficient r^2 between SNPs. A *tagging* or *tag* SNP is a representative SNP with high LD to other (*tagged*) SNPs.

Let D be a data set consisting of m haplotypes, $\mathcal{H} = \{h_1, \ldots, h_m\}$, each with n different SNPs, $S = \{s_1, \ldots, s_n\}$. The set D can be viewed as an $m \times n$ matrix. For simplicity of presentation, we assume in our analysis that each of the SNPs is biallelic. Let (A, a) and (B, b) respectively represent the two possible alleles for two different SNPs s_i and s_j in D. The correlation coefficient r^2_{ij} measures the similarity correlation between the two SNPs:

$$r^2_{ij} = \frac{(p_{AB}\, p_{ab} - p_{Ab}\, p_{aB})^2}{p_A\, p_B\, p_a\, p_b} \tag{1}$$

where p_{lk} ($l \in \{A, a\}$ and $k \in \{B, b\}$) denotes the frequency that l and k appeared together in the haplotypes of D and p_o ($o \in \{A, a, B, b\}$) denotes the frequency of o.

We say that SNP s_i tags SNP s_j if their correlation coefficient r^2_{ij} exceeds some threshold $r^2_{\min}$. We call $T_{\sin} \subseteq S$ a single-marker valid tag of S if $\forall\, s_j \in S,\ \exists\, s_i \in T_{\sin}$ such that $r^2_{ij} \geq r^2_{\min}$, i.e., for each SNP s_j in S there exists a SNP s_i in $T_{\sin}$ such that the correlation between both SNPs is higher than the threshold $r^2_{\min}$, and therefore it is expected that one SNP can be predicted from the other.

The correlation coefficient r^2_{ij} can be generalized to groups of SNPs. For example, the correlation coefficient between a couple of SNPs s_i and s_j with respective alleles (A, a) and (B, b) and a SNP s_k with possible alleles (C, c) can be computed as follows:

$$\begin{aligned} r^2_{\{i,j\},k} = \frac{1}{p_c(1-p_c)} \times \Bigg(& \frac{(p_{(a,b),c})^2}{(p_{(a,b),c} + p_{(a,b),C})} - p_{(a,b),c} \times p_c \\ & + \frac{(p_{(A,b),c})^2}{(p_{(A,b),c} + p_{(A,b),C})} - p_{(A,b),c} \times p_c + \frac{(p_{(a,B),c})^2}{(p_{(a,B),c} + p_{(a,B),C})} \\ & - p_{(a,B),c} \times p_c + \frac{(p_{(A,B),c})^2}{(p_{(A,B),c} + p_{(A,B),C})} - p_{(A,B),c} \times p_c \Bigg) \end{aligned}$$

where $p_{(l,o),t}$ ($l \in \{A, a\}$, $o \in \{B, b\}$ and $t \in \{C, c\}$) denotes the frequency that l and o, and t appeared together in the haplotypes of D and p_c denotes the frequency of allele c in SNP s_k.

We say that a subset $T_{\text{mul}} \subseteq S$ is a multi-marker valid tag of S if for all SNP s_i in S there exists a subset of SNPs $T^* \subseteq T_{\text{mul}}$ such that the correlation between the SNPs in T^* and s_i is higher than the threshold, i.e., $r^2_{T^*,i} > r^2_{\min}$.

The problem of finding the smallest single-marker tagging set is the problem of finding the smallest set $T_{\sin} \subseteq S$ that is a valid tag of S. Similarly, the problem of finding the smallest multi-marker tagging set is the problem of finding the smallest set $T_{\text{mul}} \subseteq S$ that is a valid multi-marker cover of S.

In this paper we focus on the second class of problems. We further constrain the set of multi-marker tagging sets to those where the tagging set of each SNP is formed by at most two tagging SNPs.

3. Estimation of distribution algorithms

The increasingly high computing power achievable from commodity computers has encouraged the design and implementation of non-trivial algorithms to solve different kinds of complex optimization problems. Some of these problems can be solved via an exhaustive search over the solution space, but in most cases this brute force approach is unaffordable. In these situations, deterministic or non-deterministic heuristic methods, which search inside the space of promising solutions, are often used. Some heuristic approaches are specifically designed to find good

solutions for a particular problem, but others are presented as a general framework adaptable to many different situations.

Among this second group are evolutionary algorithms such as genetic algorithms (GAs) [20,21] which have been widely used in the last decades. The main characteristic of these algorithms is that they use techniques inspired by the natural evolution of the species and find inspiration in concepts such as individuals, populations, breeding, fitness function, etc. At each step, evolutionary algorithms maintain a set of possible solutions to the problem at hand and generate a new set of solutions by mixing the current solutions.

Estimation of distribution algorithms (EDAs) [16–19] include a set of optimization approaches in the evolutionary computation field characterized by the use of explicit probability distributions. In EDAs, contrary to GAs, there are neither crossover nor mutation operators. Instead, the new population of individuals is sampled from a probability distribution, which is estimated from a database that contains the selected individuals from the current generation. Thus, the interrelations between the different variables of the problem that represent the individuals are explicitly expressed through the joint probability distribution associated with the individuals selected at each generation. This selection process, learning a probability distribution and sampling it, is repeated until a termination criterion is met. The termination criteria of an EDA can be a maximum number of generations, a homogeneous population or no improvement after a specified number of generations. As a selection operator researchers usually consider those commonly used in GAs. A general pseudo-code for all EDAs is described in Algorithm 1.

