On the application of estimation of distribution algorithms to multi-marker tagging SNP selection

Research Report EHU-KZAA-IK-4/09
Department of Computer Science and Artificial Intelligence
University of the Basque Country

Roberto Santana*, Alexander Mendiburu†, Noah Zaitlen‡
Eleazar Eskin⋆, Jose A. Lozano†
*Universidad Politécnica de Madrid
†Department of Computer Science and Artificial Intelligence
University of the Basque Country, Donostia-San Sebastián, Spain
‡Department of Bioinformatics
University of California San Diego, US.
⋆Computer Science and Human Genetics group
University of California, Los Angeles, US.

Abstract

This paper presents an algorithm for the automatic selection of a minimal subset of tagging single nucleotide polymorphisms (SNPs) using an estimation of distribution algorithm (EDA). The EDA stochastically searches the constrained space of possible feasible solutions and takes advantage of the underlying topological structure defined by the SNP correlations to model the problem interactions. The algorithm is evaluated across the HapMap reference panel data sets. 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. New reduced tagging sets are obtained for all the HapMap SNP regions considered. We also show that the information extracted from the interaction graph representing the correlations between the SNPs can help to improve the efficiency of the optimization algorithm.
keywords: SNPs, tagging SNP selection, multi-marker selection, estimation of distribution algorithms, HapMap.

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 may require the analysis of more than one locus because single locus methods can not be used to identify complex patterns. They miss the genetic contribution to the disease of the interactions between loci [13, 29]. 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 [18, 44, 28].

Nevertheless, genotyping is complicated and very costly when a huge 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 [44, 28]. The other requirement is the conception of efficient procedures for appropriate selection of tagging SNPs.

The problem of choosing tagging SNPs is usually formulated as the objective of selecting the lowest number of (tagging) SNPs such 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-marker tags can significantly outperform tagging efficiency with respect to single-marker approaches [8]. However, in the general case, the single and multi-marker SNP tagging problems are NP-complete [2].

Several approaches have been followed for the solution of the minimal tagging SNP set problem [2, 6, 25, 38]. These approaches have focused on two different but related questions: (1) To determine ways to find tagging SNPs subsets so as to maximize a predefined measure of the subset quality [2, 6, 18, 25, 38, 47] (the search problem) and (2) To find statistical criteria or predictive measures to evaluate the different candidate sets of tagging SNPs (the evaluation problem) [2, 46].

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) [24, 26, 32, 37] is employed. EDAs are evolutionary algorithms similar to genetic algorithms (GAs) [12, 20] but where probabilistic modeling is used instead of genetic operators. EDAs allow to incorporate in a natural way 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 [1, 22].

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, briefly describing their main components and reviewing different variants of these algorithms. Section 4 describes the preprocessing steps required to address the optimization problem under consideration. The EDA approach to the problem is explained in Section 5. Section 6 discusses work related to our research. The experimental framework to evaluate our proposal is presented in Section 7, where the numerical results are analyzed. The conclusions of the paper and ideas for future work are presented in Section 8.

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, $h_1, \ldots, h_m$, each with n different SNPs, $s_1, \ldots, s_n$. The set D can be viewed as an $m \times n$ matrix. D_{ij} denotes the jth SNP in the ith haplotype. 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. The correlation coefficient r^2 measures the similarity correlation between the SNPs in D:

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

where p_{lk} denotes the frequency of observing l and k together in a haplotype and p_l denotes the frequency of l. The r^2 can be generalized to groups of SNPs.

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' a single-marker valid tag of S if $T' \subseteq S$, and $\forall s_j \in S$, $\exists s_i \in T'$ such that $r^2_{ij} \geq r^2_{min}$. Similarly, if $\forall s_j \in S$, $\exists S_T \subseteq T^*$ such that $r^2_{iT} \geq r^2_{min}$, we call T^* a multi-marker valid tag of S.

The problem of finding the smallest single-marker tagging set is the problem of finding the smallest set $T' \subseteq S$ 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 $T^* \subseteq S$ 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, i.e. where $\forall s_j \in S$, $\exists S_T \subseteq T^*, |S_T| \in \{1, 2\}$.

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 GAs [12, 20] 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.

In the last two decades, GAs have been widely used to solve different problems, improving in many cases the results obtained by previous approaches. However, GAs require a large number of parameters (for example, those that control the creation of new individuals) that need to be correctly tuned in order to obtain good results. In addition, GAs show a poor performance in some problems (deceptive and separable problems) in which the existing crossover and mutation operators do not guarantee that better individuals will be obtained by changing or combining existing ones.

Some authors [20] have pointed out that making use of the relations between variables can be useful to drive a more ”intelligent” search through the solution space. This concept, together with the limitations of GAs, motivated the creation of new algorithms grouped under the name of estimation of distribution algorithms (EDAs) [24, 26, 32, 37].

In EDAs, 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 that represent the individuals are explicitly expressed through the joint probability distribution associated with the individuals selected at each generation. A common pseudo-code for all EDAs is described in Algorithm 1.

