Rules extraction in linkage disequilibrium mapping with an adaptive genetic algorithm

Laetitia Jourdan, Clarisse Dhaenens, El-Ghazali Talbi

Discovering haplotypes in linkage disequilibrium mapping with an adaptive genetic algorithm

Laetitia Jourdan**, Clarisse Dhaenens, and El-Ghazali Talbi

LIFL Université de Lille1,
Bât M3 Cité Scientifique,
59655 Villeneuve d'Ascq Cedex
FRANCE
jourdan@lifl.fr,
WWW home page: http://www.lifl.fr/~jourdan

Abstract. In this paper, we present an evolutionary approach to discover candidate haplotypes in a linkage disequilibrium study. This work takes place into the study of factor involved in multi-factorial diseases such as diabetes and obesity. A first study on the linkage disequilibrium problem structure led us to use a genetic algorithm to solve it. Due to the particular but classical evaluation function given by the biologists, we design our genetic algorithm with several populations. This model lead us to implement different cooperative operators as mutations and crossovers. Those mechanisms have their probabilities of application which are set adaptively. In order to introduce some diversity, we also implement a random immigrant strategy and to cover up the cost of the evaluation we parallelize it in a master / slave model. Different combinations of the presented mechanisms are tested on real data and compared in term of robustness and evaluation cost. We show that the most complete strategy is able to find the best solutions and is the most robust.

1 Introduction

Using single-nucleotide polymorphisms (SNPs) is currently a major way in the search for genes involved in complex diseases. In order to find candidate haplotypes of SNPs for multi-factorial diseases such as diabetes and obesity, we develop a study with the multi-factorial Disease Laboratory of Lille (France). In this study, we have to look for disease-associated haplotypes in a very large number of loci on different chromosomes. This generates a lot of data.
On a previous study of the problem structure [5], we have shown that classical algorithms are not adapted to deal with this search and that it is paramount to develop a method which is able to deal with a very large search space and which have a good exploration potential.
In this work, we propose to solve this problem with a dedicated genetic algorithm which is based on several subpopulations because of the biological problem. This

** This work is supported by the Nord-Pas de Calais region and the Genopôle of Lille.

particular model lead us to use cooperative operators which are set adaptively. As the evaluation process imposed by biologists is time consuming, we also proposed a parallel implementation.
This paper is organized as follows. The second section of this paper will give some biological definitions, presents the data and the biological problem with its particular evaluation function. In the third section, the dedicated genetic algorithm and its specificities will be presented. Then, the fourth section will present results obtained thanks to this algorithm. Finally the conclusion will give indications about exploitation of the results.

2 The biological problem

This work deals with the linkage disequilibrium for multi-factorial diseases and in particular with the studies of factors that are implied in diseases such as diabetes and obesity. We will firstly introduce some notions of biology [1, 9] that are necessary to understand the problem and show what kind of data we have to exploit. Then we will formulate the biological problem.

2.1 Definitions and Data

Genetic markers are alleles of genes, or DNA polymorphisms, that are used as experimental probes to keep track of an individual, a tissue, a cell, a nucleus, a chromosome, or a gene. Stated another way, any character that acts as a signpost or signal of the presence or location of a gene or heredity characteristic for an individual in a population. There are 4 chromosome changes that do occur from generation to generation, and these are known as markers: indel, snips (SNP's), micro-satellites and mini-satellite. In our case we are interested in single nucleotide polymorphisms (SNPs).
SNPs are DNA sequence variations that occur when a single nucleotide (A,T,C, or G) in the genome sequence is changed. Most SNPs, actually about two of every three SNPs, involve the replacement of cytosine (C) with thymine (T). SNPs occur every 100 to 300 bases along the human genome. SNPs are stable from an evolutionarily standpoint –not changing much from generation to generation –making them easier to follow in population studies.
Data available for our study are, for all the individuals that have been examined, the form of all their SNPs and their status (affected / not affected) (see table 1). Then in order to discover associations between SNPs and the disease, we know for all the SNPs the mean frequency of each alternative (1 and 2) (see table 2).

An haplotype is a set of closely linked alleles (genes or DNA polymorphisms) inherited as a unit and is a contraction of the phrase "haploid genotype". Different combinations of polymorphisms are known as haplotypes. Two reasons lead us to use multi-locus analysis. First, haplotype are more specific of the ancestral chromosome where the mutation occurred. Moreover, it is possible that several loci (SNPs) have an effect on the disease risk.
Two alleles are said in linkage equilibrium if for all SNP A and SNP B which

Table 1. Data on individuals.

Individual SNP	SNP_1	...	SNP_n	STATUS
IND_1	11	...	22	Not affected
IND_2	11	...	12	Affected
...	...	...	...	...
IND_{k-1}	12	...	22	Not affected
IND_k	11	...	22	Unknown

Table 2. Frequencies of SNPs.