Algorithm 1. Estimation of distribution algorithm

```
1 Set t ⇐ 0. Generate M points randomly.
2 do {
3     Evaluate the points using the fitness function.
4     Select a set D_t^S of N ≤ M points according to a selection method.
5     Calculate a probabilistic model of D_t^S.
6     Generate M new points sampling from the distribution represented in
      the model.
7     t ⇐ t + 1
8 } until Termination criteria are met.
```

The most important step in EDAs is the learning of the probabilistic model. This fifth step has a significant influence on the behavior of the EDA from the point of view of complexity and performance. Therefore EDAs are usually classified into three groups, according to their ability to capture the dependencies between variables:

- *Without dependencies*: It is assumed that the n-dimensional joint probability distribution factorizes as a product of n univariate and independent probability distributions. Algorithms that use this model are, among others, univariate marginal distribution algorithm (UMDA) [18], compact genetic algorithm (cGA) [24] and population based incremental learning [25].
- *Bivariate dependencies*: Only the dependencies between pairs of variables are taken into account. This way, the process of estimating the joint probability can still be fast. This group includes: mutual information maximization for input clustering (MIMIC) [26], bivariate marginal distribution algorithm (BMDA) [27] and Tree-EDA [28].
- *Multiple dependencies*: Higher order dependencies between the variables are considered. In this group we can find algorithms like estimation of Bayesian networks algorithm (EBNA) [29], estimation of Gaussian networks algorithms (EGNAs) [30] and the Bayesian optimization algorithm (BOA) [31].

The algorithms in the first group deal with computationally easy to learn probabilistic models. However, given the strong independence assumption between the variables of the problem, they are sometimes unable to solve complex optimization problems. The second group of algorithms represent a balance between the computational cost of learning a probabilistic model and their expressive power. Finally, the third group includes the least restrictive models, being able to solve very complex problems but assuming the expensive cost of learning it.

For detailed information about the characteristics of these EDAs and other algorithms that take part of this family see [16,17,19].

4. The EDA approach to the SNP problem

In order to set our EDA approach to the problem of finding the minimal multi-marker tagging set, we start by establishing the search space and problem representation, and then the function to optimize. Finally, we will present the specific EDA used in the problem by describing the probabilistic model chosen and the learning and sampling algorithm, as well as the way in which a priori information about the problem is incorporated.

4.1. Search space and problem representation

Given a set S of n SNPs, the search space is composed of the valid multi-marker subset of S (note that as we pointed out in Section 2 we only consider tagging sets formed by at most two tagging SNPs). However there may exist SNPs that are not covered by any single or pair of tagging SNPs. The existence of SNPs that show almost no linkage disequilibrium with any other SNPs in the haplotype has been acknowledged as a feature that illustrates the full complexity of empirical patterns of genetic variation [4]. These SNPs can be only self-tagged, so we call them fixed SNPs. Given that fixed SNPs should appear in each valid multi-marker set of S we do not consider them in the search and therefore we need to carry out a search inside a set of n' SNPs.

We codify a possible solution to the problem (valid multi-maker set) as a binary n' vector $\mathbf{x} = (x_1, \ldots, x_{n'})$. Variable X_i will represent whether the ith SNP is part of the tagging set ($x_i = 1$), or it is tagged ($x_i = 0$).

The final solution comprises all fixed SNPs and those found during the search.

4.2. Fitness function

For implementational reasons, the minimization of the number of tagging SNPs is transformed in the maximization of Eq. (2), where each solution $\mathbf{x}$ satisfies that all the non-tagging SNPs are covered by another single or pair of tagging SNPs. Hence given a solution $\mathbf{x}$, $f(\mathbf{x})$ gives back n' minus the number of SNPs in the valid multi-marker set.

$$f(\mathbf{x}) = n' - \sum_{i=1}^{n'} x_i \tag{2}$$

4.3. Tree-based EDA approach

4.3.1. Probabilistic model

The EDA of choice uses a probabilistic model that captures bivariate dependencies between the variables. This probabilistic model is based on a tree structure where each variable may depend on, at most, another variable, which is called the parent. A probability distribution $p_{Tree}(\mathbf{x})$ that is conformal with a tree is defined as:

$$p_{Tree}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | pa(x_i)) \tag{3}$$

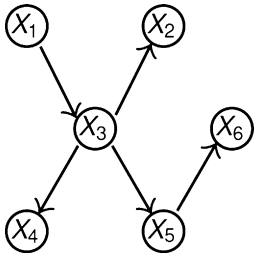

Fig. 1. Tree structure between six variables and its associated factorization.

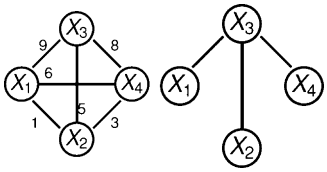

Fig. 2. Example of the application of the structure learning algorithm: (a) initial graph with the mutual information values and (b) the resultant structure of the probabilistic model.

where $Pa(X_i)$ is the parent of X_i in the tree, and $p(x_i|\,pa(x_i)) = p(x_i)$ when $Pa(X_i) = \varnothing$, i.e. X_i is the root of the tree. The distribution $p_{Tree}(\mathbf{x})$ itself will be called a tree model when no confusion is possible. Probabilistic trees are represented by directed acyclic graphs. An example of a tree over six variables and its associated factorization can be seen in Fig. 1.

There are two main reasons behind the choice of this model. The first is efficiency. The computation of the bivariate statistics needed to compute a tree is less expensive than the structural learning procedure required to construct more complex models such as general Bayesian networks [32]. This efficiency factor is particularly relevant when the number of variables increases. The second reason in the choice of the model is that pairwise interactions between the variables represent an important contribution to the fitness function of the minimal tagging SNP set problem.

4.3.2. Learning and sampling algorithms

The construction of the tree structure from data implies the detection of the most important bivariate interactions between the variables. This can be done applying statistical independence tests [27] or methods based on the analysis of the mutual information between variables [33]. We follow the second approach as shown in Algorithm 2.