Algorithm 1: **Estimation of distribution algorithm**

```
Set $t \Leftarrow 0$. Generate $M$ points randomly.
do {
    Evaluate the points using the fitness function.
    Select a set $D_t^S$ of $N \leq M$ points according to a selection method.
    Calculate a probabilistic model of $D_t^S$.
    Generate $M$ new points sampling from the distribution represented in
    the model.
    $t \Leftarrow t + 1$
} until Termination criteria are 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. The probabilistic model learnt at step 5 has a significant influence on the behavior of the EDA from the point of view of complexity and performance. 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) [32], compact genetic algorithm (cGA) [15] and population based incremental learning [3].
- 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) [9], bivariate marginal distribution algorithm (BMDA) [36] and Tree-EDA [41].
- Multiple dependencies: Higher order dependencies between the variables are considered. In this group we can find algorithms like estimation of Bayesian networks algorithm (EBNA) [10], estimation of Gaussian networks algorithms (EGNAs)[23] and the Bayesian optimization algorithm (BOA) [35].

For detailed information about the characteristics of these EDAs, and other algorithms that form part of this family, see [24, 26, 37].

4 Optimization approach: Preprocessing step

The application of EDAs to the minimal tagging set problem requires some preprocessing steps which are analyzed in this section.

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_{ijk} is computed for $i < j < k$ in the original order of SNPs in D. Only pairs of SNPs that are in the sequence at a distance lower than $d = 40000$ 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 $r^2_{min} = 0.8$. 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 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 [6]. The structure of interactions represented by this graph can also be displayed using the adjacency matrix. Figure 1 left) shows the interaction graph for SNPs in the ENm010.CEU

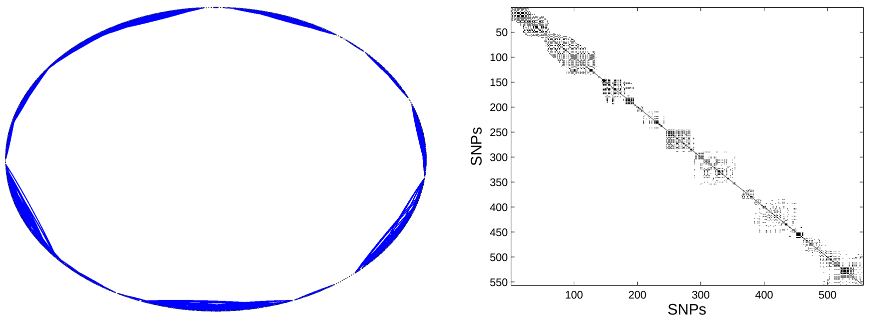

Figure 1: Representation of the interactions between the SNPs in the ENm010.CEU HapMap Encode region. Single tagging SNPs are represented in the graph. Left) Interaction graph. Right) Adjacency matrix.

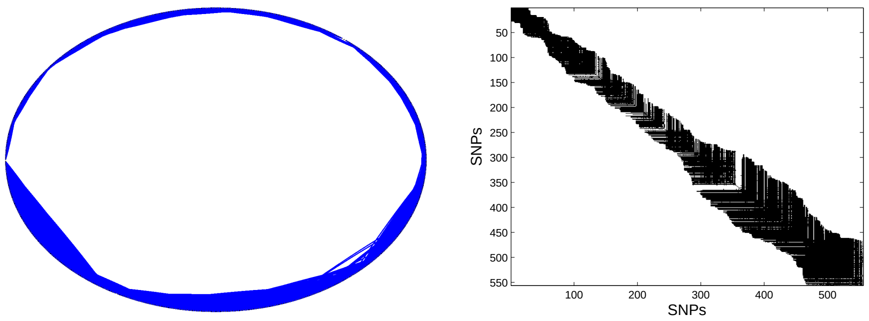

Figure 2: Representation of the interactions between the SNPs in the ENm010.CEU HapMap Encode region. Single and pairs of tagging SNPs are represented in the graph. Left) Interaction graph. Right) Adjacency matrix.

HapMap Encode region [44]. The 556 SNPs are positioned in a circle following the order of the sequence. Figure 1 right) 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. Figure 2 left) 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. Figure 2 right) shows the corresponding adjacency matrix.

Notice that 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 [44]. We call these SNPs fixed. In an interaction graph they can be identified as disconnected nodes.

5 Description of the EDA approach to the SNP problem

To approach the problem of finding the minimal multi-marker tagging set as an optimization problem we define the optimization problem representation and the objective function.

5.1 Problem representation

In our codification of the problem, 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 search of a minimal set of tagging SNPs is done in the subset of $n' \leq n$ SNPs which are not fixed. Therefore, the search space has dimension $2^{n'}$. The final solution comprises all fixed SNPs and those found during the search.

5.2 Fitness function

For implementational reasons, the minimization of the number of tagging SNPs is transformed in the maximization of Equation (2), where each solution $\mathbf{x}$ satisfies that all the non-tagging SNPs are covered by another single or pair of tagging SNPs.

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

5.3 Repairing procedure

It must be taken into account that not all the solutions of the search space 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 feasibility. This procedure is applied during the evaluation step. It is described in Algorithm 2.

Algorithm 2 starts by checking whether $\mathbf{x}$ is a feasible solution. For efficiency reasons, the check is carried out 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 to tagging SNPs and 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.

5.4 Tree-based EDA approach

The EDA of choice uses a probabilistic model that captures bivariate dependencies between the variables. This probabilisti model is based on a tree structure where each variable may depend on at most another variable, which is called

Algorithm 2: **Repairing and evaluation procedure**