SNP	Freq. of 1	Freq. of 2
1	0.997	0.003
2	0.856	0.144
...	...	...
n-1	0.576	0.424
n	0.389	0.611

Table 3. Disequilibrium between SNPs.

SNP_1	SNP_2	Disequilibrium
1	2	-1.00
1	3	0.67
...	...	...
n-1	n	0.42

are between these two loci the frequency of the haplotype AB is equal to the product of the frequencies of the SNPs A and B. In the other case, we will talk about linkage disequilibrium.
For example, let us compute the linkage disequilibrium for HLA (Human Leukocyte Antigen). The HLA has four markers (A, B, C and D) that have several forms. Let the frequencies $F(HLA-A)$ and $F(HLA-C)$ be the frequencies for markers A and C of HLA. $F(HLA-A) = 0.161$ and $F(HLA-C) = 0.153$. The frequency of HLA-A / HLA-C with no linkage disequilibrium is : $0.161 \times 0.153 = 0.0246$ but the observed frequency is 0.089. So the linkage disequilibrium is $D = 0.089 - 0.0246$ and equals to 0.064. The linkage disequilibrium is then tested with a χ^2 test for evaluating the relevance of the frequencies difference. In our study, we know for all pairs of SNPs their linkage disequilibrium (see table 3). This disequilibrium belongs to the interval in [-1,1] where -1 and 1 signify a huge disequilibrium whereas a value near 0 signifies no linkage disequilibrium.

2.2 Description of the problem

The objective of our application is to find haplotypes that are able to explain the disease under study. Two SNPs of an haplotype must verify two properties (to be in disequilibrium). Firstly, their two by two disequilibrium must be less than a threshold S_1. Secondly, the difference between the smaller frequencies of the two alternatives must be greater than a threshold S_2. There will be several haplotypes that verify these constraints. In order to evaluate the quality of an haplotype biologists often use the two procedures EH-DIALL [10] and CLUMP [3].
EH-DIALL (EH is for Estimated Haplotype) is a procedure that determines the

most probable distribution of alleles in an haplotype according to values of the SNPs. Given a sample consisting on a large number of individuals collected at random from the population (see table 1), EH-DIALL program estimates allele frequencies for each marker. Haplotype frequencies are estimated with allelic association (Hypothesis H_1) and without allelic association (Hypothesis H_0).
CLUMP is a program designed to assess the significance of the departure of observed values in a contingency table from the expected values conditional on the marginal totals [3]. CLUMP produces several statistics. The one that corresponds to our problem is referred to as T_1. A good haplotype is an haplotype that is highly correlated with the disease, which corresponds to a high value of T_1.

The whole evaluation process for an haplotype is:

- Starting from a set of candidate SNPs forming an haplotype, estimate independently, for affected and unaffected people, the distribution of alleles in the haplotype thanks to EH-DIALL.
- Use CLUMP to evaluate the association haplotype-disease.

This process allows us to evaluate the quality of an haplotype in biological terms and our objective will be to find haplotypes that maximize this criterion of quality. We will consider this problem as a combinatorial optimization problem where the search space is composed of all the combinations of SNPs and the objective function is the maximization of the quality describes above.

3 The specific genetic algorithm

We will present our steady state genetic algorithm to discover designed candidate haplotypes that are able to explain the disease. The general scheme of the algorithm is given in figure 1. To present the genetic algorithm adapted to this problem, we have to define the encoding, operators and advanced search mechanisms that are used.

3.1 Encoding

The encoding is very intuitive: an individual represents an haplotype (a set of SNPs). It is encoded with a structure that stores the size of the haplotype, a table of the SNPs that compose it and its fitness.

3.2 Multi-Populations

A charasteritic of this problem is that haplotypes of different sizes are not directly comparable between them. For example, for one of the dataset, the best haplotypes of size 3 has a fitness of 58.81 and the best haplotype of size 4 has a fitness of 84.85. These values have been found thanks to a complete enumeration of haplotypes of size three and four. Hence the global population will be divided into several subpopulations, where each subpopulation corresponds to a

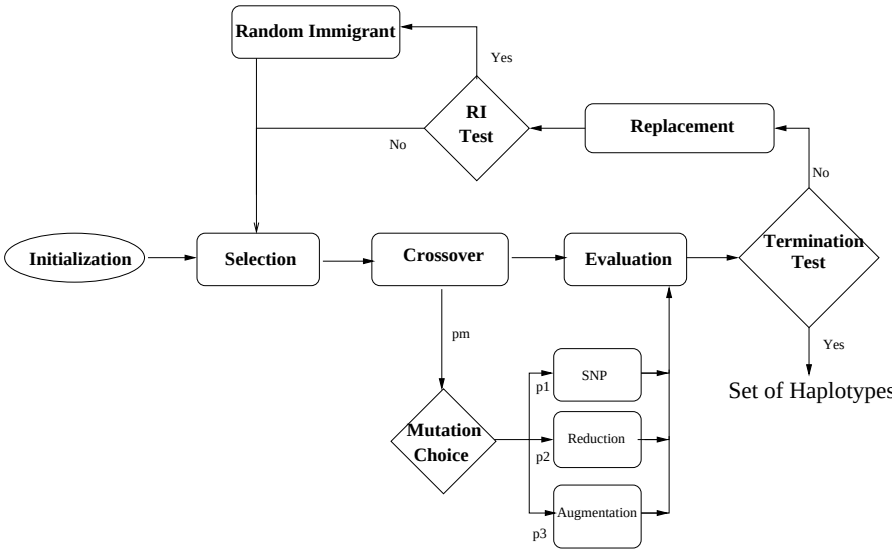

Fig. 1. The general scheme of our genetic algorithm.