Algorithm 2. Tree-EDA

1 $D_0 \leftarrow$ Generate M individuals randomly
2 $l = 1$
3 **do** {
4 $D^s_{l-1} \leftarrow$ Select $N \leq M$ individuals from D_{l-1} according to a selection method
5 Compute the univariate and bivariate marginal frequencies $p^s_i(x_i|D^s_{l-1})$ and $p^s_{i,j}(x_i, x_j|D^s_{l-1})$ of D^s_{l-1}
6 Calculate the matrix of mutual information using bivariate and univariate marginals.
7 Calculate the maximum weight spanning tree from the matrix of mutual information.
8 Compute the parameters of the model.
9 $D_l \leftarrow$ Sample M individuals (the new population) from the tree and add elitist solutions.
10 } **until** A stop criterion is met

Initially, the univariate and bivariate probabilities are respectively calculated for every variable and pair of variables. To determine the marginal probabilities, we compute, from the set of selected solutions, the frequencies corresponding to each marginal configuration. In our binary representation, this corresponds to 2 univariate (each variable takes 2 values) and 4 bivariate (the two values corresponding to the child and the two values for its parent) frequency values, for n variables and $n(n-1)/2$ pairs of variables. Frequencies are normalized in order to obtain the probabilities. From these marginal probabilities, the mutual information between each pair (X_i, X_j) of variables is computed:

$$I(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)\, p(x_j)}$$

To construct the tree structure, an algorithm introduced in [34], that calculates the maximum weight spanning tree from the matrix of mutual information between pairs of variables is used. An example of the application of the algorithm to a problem with four variables can be consulted in Fig. 2.

Probabilistic logic sampling [35] is applied to sample new solutions from the tree. New solutions are generated by sampling, for each tree, firstly the root, and subsequently each variable conditioned by its parent. The value of a root variable is chosen by randomly selecting one of its two configurations proportionally to its univariate probability. Similarly, the value of a children in the tree is randomly selected proportionally to its conditional probability values conditioned in the value already assigned to its parent.

Finally, the new sampled solutions are combined with the set of best solutions (elitist solutions) selected from the previous iteration.

4.3.3. Using the problem structure to increase the EDA efficiency

It is a common practice in EDAs to use available information about the problem to improve the efficiency of the learning and sampling steps of the algorithms. This can be achieved in a variety of ways:

- Using the known structural information to define a factorization of the probabilistic model [36,37].
- Constraining the set of interactions to be included in the probabilistic model [38,39].
- Specifying soft constraints to bias the construction of the probabilistic model [40,41].

In the problem under consideration, there is information about the correlations between the SNPs that can be incorporated to the model using the second of the previous approaches.

Step 6 of the tree learning algorithm (see Algorithm 2) calculates the mutual information between each pair of variables of the individuals X_i and X_j (note that each variable X_i makes reference to SNP s_i), for calculating a tree structure between the variables in the next step. However, it is possible that for two SNPs s_i and s_j there is not a tag relationship between them. Therefore it does not seem to make sense to consider this possible relation in the probabilistic model. This a priori information can be incorporated in the learning algorithm: our proposal only considers the mutual information between two variables X_i and X_j if their corresponding SNPs, s_i and s_j are involved in a tagging relationship, i.e. they belong to a pair (tagging-tagged) or to a triple (tagging,tagging,tagged) of SNPs. This new algorithm reduces the computational time of the learning step in the EDA and also improves the reached solutions.

In a preprocessing stage, the set of pairs and triples that have a potential type of tagging relationship are computed using the parameters set by the user (e.g. maximum distance sequence, correlation coefficient threshold, etc.). These subsets will be the input of the minimum multi-marker subset search algorithm. They can also be employed to construct an interaction graph that reflects

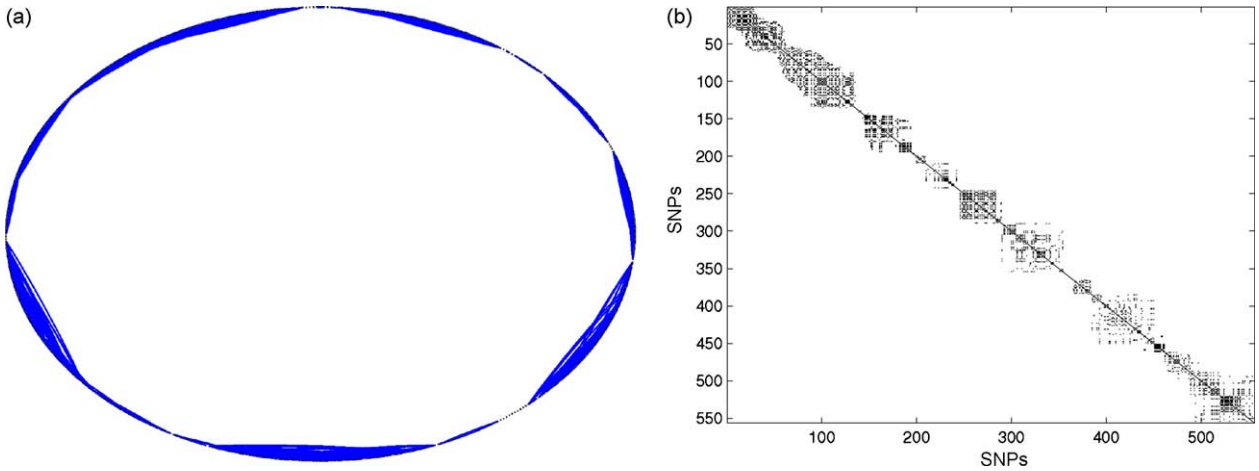

Fig. 3. Representation of the interactions between the SNPs in the ENm010.CEU HapMap Encode region. Single tagging SNPs are represented in the graph. (a) Interaction graph. (b) Adjacency matrix.

the structure of the interactions between tagging and tagged SNPs and which serves as a convenient representation to illustrate the type of structural information used by the optimization algorithm. In the case of single marker SNPs, the interaction graph is constructed by mapping one vertex to each SNP and an edge in the graph represents that the r^2 between the corresponding SNPs is above the threshold [9]. The structure of interactions represented by this graph can also be displayed using the adjacency matrix. Fig. 3(a) shows the interaction graph for SNPs in the ENm010.CEU HapMap Encode region [4]. The 556 SNPs are positioned in a circle following the order of the sequence. Fig. 3(b) shows the corresponding adjacency matrix where interactions between proximal SNPs can be also identified.