```
Compute the set $C_p$ of all SNPs not tagged in the current solution by a single tagging SNP
If $C_p = \emptyset$ output $f(\mathbf{x})$ and exit
do {
    Choose randomly SNP $i$ from $C_p$
    If the set of single tagging SNPs that can potentially tag $i$ is not empty
        Randomly select a SNP $j$ that belongs to this set
    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
        Randomly select a pair $(j, k)$ that belongs to this set, where $k$ is the tagging SNP which is already in the current solution
    Else
        Randomly choose a pair of SNPs $(j, k)$ that can tag $i$
    Set $j$ or $j$ and $k$, as tagging SNPs
    Remove $j$ and all the SNPs tagged by $j$ or by $(j, k)$ from $C_p$
} until $C_p = \emptyset$
Output $f(\mathbf{x})$, $\mathbf{x}$
```

the parent. A probability distribution $p_{Tree}(\mathbf{x})$ that is conformal with a tree is defined as:

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

where $Pa(X_i)$ is the parent of X_i in the tree, and $p(x_i|pa(x_i)) = p(x_i)$ when $Pa(X_i) = \emptyset$, 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 acyclic connected graphs.

There are two main reasons behind the choice of this model. The first is efficiency. The computation of the bivariate dependencies needed to compute a tree is less expensive than the structural learning procedure required to construct more complex models such as general Bayesian networks [34]. 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.

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 [36] or methods based on the analysis of the mutual information between variables [5]. We follow the second approach as shown in Algorithm 3.

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

Algorithm 3: **Tree-EDA**

```
$D_0 \leftarrow$ Generate $M$ individuals randomly
$l = 1$
do {
    $D^s_{l-1} \leftarrow$ Select $N \leq M$ individuals from $D_{l-1}$ according to a selection method
    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 $D^s_{l-1}$
    Calculate the matrix of mutual information using bivariate and univariate marginals.
    Calculate the maximum weight spanning tree from the matrix of mutual information.
    Compute the parameters of the model.
    $D_l \leftarrow$ Sample $M$ individuals (the new population) from the tree and add elitist solutions.
} until A stop criterion is met
```

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 $\frac{n(n-1)}{2}$ pairs of variables. Frequencies are normalized in order to obtain the probabilities. From these marginal probabilities, the mutual information between each pair of variables is computed.

To construct the tree structure, an algorithm introduced in [7], that calculates the maximum weight spanning tree from the matrix of mutual information between pairs of variables, is used. We set a threshold on the minimal mutual information value required to connect two variables. This allows for representing disconnected trees, i.e. a forest. The idea is to capture in the tree structure interactions between those pairs of variables that have the strongest dependence in the data but avoiding the capture of weak dependencies when there are few interactions in the data.

Probabilistic logic sampling [19] is applied to sample new solutions from the tree. New solutions are generated 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 son in the tree is randomly selected proportionally to its conditional probability values conditioned on 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.

5.5 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 [31, 33].

- Constraining the set of interactions to be included in the probabilistic model [4, 39].
- Specifying soft constraints to bias the construction of the probabilistic model [16, 17].

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.

We define a variant of the tree learning algorithm that constrains the calculation of the mutual information to those pairs of variables corresponding to SNPs that have some potential type of tagging relationship, given that their correlation is above the threshold, i.e. they belong to a pair (tagging-tagged) or to a triple (tagging,tagging,tagged) of SNPs. The assumption is that any other pairwise relationship between SNPs is not relevant for the search of the optimal solutions. The variant of Tree-EDA that restricts the interactions represented by the tree structure to interacting pairs of variables is called Tree-EDA^r.

This approach 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 influences 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 problem structure, as with Tree-EDA^r, drastically reduces the time spent to learn the probabilistic model [39, 40].

6 Related work

Minimal tagging SNP selection has been mainly focused on single-marker tagging sets [2, 6, 25, 38]. In multi-marker tagging set, some work has been reported: de Bakker et al. [8] start the search for a multi-marker set from single-marker tagging set. 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. Multi-marker sets of up to three tagging SNPs are allowed. The result of this greedy approach will highly depend 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. [6] approach the minimal single-maker tagging SNP selection problem as an instance of the satisfiability (SAT) problem [42]. The optimal tagging set is obtained by enumerating the solutions to the SAT problem. Preliminary results on the extension of the satisfiability approach to the multi-marker problem are presented for one region of the HapMap benchmark. 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 the tag SNP selection problem [25], haplotype block partitioning [14] and haplotype phasing [45]. However, to the knowledge of the authors, they have not been applied to the minimal tagging SNP set selection problem or other SNP problems within the framework of the optimization algorithms as the proposal presented in this paper.

EDAs have been extensively applied to solve problems from Bioinformatics (see [1] for a survey of EDA applications in this domain). In particular, EDAs based on trees have been used for protein side chain optimization [40] and the minimization of protein contact potentials [39]. Results presented in [39] support evidence that the use of a priori information about the problem structure can notably improve the accuracy and efficiency of the results achieved with EDAs.

7 Experiments

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

7.1 Description of the SNP problem benchmark

To evaluate the introduced algorithms, we used the HapMap reference panel [44]. As done in a previous work [6], 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 are 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.

7.2 Parameters of the algorithms

In order to work, EDAs require the definition of some parameters. The quality of the results achieved by the algorithms will depend on these settings. We have used two different sets of parameters and the same settings have been employed for all instances considered. The population size was set to 5000 and two different number of generations were used (1000 and 5000). Truncation selection with parameter $T = 0.15$ 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 stops when the maximum number of generations is reached or the selected population has become too homogeneous (no more than 10 different individuals).

Table 1: Details of the SNP problem benchmark

name	n	n'	nPairs	Ntriples
$ENm010.CEU$	556	502	2716	102222
$ENm010.CHB$	433	381	3324	113986