When multi-marker SNPs are considered, the graph representation is not straightforward because it might be necessary to distinguish whether a tagged SNP is covered by a single SNP or by a pair of tagging SNPs. As regards the analysis that will follow, this distinction is not relevant and therefore, when a SNP is tagged by a pair, there will be an edge between the tagging SNP and each of the tagged SNPs. Fig. 4(a) shows the interaction graph for SNPs in the ENm010.CEU HapMap Encode region when single and pairs of tagging SNPs are represented in the graph. Fig. 4(b) shows the corresponding adjacency matrix.

Fixed SNPs can be identified as disconnected nodes in an interaction graph.

Constraining the set of interactions to be included in the probabilistic model helps to reduce the number of spurious correlations that arise between variables during the search. Generally, the spurious correlations learned during the learning step may contribute to deteriorate the accuracy of the models in the representation of the selected solutions, and negatively influence the efficiency of the search.

The computational complexity of EDAs is mainly dependent on the complexity of the learning algorithm, but it also depends on the population size and number of generations needed for convergence, which are both problem-dependent. The computational complexity of Tree-EDA is quadratic. Nevertheless, the use of a priori information about the problem structure, drastically reduces the time spent to learn the probabilistic model [39,42].

4.3.4. Repairing procedure

It must be taken into account that not all the sampled solutions are feasible, in the sense that there are binary vectors that represent situations in which one or more SNPs could be not covered. To keep the search in the space of feasible solutions, we implement a repairing procedure that enforces the solutions

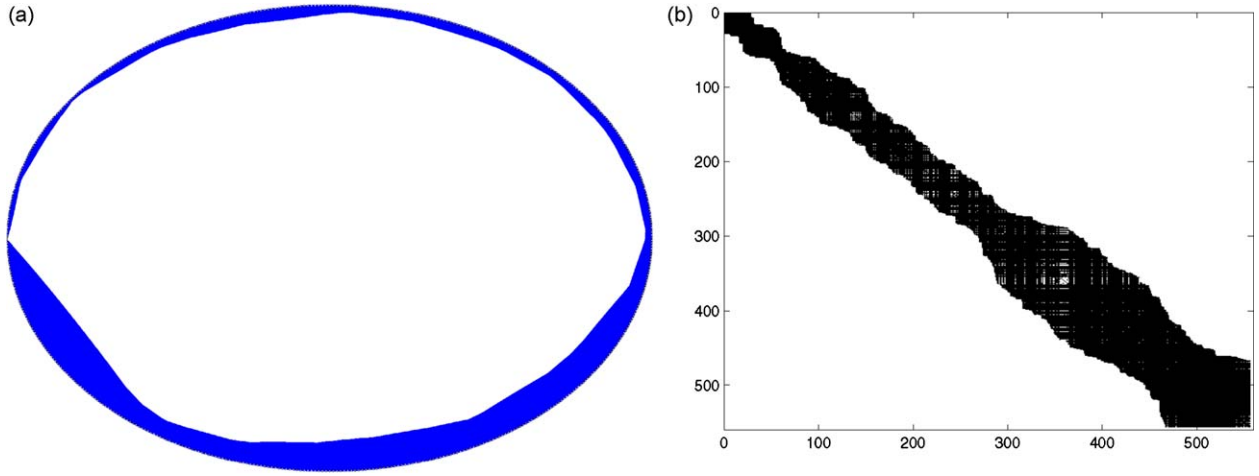

Fig. 4. Representation of the interactions between the SNPs in the ENm010.CEU HapMap Encode region. Single and pairs of tagging SNPs are represented in the graph. (a) Interaction graph. (b) Adjacency matrix.

feasibility. This procedure is applied during the evaluation step. It is described in Algorithm 3.

Algorithm 3. Repairing and evaluation procedure

```
1  Compute the set C_p of all SNPs not tagged in the current solution by a single tagging SNP
2  If C_p = ∅ output f(x) and exit
3  do {
4     Choose randomly SNP i from C_p
5     If the set of single tagging SNPs that can potentially tag i is not empty
6        Randomly select a SNP j that belongs to this set
7     Elseif the set formed by all SNP pairs that potentially tag i, where one of the two SNPs is already a tagging SNP in the solution, is not empty
8        Randomly select a pair (j, k) that belongs to this set, where k is the tagging SNP which is already in the current solution
9     Else
10       Randomly choose a pair of SNPs (j, k) that can tag i
11    Set j or j and k, as tagging SNPs
12    Remove j and all the SNPs tagged by j or by (j, k) from C_p
13 } until C_p = ∅
14 Output f(x), x
```

Algorithm 3 starts by checking whether **x** is a feasible solution. For efficiency reasons, the verification is carried out by firstly taking into account the single tagging SNPs and then the pairs of tagging SNPs. If the set of non-tagged SNPs is not empty (i.e. the solution is unfeasible), each of the non-tagged SNPs becomes tagged by transforming some of them into tagging SNPs (x_i from 0 to 1). The repairing procedure is conceived to set as few tagging SNPs as possible. It finishes when all the SNPs are tagged.

5. Experiments

First, we introduce the SNP reference panel and the parameters used by Tree-EDA. Then, we explain how the experiments were designed. Finally, the numerical results of the experiments are presented.

5.1. Experimental setup

5.1.1. Description of the SNP problem benchmark

To evaluate the introduced algorithms, we used the HapMap reference panel [4]. As done in a previous work [9], samples over the ENCODE regions are used for the experiments. These data, from 270 individuals from four populations (people of European ancestry [CEU], Yoruba of Ibadan, Nigeria [YRI], Han Chinese [CHB], and Japanese [JPT]) are made up of polymorphisms over 10 genomic regions spanning a total 5 Mb of the sequence. These regions have been carefully studied and are believed to have complete ascertainment for SNPs with frequency higher than 5%.

Table 1 shows the details of 40 SNP problem instances used as benchmark for evaluating the algorithms. In the table, name refers to the HapMap region and population, n is the total number of SNPs, n' is the number of SNPs that can be tagged by another SNP or pair of SNPs (the rest of SNPs are fixed since they can be only self-tagged), nPairs is the number of pairs of SNPs above the correlation threshold and similarly, nTriples is the number of triples such that the correlation of the tagged SNP given a pair of tagging SNPs is above the correlation threshold.