$ENm010.JPT$	441	406	2711	82594
$ENm010.YRI$	630	502	1561	61073
$ENm013.CEU$	745	711	7294	434917
$ENm013.CHB$	635	594	5907	300812
$ENm013.JPT$	636	595	6392	326319
$ENm013.YRI$	792	726	3524	187551
$ENm014.CEU$	895	851	7918	510168
$ENm014.CHB$	643	601	6324	252769
$ENm014.JPT$	561	512	5232	200461
$ENm014.YRI$	951	870	4947	304396
$ENr112.CEU$	922	873	9215	692640
$ENr112.CHB$	1015	976	11330	986704
$ENr112.JPT$	997	955	7870	636485
$ENr112.YRI$	1298	1192	5712	527937
$ENr113.CEU$	1054	1004	14535	1273712
$ENr113.CHB$	903	864	16384	1169142
$ENr113.JPT$	829	793	15262	819508
$ENr113.YRI$	1135	1026	5478	399548
$ENr123.CEU$	934	886	6550	531008
$ENr123.CHB$	881	763	9331	746402
$ENr123.JPT$	836	687	5746	387718
$ENr123.YRI$	904	834	5523	404412
$ENr131.CEU$	1026	957	7617	673265
$ENr131.CHB$	1018	920	7290	564586
$ENr131.JPT$	993	893	7367	555791
$ENr131.YRI$	1137	951	5174	426600
$ENr213.CEU$	648	616	5635	276130
$ENr213.CHB$	519	494	5354	181975
$ENr213.JPT$	562	529	5250	220524
$ENr213.YRI$	846	722	3979	206050
$ENr232.CEU$	521	454	4644	166273
$ENr232.CHB$	596	516	3406	141074
$ENr232.JPT$	573	496	3188	134840
$ENr232.YRI$	724	532	1986	78068
$ENr321.CEU$	594	550	5082	242850
$ENr321.CHB$	695	647	6332	365926
$ENr321.JPT$	682	621	5317	305196
$ENr321.YRI$	981	856	3579	236381

7.3 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 and Tree-EDA^r are used to optimize the objective function that measures the number of 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 and Tree-EDA^r was evaluated considering the fitness of the best, average, and worst solutions found in all the experiments. The maximum number of experiments conducted for each instance was 30.

Table 2: Results achieved by Tree-EDA with 1000 generations for the selected instances.

name	nruns	ubest	best	nbest	mean	worst
ENm010.CEU	30	159	123	3	124.33	125
ENm010.CHB	30	99	91	5	92.20	94
ENm010.JPT	30	104	85	18	85.53	87
ENm010.YRI	30	302	255	6	256.37	258
ENm013.CEU	30	114	96	1	99.40	102
ENm013.CHB	30	104	86	2	88.73	91
ENm013.JPT	30	101	89	2	91.37	94
ENm013.YRI	30	235	189	3	191.13	193
ENm014.CEU	30	167	138	2	139.87	142
ENm014.CHB	30	122	103	8	104.10	106
ENm014.JPT	30	121	104	12	104.70	106
ENm014.YRI	30	270	226	5	227.93	231
ENr112.CEU	30	181	139	6	140.53	143
ENr112.CHB	30	165	127	1	130.33	134
ENr112.JPT	30	190	143	2	146.37	149
ENr112.YRI	30	451	323	1	328.30	333
ENr113.CEU	30	183	141	1	143.30	147
ENr113.CHB	30	109	87	1	88.47	89
ENr113.JPT	30	105	85	9	86.17	87
ENr113.YRI	30	367	286	1	290.00	295
ENr123.CEU	30	197	155	1	158.37	161
ENr123.CHB	30	251	228	5	229.43	232
ENr123.JPT	30	289	262	1	263.77	265
ENr123.YRI	30	255	207	1	211.07	215
ENr131.CEU	30	225	173	1	177.23	180
ENr131.CHB	30	271	216	3	218.10	221
ENr131.JPT	30	260	213	3	214.60	216
ENr131.YRI	30	467	386	2	388.00	390
ENr213.CEU	30	128	101	4	102.47	105
ENr213.CHB	30	100	78	3	80.10	82
ENr213.JPT	30	110	86	10	86.73	88
ENr213.YRI	30	328	268	1	271.77	275
ENr232.CEU	30	139	124	6	125.00	126
ENr232.CHB	30	199	165	7	166.73	169
ENr232.JPT	30	194	159	1	161.07	162
ENr232.YRI	30	401	351	6	352.20	354
ENr321.CEU	30	132	106	21	106.33	108
ENr321.CHB	30	159	122	3	123.70	125
ENr321.JPT	30	165	132	2	134.20	136
ENr321.YRI	30	364	288	1	291.50	295

Table 3: Results achieved by Tree-EDA^r with 1000 generations for the selected instances.

name	nruns	ubest	best	nbest	mean	worst
ENm010.CEU	30	159	123	6	124.13	125
ENm010.CHB	30	99	90	1	91.93	94
ENm010.JPT	30	104	85	18	85.40	86
ENm010.YRI	30	302	254	1	256.07	258
ENm013.CEU	30	114	97	1	99.83	102
ENm013.CHB	30	104	87	3	88.77	91
ENm013.JPT	30	101	89	6	90.80	95
ENm013.YRI	30	235	187	1	190.60	195
ENm014.CEU	30	167	137	2	139.90	143
ENm014.CHB	30	122	103	4	104.77	107
ENm014.JPT	30	121	104	13	104.63	106
ENm014.YRI	30	270	225	2	227.80	232
ENr112.CEU	30	181	138	6	139.97	142
ENr112.CHB	30	165	127	1	129.73	133
ENr112.JPT	30	190	143	1	146.67	150
ENr112.YRI	30	451	321	1	326.33	330
ENr113.CEU	30	183	141	1	142.87	145
ENr113.CHB	30	109	87	4	88.33	89
ENr113.JPT	30	105	85	9	86.33	88
ENr113.YRI	30	367	285	1	289.03	293
ENr123.CEU	30	197	154	1	157.67	162
ENr123.CHB	30	251	228	8	229.47	231
ENr123.JPT	30	289	262	3	263.77	266
ENr123.YRI	30	255	208	2	211.00	214
ENr131.CEU	30	225	174	3	176.63	179
ENr131.CHB	30	271	216	2	218.17	221
ENr131.JPT	30	260	211	1	213.97	217
ENr131.YRI	30	467	385	1	387.80	390
ENr213.CEU	30	128	100	4	101.43	103
ENr213.CHB	30	100	78	8	79.20	81
ENr213.JPT	30	110	86	22	86.30	88
ENr213.YRI	30	328	268	1	271.30	275
ENr232.CEU	30	139	123	1	124.77	126
ENr232.CHB	30	199	165	5	166.37	168
ENr232.JPT	30	194	159	1	160.87	163
ENr232.YRI	30	401	350	1	352.13	354
ENr321.CEU	30	132	106	14	106.57	108
ENr321.CHB	30	159	122	2	123.90	126
ENr321.JPT	30	165	132	2	134.13	137
ENr321.YRI	30	364	287	1	290.70	294

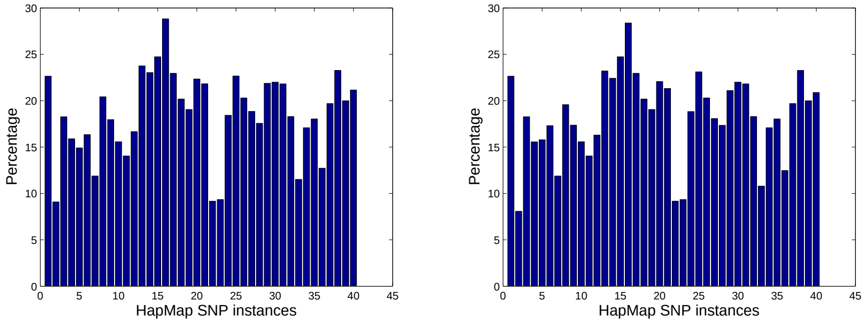

Figure 3: Reduction in the number of tagging SNPs of the minimal multi-marker tagging set with respect to the single-marker minimal tagging set. Left) Best solution obtained by Tree-EDA^r. Right) Best solution obtained by Tree-EDA.

7.4 Numerical results

We compared the quality of the solutions obtained by Tree-EDA and Tree-EDA^r using the SNP problem benchmark. Tables 2 and 3 respectively show the results of Tree-EDA and Tree-EDA^r with 1000 generations. The tables show the number of experiments, out of 30, that were successfully completed (nruns), the best solution obtained when only a single tagging SNP is allowed (ubest) as obtained using the SAT tagger [6], the best solution obtained in all the completed experiments (best), the number of times a solution with this score has been achieved (nbest), the average (mean) and worst (worst) values of the solutions found in all the experiments.

An analysis of the tables reveals that the worst solution obtained in all the experiments by Tree-EDA^r and Tree-EDA is always better than the minimal single-marker tagging set. The reduction in the number of tagging SNPs reaches 30% for some problems. Figure 3 shows the percentage of reduction in the number of tagging SNPs of the minimal multi-marker tagging set with respect to the single-marker minimal tagging set.

In terms of the difference between Tree-EDA^r and Tree-EDA, the first algorithms achieves a better average of the solutions for 29 of the 40 instances, in one case both algorithms achieve the same average result, and for 10 instances Tree-EDA achieves a better performance. Even if the use of a priori problem information improves the results for most of the instances, this is not always the case. To investigate the reasons that explain this behavior, and in particular, to determine when the learned dependencies contribute to a more efficient search, are relevant issues which we postpone for future research.

We conducted additional experiments to investigate whether the increase in the number of generations leads to an improvement in the solutions. For computational reasons, only 15 experiments were conducted for each problem. Table 4 shows the results of Tree-EDA^r with 5000 generations. These results show that by spending more time in the search the solutions can be further improved.

Table 4: Results achieved by Tree-EDA^r with 5000 generations for the selected instances.

name	nruns	ubest	best	nbest	mean	worst
ENm010.CEU	15	159	122	3	123.27	125
ENm010.CHB	15	99	91	6	91.60	92
ENm010.JPT	15	104	85	11	85.27	86
ENm010.YRI	15	302	254	1	255.53	257
ENm013.CEU	15	114	95	1	98.07	100
ENm013.CHB	15	104	87	2	88.27	90
ENm013.JPT	15	101	88	6	89.00	91
ENm013.YRI	15	235	186	2	188.20	193
ENm014.CEU	15	167	136	1	138.87	142
ENm014.CHB	15	122	103	8	103.67	105
ENm014.JPT	15	121	104	15	104.00	104
ENm014.YRI	15	270	221	1	225.87	229
ENr112.CEU	15	181	137	3	139.00	141
ENr112.CHB	15	165	127	1	129.07	131
ENr112.JPT	15	190	142	2	144.53	147
ENr112.YRI	15	451	318	1	323.80	328
ENr113.CEU	15	183	141	5	142.40	144
ENr113.CHB	15	109	86	3	87.60	89
ENr113.JPT	15	105	85	6	86.20	87
ENr113.YRI	15	367	284	2	287.67	291
ENr123.CEU	15	197	154	2	156.80	160
ENr123.CHB	15	251	227	1	228.47	230
ENr123.JPT	15	289	261	3	262.47	265
ENr123.YRI	15	255	207	1	209.87	212
ENr131.CEU	15	225	171	1	174.20	177
ENr131.CHB	15	271	215	1	218.00	220
ENr131.JPT	15	260	213	7	213.60	215
ENr131.YRI	15	467	385	1	387.13	389
ENr213.CEU	15	128	99	1	100.80	103
ENr213.CHB	15	100	78	7	78.60	80
ENr213.JPT	15	110	86	15	86.00	86
ENr213.YRI	15	328	266	1	269.47	273
ENr232.CEU	15	139	123	5	123.93	125
ENr232.CHB	15	199	163	1	165.00	167
ENr232.JPT	15	194	159	5	159.73	161
ENr232.YRI	15	401	351	7	351.80	353
ENr321.CEU	15	132	106	14	106.07	107
ENr321.CHB	15	159	120	1	122.40	124
ENr321.JPT	15	165	130	2	131.87	134
ENr321.YRI	15	364	288	5	289.93	293

8 Conclusions and future work

We have presented an optimization approach for finding the minimal set of multi-marker tagging SNPs. The optimization problem has dealt with using an estimation of distribution algorithm. The obtained solutions considerably improved those achieved by exact algorithms for the single-marker tagging SNP problem.

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 EDAs we have applied exhibit other particular features that explain their success for computing the minimal set of multi-marker tagging SNPs: 1) They can incorporate structural information about the problem into the search. 2) They take 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 analyze in detail some of the possibilities for future work.

8.1 Future work to improve the results of the minimal tagging problem

8.1.1 Biasing the initial population

The EDAs used in our experiments start from a randomly generated population of solutions. However, incorporating knowledge about the problem in the starting population can improve the results of the algorithm. Seeding is the process of constructing the initial solutions according to previous information about the problem. In our case, seeding can be applied by first ranking SNPs according to the number of SNPs they can potentially tag [8] and generating then the initial populations prioritizing solutions that contain SNPs with better ranking.

8.1.2 Use of other probabilistic models

Trees are very convenient models for EDAs because they are able to represent to some extent the interactions between the variables but with a constrained complexity. This means that by representing only pairwise variable interactions they guarantee a balance between the accuracy of the representation and the efficiency of the model. However, 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 (even at the expense of a higher computational time). Two direct extensions of EDAs based on trees are: EDAs that use mixtures of trees [41] and polytrees [43]. Mixtures of trees can serve to investigate the effect of a clustering of the solutions in the accuracy of the probabilistic representation. In terms of complexity, polytrees are an intermediate model between trees and general Bayesian networks and could also serve to increase the accuracy of the representation but keeping the complexity of the model feasible.

8.1.3 Combination with local optimization methods

The ”peel back” approach of de Bakker et al. [8], commented in Section 6 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.

8.2 Future work to extend the applications of EDAs to similar problems

8.2.1 Relaxing the fitness function to consider global strength of correlations

We have just considered the case of the minimal tagging set. However, it is possible to include in the fitness function the strength of the r^2 correlations.

To determine the strength of the correlation between the tagging SNP set S and the tagged SNP s_j, the SNP or pair of SNPs in S for which the correlation value with SNP s_j is *maximum* is taken.