Table 1
Details of the SNP problem benchmark.

Name	n	n'	nPairs	nTriples
ENm010.CEU	556	502	2716	796,782
ENm010.CHB	433	381	3324	909,938
ENm010.JPT	441	406	2711	658,370
ENm010.YRI	630	502	1561	476,061
ENm013.CEU	745	711	7294	3,385,226
ENm013.CHB	635	594	5907	2,324,625
ENm013.JPT	636	595	6392	2,497,352
ENm013.YRI	792	726	3524	1,471,646
ENm014.CEU	895	851	7918	4,164,642
ENm014.CHB	643	601	6324	2,187,202
ENm014.JPT	561	512	5232	1,709,585
ENm014.YRI	951	870	4947	2,548,920
ENr112.CEU	922	873	9215	5,808,697
ENr112.CHB	1015	976	11,330	7,680,843
ENr112.JPT	997	955	7870	5,384,780
ENr112.YRI	1298	1192	5712	4,332,098
ENr113.CEU	1054	1004	14,535	10,133,619
ENr113.CHB	903	864	16,384	9,224,261
ENr113.JPT	829	793	15,262	7,233,779
ENr113.YRI	1135	1026	5478	3,301,762
ENr123.CEU	934	886	6550	4,253,145
ENr123.CHB	881	763	9331	5,680,830
ENr123.JPT	836	687	5746	2,993,247
ENr123.YRI	904	834	5523	3,120,904
ENr131.CEU	1026	957	7617	5,137,622
ENr131.CHB	1018	920	7290	4,394,708
ENr131.JPT	993	893	7367	4,317,602
ENr131.YRI	1137	951	5174	3,228,853
ENr213.CEU	648	616	5635	2,183,354
ENr213.CHB	519	494	5354	1,478,774
ENr213.JPT	562	529	5250	1,759,928
ENr213.YRI	846	722	3979	1,642,158
ENr232.CEU	521	454	4644	1,377,633
ENr232.CHB	596	516	3406	1,126,619
ENr232.JPT	573	496	3188	1,076,740
ENr232.YRI	724	532	1986	634,306
ENr321.CEU	594	550	5082	1,808,674
ENr321.CHB	695	647	6332	2,705,717
ENr321.JPT	682	621	5317	2,282,908
ENr321.YRI	981	856	3579	1,820,721

5.1.2. Preprocess

Given a data set D consisting of m haplotypes, first we compute the r^2_{ij} for each pair of SNPs s_i and s_j. Those SNPs for which the frequency of the most probable allele is above 0.95 are not considered. Then $r^2_{\{i,j\},k}$ is computed for $i \neq j \neq k$. Only pairs of SNPs that are in the sequence at a distance lower than $d = 40,000$ are considered. The resulting set of all initial pairs and triples is reduced by eliminating those subsets of SNPs with an r^2 below the minimum threshold $r^2_{\min} = 0.8$.

5.1.3. Parameters of the algorithms

Tree-EDA, as other approaches based on EDAs, has a set of parameters to be selected. In this work we have chosen some default values based on our experience, without looking exhaustively for the best combination. The population size was set to 5000 and the number of generations was set to 100. Truncation selection with parameter $T = 15(\%)$ was employed. In this selection scheme, the best $T{*}N$ individuals of the population are selected to construct the probabilistic model. We apply a replacement strategy called best elitism in which the selected population at generation t is incorporated into the population of generation $t + 1$, keeping the best individuals found so far and avoiding to reevaluate their fitness function. The algorithm will stop when the maximum number of generations is reached or the selected population has become too homogeneous (no more than 10 different individuals).

5.1.4. Design of the experiments

The main goal of the experiments was to determine whether the consideration of pairs of tagging SNPs can improve the results achieved when only single tagging SNPs are used. Tree-EDA is used to optimize the objective function that measures the number of

Table 2
Results achieved by SAT Tagger, Tagger and Tree-EDA for the 40 SNP problem instances.

Name	SAT Tagger	Tagger			Tree-EDA			
		Pairwise	aggr-2M	aggr-2M-3M	Best	nbest	Mean	Worst
ENm010.CEU	159	161	126	126	101	2	102.6	104
ENm010.CHB	99	100	92	92	81	4	82.1	84
ENm010.JPT	104	104	89	89	69	1	70.5	72
ENm010.YRI	301	301	247	249	201	5	201.8	204
ENm013.CEU	113	118	98	103	79	1	82.6	85
ENm013.CHB	103	105	92	93	71	2	72.4	74
ENm013.JPT	101	103	91	91	76	2	78.4	82
ENm013.YRI	235	239	199	196	154	1	156.8	160
ENm014.CEU	167	169	141	143	126	2	128.8	131
ENm014.CHB	122	125	106	108	91	2	93.2	96
ENm014.JPT	121	122	106	106	87	1	90.9	93
ENm014.YRI	269	270	223	226	183	1	186.9	190
ENr112.CEU	181	185	141	144	116	1	118.1	120
ENr112.CHB	165	167	137	134	110	1	113.8	116
ENr112.JPT	190	193	156	157	121	1	126.9	132
ENr112.YRI	449	452	340	345	256	2	260.0	265
ENr113.CEU	183	185	146	153	121	1	123.8	125
ENr113.CHB	109	112	92	94	73	2	75.1	77
ENr113.JPT	105	108	87	89	69	3	70.6	72
ENr113.YRI	365	366	289	288	229	1	233.1	237
ENr123.CEU	196	199	161	158	132	1	134.5	140
ENr123.CHB	248	249	222	223	197	1	199.8	203
ENr123.JPT	288	289	257	256	227	1	229.6	234
ENr123.YRI	255	260	208	211	162	1	165.0	166
ENr131.CEU	225	229	175	175	150	2	152.2	154
ENr131.CHB	268	270	230	229	178	1	182.1	186
ENr131.JPT	260	260	223	224	176	3	177.8	181
ENr131.YRI	467	467	374	375	298	2	300.6	304
ENr213.CEU	128	133	107	108	81	2	83.3	86
ENr213.CHB	100	101	80	81	65	4	65.8	67
ENr213.JPT	110	111	95	95	76	1	78.1	80
ENr213.YRI	328	328	267	270	208	1	212.5	216
ENr232.CEU	138	139	124	125	106	2	107.4	109
ENr232.CHB	199	199	166	167	131	1	132.7	135
ENr232.JPT	194	195	166	169	132	1	135.5	140
ENr232.YRI	401	402	345	343	277	2	278.6	280
ENr321.CEU	132	133	112	111	87	1	89.2	91
ENr321.CHB	158	158	130	131	99	1	101.0	103
ENr321.JPT	164	167	134	134	108	1	109.8	111
ENr321.YRI	364	365	283	287	226	1	230.7	234

tagging SNPs. Since EDAs are stochastic methods, we conduct for each SNP problem a set of experiments and extract statistical information from the analysis of these experiments. The performance of Tree-EDA was evaluated considering the fitness of the best, average, and worst solutions found in all the experiments. The number of experiments conducted for each instance was 10.

5.2. Numerical results

Using the SNP problem benchmark, we compare the quality of the solutions obtained by Tree-EDA to the solutions obtained by SAT tagger [9], and three variants of the Bakker's Tagger tag SNP selection algorithm [7], available in Haploview [12]. The first

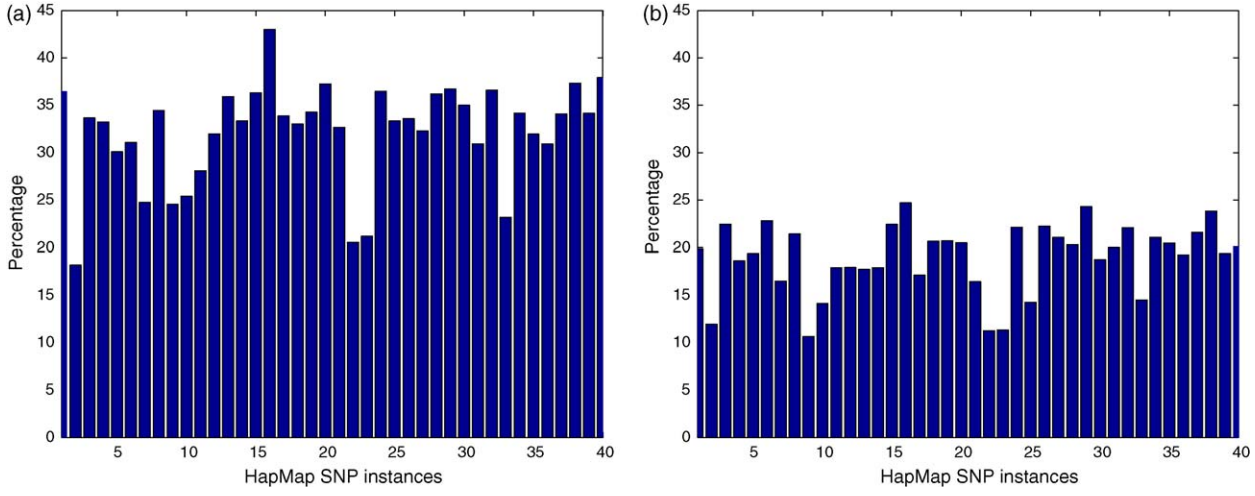

Fig. 5. Percentage reduction in the number of tagging SNPs of the minimal multi-marker tagging set found by Tree-EDA with respect: (a) To the single-marker minimal tagging set. (b) To the multi-marker minimal tagging set found by Tagger.

algorithm guarantees to find the best possible solution obtained when only a single tagging SNP is used. The second algorithm is one of the state-of-the-art algorithms of its kind and allows single and multi-tagging SNPs.

Table 2 shows the results achieved using the SAT tagger [9], the best results achieved by Tagger using single SNP tagging (pairwise), pair SNP tagging (aggr-2M), and two or three SNP tagging sets (aggr-2M-3M). Regarding Tree-EDA, the following results are presented: best solution, the number of times that the best solution was achieved (nbest), and the average (mean) and worst (worst) values of the solutions.

An analysis of the table reveals that the solutions obtained in all the experiments by Tree-EDA are always better than the minimal tagging sets provided by the rest of the algorithms. Fig. 5 shows the percentage of reduction (ranging from 10% to 43%) in the number of tagging SNPs of the minimal multi-marker tagging set found by Tree-EDA with respect to the single-marker minimal tagging set and to the best solution between the three variants of the Tagger algorithm. In addition, regarding the information loss of our approach, we must point out that this is similar (or even a bit lower) than that suffered by Tagger, with values of R^2 ranging between 0.943 and 0.982.

6. Conclusions and future work

We have presented an optimization approach, Tree-EDA,[1] for finding the minimal set of multi-marker tagging SNPs. The optimization problem was dealt by using an estimation of distribution algorithm. The obtained solutions considerably improved those achieved by exact algorithms for the single-marker tagging SNP problem and state-of-the-art multi-marker tagging SNPs.

The approach introduced in this paper shares a number of suitable characteristics with other evolutionary algorithms: by using a population of solutions it allows a better exploration of the search space and avoids getting stuck in local optima. In addition, the fact of being a stochastic algorithm allows to obtain different solutions in different runs.

The EDA we have applied exhibits other particular features that explain its success for computing the minimal set of multi-marker tagging SNPs: (1) It incorporates structural information about the problem into the search; (2) It takes advantage of probabilistic modeling of the promising solutions to efficiently sample the solution space. These features are also advantages over traditional GAs and other evolutionary algorithms.

Another virtue of the introduced approach is that it can be adapted to similar problems with minor modifications. We briefly review some of the possibilities for future work.

6.1. Future work to improve the results of the minimal tagging problem

The EDA used in our experiments starts from a randomly generated population of solutions. However, incorporating knowledge about the problem in the starting population can improve the results of the algorithm. We could first rank the SNPs according to the number of SNPs they can potentially tag [7], and then generating initial populations prioritizing solutions that contain better ranking SNPs.