Let $\hat{r}$ be the average of the correlation values computed for all tagged SNPs. The maximum value it can take is 1 (perfect correlation). Since the fitness function we use is the number of tagging SNPs $f(\mathbf{x}) = n' - \sum_{i=1}^{n'} x_i$, we can include the quality of the tagging set by setting $\hat{f}(\mathbf{x}) = f(\mathbf{x}) + \hat{r}(\mathbf{x}) * 0.99$. This function will increase with $\hat{r}$, but it is guaranteed that a solution whose number of tagging SNPs is q is always better than a solution with $q + 1$ tagging SNPs.

We have assumed that $\hat{r}$ is computed as the maximum of the correlations between each tagged SNP and its tagging SNPs. However, we can introduce another way to measure the strength of the correlation based on an *average* of the correlations between the tagging set of SNPs and the tagged SNP s_j. This *average* could be a measure of a consensus evidence between a subset of tagging SNPs and the tagged SNP. For an optimal solution given this measure, we can expect that if information for one of the tagging SNPs fails, the remaining tagging SNPs could still give a good prediction of the failed SNP.

By using a parameter k, we can set a compromise criterion between the maximum and average criteria. The *k-average* criterion will be the average of the correlation between the tagged SNP and the k tagging SNPs with maximum correlation where k is a parameter of the problem. The maximum criterion is subsumed by the k-average criterion when we take $k = 1$.

To summarize, the following are the three strategies that can be used to measure the strength of correlations and compute $\hat{r}$.

- Maximum of the correlation between the tagged and its tagging SNPs.
- Average of the correlation between tagged and all its tagging SNPs.
- (k-average) Average of the correlation between the tagged SNP and the k tagging SNPs with maximum correlation where k is a parameter of the problem.

8.2.2 Block-free problem formulation

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 [44], it does not appear to be possible to unambiguously and uniquely infer the true block partitioning [2]. These blocks are capturing general regions of low diversity, but the boundaries between them are not rigourously 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 [11]. 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 [2]. 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 [27, 30] could be an interesting alternative in this case.

8.2.3 Formulation as a constrained and/or multiple objective optimization problem

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 minimum k such that all the SNPs are tagged has been the solution of the problem investigated in this paper. 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 SNP 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 quality of the SNP correlations could be included in the objective that measures the number of SNPs tagged, as discussed in Section 8.2.1. Multi-objective formulations could also be employed to include the cost of the solutions, given some a priori information about the difficulties associated to genotyping each SNP. One multi-objective approach to this type of problem has been proposed in [21].

The Tree-EDA algorithm can be adapted to deal with multi-objective problems by modifying the selection step to include Pareto-set approximation.

References

[1] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. Van de Peer, R. Blanco, V. Robles, C. Bielza, and P. Larrañaga. A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, 1(6), 2008.

[2] V. Bafna, B. V. Halldorsson, R. Schwartz, A. G. Clark, and S. Istrail. 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*, pages 19–27, New York, NY, USA, 2003. ACM.

[3] S. Baluja. 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.

[4] S. Baluja. Incorporating a priori knowledge in probabilistic-model based optimization. In M. Pelikan, K. Sastry, and E. Cantú-Paz, editors, *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Studies in Computational Intelligence, pages 205–222. Springer, 2006.

[5] S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proceedings of the 14th International Conference on Machine Learning*, pages 30–38. Morgan Kaufmann, 1997.

[6] A. Choi, N. Zaitlen, B. Han, K. Pipatsrisawat, A. Darwiche, and E. Eskin. Efficient genome wide tagging by reduction to SAT. In *Proceedings of the 8th International Workshop Algorithms in Bioinformatics WABI-2008*, volume 5251 of *Lectures Notes in Computer Science*, pages 135–147. Springer, 2008.

[7] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[8] P. I. W. de Bakker, R. Yelensky, I. Pe'er, S. B. Gabriel, M. J. Daly, and D. Altshuler. Efficiency and power in genetic association studies. *Nature Genetics*, 37:1217–1223, 2005.

[9] J. S. De Bonet, C. L. Isbell, and P. Viola. MIMIC: Finding optima by estimating probability densities. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 424–430. The MIT Press, Cambridge, 1997.

[10] R. Etxeberria and P. Larrañaga. Global optimization using Bayesian networks. In A. Ochoa, M. R. Soto, and R. Santana, editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 151–173, Havana, Cuba, 1999.

[11] S. B. Gabriel et al. The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229, 2002.

[12] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[13] J. E. Goodman, L. E. Mechanic, B. T. Luke, S. Ambs, S. Chanock, and C. C. Harris. Exploring SNP-SNP interactions and colon cancer risk using polymorphism interaction analysis. *Journal of Cancer*, 118(7):1790–1797, 2006.

[14] G. Greenspan and D. Geiger. High density linkage disequilibrium mapping using models of haplotype block variation. *Bioinformatics*, 20(Suppl. 1):i137–i144, 2004.

[15] G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297, 1999.

[16] M. Hauschild and M. Pelikan. Enhancing efficiency of hierarchical BOA via distance-based model restrictions. MEDAL Report No. 2008007, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL), April 2008.

[17] M. Hauschild, M. Pelikan, K. Sastry, and D. E. Goldberg. Using previous models to bias structural learning in the hierarchical BOA. MEDAL Report No. 2008003, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL), 2008.

[18] A. G. Heidema, J. M. A. Boer, N. Nagelkerke, E. C. M. Mariman, D. L. van der A, and E. J. M. Feskens. The challenge for genetic epidemiologists: how to analyze large numbers of SNPs in relation to complex diseases. *BMC Genetics*, 7(23), 2006.

[19] M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In J. F. Lemmer and L. N. Kanal, editors, *Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence*, pages 149–164. Elsevier, 1988.

[20] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[21] R. M. Hubley, E. Zitzler, and J. C. Roach. Evolutionary algorithms for the selection of single nucleotide polymorphisms. *BMC Bioinformatics*, 4(30):1790–1797, 2003.

[22] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7:86–112, 2006.

[23] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. 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.

[24] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation.* Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[25] P. H. Lee and H. Shatkay. Bntagger: improved tagging SNP selection using Bayesian networks. *Bioinformatics*, 22(14):e211–219, 2006.

[26] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms.* Springer, 2006.

[27] J. A. Lozano, R. Sagarna, and P. Larrañaga. Parallel estimation of distribution algorithms. In P. Larrañaga and J. A. Lozano, editors, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, pages 125–142. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[28] T. A. Manolio, L. D. Brooks, and F. S. Collins. A HapMap harvest of insights into the genetics of common disease. *The Journal of Clinical Investigation*, 118(5):1590–1605, 2008.

[29] L. E. Mechanic, B. T. Luke, J. E. Goodman, S. Chanock, and C. C. Harris. Polymorphism interaction analysis (PIA): a method for investigating complex gene-gene interactions. *BMC Bioinformatics*, 9(146):1790–1797, 2008.

[30] A. Mendiburu, J. Lozano, and J. Miguel-Alonso. Parallel implementation of EDAs based on probabilistic graphical models. *IEEE Transactions on Evolutionary Computation*, 9(4):406–423, 2005.

[31] H. Mühlenbein, T. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.

[32] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, volume 1141 of *Lectures Notes in Computer Science*, pages 178–187, Berlin, 1996. Springer.

[33] A. Ochoa, M. R. Soto, R. Santana, J. Madera, and N. Jorge. The factorized distribution algorithm and the junction tree: A learning perspective. In A. Ochoa, M. R. Soto, and R. Santana, editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 368–377, Havana, Cuba, March 1999.

[34] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, California, 1988.

[35] M. Pelikan. *Hierarchical Bayesian Optimization Algorithm. Toward a New Generation of Evolutionary Algorithms.* Studies in Fuzziness and Soft Computing. Springer, 2005.

[36] M. Pelikan and H. Mühlenbein. The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535, London, 1999. Springer.

[37] M. Pelikan, K. Sastry, and E. Cantú-Paz, editors. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications.* Studies in Computational Intelligence. Springer, 2006.

[38] T. M. Phuong, Z. Lin, and R. B. Altman. Choosing SNPs using feature selection. In *Proceedings of the Fourth International IEEE Computer Society Computational Systems Bioinformatics Conference*, pages 301–309. IEEE Computer Society, 2005.

[39] R. Santana, P. Larrañaga, and J. A. Lozano. The role of a priori information in the minimization of contact potentials by means of estimation of distribution algorithms. In E. Marchiori, J. H. Moore, and J. C. Rajapakse, 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*, pages 247–257. Springer, 2007.

[40] R. Santana, P. Larrañaga, and J. A. Lozano. Adding probabilistic dependencies to the search of protein side chain configurations using EDAs. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature - PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 1120–1129, Dortmund,Germany, 2008. Springer.

[41] R. Santana, A. Ochoa, and M. R. Soto. The mixture of trees factorized distribution algorithm. In L. Spector, E. Goodman, A. Wu, W. Langdon, H. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and

E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, pages 543–550, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[42] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92)*, pages 440–446, San Jose, CA, USA, 1992.

[43] M. R. Soto and A. Ochoa. A factorized distribution algorithm based on polytrees. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC-2000*, pages 232–237, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 July 2000. IEEE Press.

[44] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million snps. *Nature*, 449(7164):851–861, 2007.

[45] E. P. Xing, R. Sharan, and M. I. Jordan. Bayesian haplo-type inference via the Dirichlet process. In *Proceedings of the twenty-first international conference on Machine learning (ICML-04):*, pages 879–886, New York, NY, USA, 2004. ACM.

[46] N. Zaitlen, H. M. Kang, E. Eskin, and E. Halperin. Leveraging the HapMap correlation structure in association studies. *American Journal of Human Genetics*, 80(4):683–691, 2007.

[47] N. Zhou and L. Wang. Effective selection of informative snp and classification on the HapMap genotype data. *BMC Bioinformatics*, 8:484, 2007.