It is an open question to investigate whether better solutions of the minimum SNP tagging set can be obtained by increasing the complexity of the models used by EDAs. Two direct extensions of EDAs based on trees that could be tried are EDAs that use mixtures of trees [28] and polytrees [43].

Different approaches can be used as a basis to devise local optimization methods to be combined with EDAs. The solutions obtained by the EDA can be improved by trying to remove redundant tagging SNPs by keeping the covering of all tagged SNPs. The interaction graph could be used to implement this type of local optimization methods.

6.2. Future work to extend the applications of EDAs to similar SNP problems

The optimization approach we have followed is based on the existence of haplotype blocks. Although recent results have led to more accurate estimation of haplotype blocks [4], it does not appear to be possible to unambiguously and uniquely infer the true block partitioning [8]. These blocks are capturing general regions of low diversity, but the boundaries between them are not rigorously defined. In addition, common haplotypes capture most of the genetic variation across sizable regions, in particular haplotype blocks, but there is substantial linkage disequilibrium between adjacent blocks [44]. An open question is how to select a minimum informative subset of SNPs without partitioning the SNPs into blocks. This is achieved by other algorithms [8]. It is an interesting question to investigate whether our optimization approach can be applied without requiring the block partitioning, or by increasing the distance threshold currently imposed to potential correlations between SNPs. Parallel and distributed EDAs schemes [45,46] could be an interesting alternative in this case.

The problem of finding the minimal tagging SNP set can be generalized to consider which the maximum number of SNPs that can be tagged with k tagging SNPs is. The k tagging SNP problem can be approached as a problem with constraints, where all solutions are forced to have exactly k tagging SNPs (i.e. in our codification, binary solutions with exactly k ones).

Another approach is to redefine it as a multi-objective problem with two objectives: Minimize k and maximize the number of SNPs tagged. This way, a solution $\mathbf{x}$ with a given value of $(k(\mathbf{x}), f(\mathbf{x}))$ will be dominated only by solutions that tag more SNPs with fewer tagging SNPs. The Pareto set approximation will give an idea of the gain in the number of SNPs tagged as a result of increasing the number of tagged SNPs. The Tree-EDA algorithm can be adapted to deal with multi-objective problems by modifying the selection step to include a Pareto-set approximation.

Acknowledgements

The authors want to thank Buhm Han for the support provided. This work has been partially supported by the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2008-06815-C02-01,TIN2007-62626, the CajalBlueBrain project, and Consolider Ingenio 2010 - CSD2007-00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

References

[1] Goodman JE, Mechanic LE, Luke BT, Ambs S, Chanock S, Harris CC. Exploring SNP-SNP interactions and colon cancer risk using polymorphism interaction analysis. Journal of Cancer 2006;118(7):1790–7.
[2] Mechanic LE, Luke BT, Goodman JE, Chanock S, Harris CC. Polymorphism interaction analysis (PIA): a method for investigating complex gene-gene interactions. BMC Bioinformatics 2008;9(146):1790–7.
[3] Heidema AG, Boer JMA, Nagelkerke N, Mariman ECM, van der A DL, Feskens EJM. The challenge for genetic epidemiologists: how to analyze large numbers of SNPs in relation to complex diseases. BMC Genetics 2006;7(23).
[4] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million snps. Nature 2007;449(7164):851–61.

[1] Tree-EDA software is available at http://www.sc.ehu.es/ccwbayes/software/EDATagger.html.

[5] Manolio TA, Brooks LD, Collins FS. A HapMap harvest of insights into the genetics of common disease. The Journal of Clinical Investigation 2008;118(5):1590–605.
[6] Montpetit A, Nelis M, Laflamme P, Magi R, Ke X, Remm M, et al. An evaluation of the performance of tag SNPs derived from HapMap in a Caucasian population. PLoS Genetics 2007;2(3):e27.
[7] de Bakker PIW, Yelensky R, Pe'er I, Gabriel SB, Daly MJ, Altshuler D. Efficiency and power in genetic association studies. Nature Genetics 2005;37:1217–23.
[8] Bafna V, Halldorsson BV, Schwartz R, Clark AG, Istrail S. Haplotypes and informative SNP selection algorithms: don't block out information. In: Proceedings of the seventh annual international conference on research in computational molecular biology RECOMB'03. New York, NY, USA: ACM; 2003. p. 19–27.
[9] Choi A, Zaitlen N, Han B, Pipatsrisawat K, Darwiche A, Eskin E. Efficient genome wide tagging by reduction to SAT. In: Crandall KA, Lagergren J, editors. Proceedings of the 8th International Workshop Algorithms in Bioinformatics WABI-2008, volume 5251 of Lectures Notes in Bioinformatics. Heidelberg: Springer; 2008. p. 135–47.
[10] Lee PH, Shatkay H. Bntagger: improved tagging SNP selection using Bayesian networks. Bioinformatics 2006;22(14):e211–9.
[11] Phuong TM, Lin Z, Altman RB. Choosing SNPs using feature selection. Journal of Bioinformatics and Computational Biology 2006;4(2):241–57.
[12] Barrett JC, Fry B, Maller J, Daly MJ. Haploview: analysis and visualization of ld and haplotype maps. Bioinformatics 2005;21(2):263–5.
[13] Selman B, Levesque H, Mitchell D. A new method for solving hard satisfiability problems. In: CORPORATE American Association for Artificial Intelligence, editor. Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92), San Jose, CA, USA, 1992. Menlo Park, CA, USA: American Association for Artificial Intelligence. p. 440–6.
[14] Greenspan G, Geiger D. High density linkage disequilibrium mapping using models of haplotype block variation. Bioinformatics 2004;20(Suppl 1):i137–44.
[15] Xing EP, Sharan R, Jordan MI. Bayesian haplo-type inference via the Dirichlet process. In: Proceedings of the twenty-first international conference on Machine learning (ICML-04). New York, NY, USA: ACM; 2004. p. 879–86.
[16] Larrañaga P, Lozano JA, editors. Estimation of distribution algorithms. A new tool for evolutionary computation. Boston/Dordrecht/London: Kluwer Academic Publishers; 2002.
[17] Lozano JA, Larrañaga P, Inza I, Bengoetxea E, editors. Towards a new evolutionary computation: advances on estimation of distribution algorithms. Springer; 2006.
[18] Mühlenbein H, Paaß G. From recombination of genes to the estimation of distributions I. Binary parameters. In: Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-P, editors. Parallel Problem Solving from Nature—PPSN IV, volume 1141 of Lectures Notes in Computer Science, Berlin. Springer; 1996. p. 178–87.
[19] Pelikan M, Sastry K, Cantú-Paz E, editors. Scalable optimization via probabilistic modeling: from algorithms to applications. Studies in computational intelligence. Springer; 2006.
[20] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley; 1989.
[21] Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. Ann Arbor, MI: University of Michigan Press; 1975.
[22] Armañanzas R, Inza I, Santana R, Saeys Y, Flores JL, Lozano JA, et al. A review of estimation of distribution algorithms in bioinformatics. BioData Mining 2008;1(6). doi:10.1186/1756-0381-1-6.
[23] Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, et al. Machine learning in bioinformatics. Briefings in Bioinformatics 2006;7:86–112.
[24] Harik GR, Lobo FG, Goldberg DE. The compact genetic algorithm. IEEE Transactions on Evolutionary Computation 1999;3(4):287–97.
[25] Baluja S. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA; 1994.
[26] De Bonet JS, Isbell CL, Viola P. MIMIC: finding optima by estimating probability densities. In: Mozer MC, Jordan MI, Petsche T, editors. Advances in neural information processing systems, vol. 9. Cambridge: The MIT Press; 1997. p. 424–30.
[27] Pelikan M, Mühlenbein H. The bivariate marginal distribution algorithm. In: Roy R, Furuhashi T, Chawdhry PK, editors. Advances in soft computing—engineering design and manufacturing. London: Springer; 1999. p. 521–35. ISBN 1-85233-062-7.
[28] Santana R, Ochoa A, Soto MR. The mixture of trees factorized distribution algorithm. In: Spector L, Goodman E, Wu A, Langdon WB, Voigt HM, Gen M, Sen S, Dorigo M, Pezeshk S, Garzon M, Burke E, editors. Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001. San Francisco, CA: Morgan Kaufmann Publishers; 2001. p. 543–50.
[29] Etxeberria R, Larrañaga P. Global optimization using Bayesian networks. In: Ochoa A, Soto MR, Santana R, editors. Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99). Havana, Cuba: Editorial Academia; 1999. p. 151–73. ISBN 959-02-024101.
[30] Larrañaga P, Etxeberria R, Lozano JA, Peña JM. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of the Basque Country; 1999.
[31] Pelikan M. Hierarchical Bayesian Optimization Algorithm. Toward a New Generation of Evolutionary Algorithms, volume 170 of Studies in Fuzziness and Soft Computing. Springer; 2005.
[32] Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. San Mateo, California: Morgan Kaufmann; 1988.
[33] Baluja S, Davies S. Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space. In: Fisher DH, editor. Proceedings of the 14th International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann; 1997. p. 30–8.
[34] Chow CK, Liu CN. Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 1968;14(3): 462–7.
[35] Henrion M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Lemmer JF, Kanal LN, editors. Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence. Elsevier; 1988. p. 149–64.
[36] Mühlenbein H, Mahnig T, Ochoa A. Schemata, distributions and graphical models in evolutionary optimization. Journal of Heuristics 1999;5(2):213–47.
[37] Ochoa A, Soto MR, Santana R, Madera J, Jorge N. The factorized distribution algorithm and the junction tree: a learning perspective. In: Ochoa A, Soto MR, Santana R, editors. Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99), Havana, Cuba, March 1999. Havana, Cuba: Editorial Academia. p. 368–77. ISBN 959-02-024101.
[38] Baluja S. Incorporating a priori knowledge in probabilistic-model based optimization. In: Pelikan M, Sastry K, Cantú-Paz E, editors. Scalable optimization via probabilistic modeling: from algorithms to applications, studies in computational intelligence. Springer; 2006. p. 205–22.
[39] Santana R, Larrañaga P, Lozano JA. The role of a priori information in the minimization of contact potentials by means of estimation of distribution algorithms. In: Marchiori E, Moore JH, Rajapakse JC, editors. Proceedings of the Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, volume 4447 of Lecture Notes in Computer Science. Valencia, Spain: Springer; 2007. p. 247–57.
[40] Hauschild M, Pelikan M. Enhancing efficiency of hierarchical BOA via distance-based model restrictions. MEDAL Report No. 2008007, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL); April 2008.
[41] Hauschild M, Pelikan M, Sastry K, Goldberg DE. Using previous models to bias structural learning in the hierarchical BOA. MEDAL Report No. 2008003, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL); 2008.
[42] Santana R, Larrañaga P, Lozano JA. Adding probabilistic dependencies to the search of protein side chain configurations using EDAs. In: Rudolph G, Jansen T, Lucas S, Poloni C, Beume N, editors. Parallel Problem Solving from Nature—PPSN X, volume 5199 of Lecture Notes in Computer Science. Dortmund, Germany: Springer; 2008. p. 1120–9.
[43] Soto MR, Ochoa A. A factorized distribution algorithm based on polytrees. In: Proceedings of the 2000 Congress on Evolutionary Computation CEC-2000, La Jolla Marriott Hotel La Jolla, California, USA, 6–9 July 2000. Piscataway, NJ, USA: IEEE Press. p. 232–7.
[44] Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, et al. The structure of haplotype blocks in the human genome. Science 2002;296:2225–9.
[45] Lozano JA, Sagarna R, Larrañaga P. Parallel estimation of distribution algorithms. In: Larrañaga P, Lozano JA, editors. Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, Boston/Dordrecht/London: Kluwer Academic Publishers; 2002. p. 125–42.
[46] Mendiburu A, Lozano J, Miguel-Alonso J. Parallel implementation of EDAs based on probabilistic graphical models. IEEE Transactions on Evolutionary Computation 2005;9(4):406–23.