given size of haplotype. The number of individuals in each subpopulation are not equal and increase with the size of the haplotypes in order to follow the growth of the size of the search space related to each size (see figure 2). Using some genetic operators, some cooperations between subpopulations will occur during mutation and crossover.

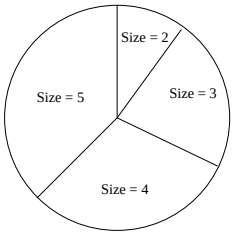

Fig. 2. An example of subpopulations repartition.

3.3 Operators

Mutation: We implement three kinds of mutation operators that sightly modify an individual:

- Mutation of a SNP: we randomly choose a SNP of the individual and replace it by another randomly chosen SNP. This process is similar to a local

search which allows to explore the neighborhood of the solution. We use this mutation several times in parallel and keep the best individual found.
- Reduction Mutation: we randomly choose a SNP of the individual and remove it. The individual has now a lower size. This operator allows to move individuals from a subpopulation to another. This operator constructs smaller haplotype and tries to generalize the association.
- Augmentation Mutation: we add a randomly chosen SNP. This mutation constructs increasingly large size haplotypes and specialize the association of SNPs.

Probabilities of mutation are hard to set when we have several mutation operators and are often set them experimentally. To overcome this problem, we implement an adaptive strategy for calculating the rate of each mutation operator. Many authors have worked on setting automatically probabilities of applying operator [2, 8, 6]. In [7], authors proposed to compute the new rate of mutation by calculating the progress of the j^{th} application of mutation M_i, for an individual *ind* mutated into an individual *mut* as follows:

$$progress_j(M_i) = Max(fitness(ind), fitness(mut)) - fitness(ind)$$

But we have mutation operators that increase or decrease the number of SNPs and we saw that the fitness function gave by the biologists is correlated to the number of SNPs. In order to adapt the notion of progress to our problem, we normalize the progress with the best individual and the worst of the subpopulation corresponding to the individual (the best and the worst individuals of the same size). The progress is:

$$progress_j(M_i) = Max(norm(ind), norm(mut)) - norm(ind)$$

Then for each mutation operator M_i, assume $Nb\_mut(M_i)$ applications of the mutation are done during a given generation ($j = 1, .., Nb_{mut}(M_i)$). Then we can compute the profit of a mutation M_k :

$$Profit(M_k) = \frac{\sum_j progress_j(M_k)/Nb\_mut(M_k)}{\sum_i \left(\sum_j progress_j(M_i)/Nb\_mut(M_i) \right)}$$

We set a minimum rate δ and a global mutation rate $p_{mutation}$ for N mutation operators to apply. The new mutation ratio for each M_i is calculated using the following formula [7]:

$$p(M_i) = Profit(M_i) \times (p_{mutation} - N \times \delta) + \delta$$

The sum of all the mutation rates is equal to the global rate of mutation $p_{mutation}$. The initial rate of each mutation operator is set to $p_{mutation}/N$.

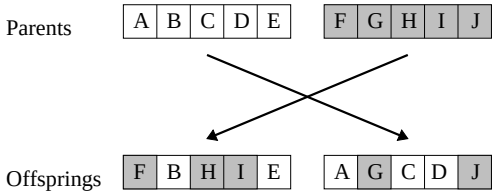

Fig. 3. Uniform crossover of individuals of size 5.

Crossover: We use an uniform quadratic crossover: take the two strings of SNPs of the parents and create two children by randomly shuffling the variables corresponding to the SNP at each site (see figure 3). Then we calculate the number of SNPs and the score.

We use two kinds of crossovers:

- Intra-population: only crossovers between individuals of a same subpopulation are allowed
- Inter-population: crossovers between individuals of different subpopulations are allowed.

The probabilities of application of each kind of crossover are also set in an adaptive manor. We use the same strategy used for mutation (see 3.3) to the case of the quadratic crossover. We define the improvement of a child e with regards to its parents p_1 and p_2 for intra-population crossover as:

$$2 \times Improve_{Intra}(e, p_1, p_2) = \begin{array}{l} \frac{Max(fitness(p_1), fitness(e)) - fitness(p_1)}{best\_of\_size(e.size)} \\ + \\ \frac{Max(fitness(p_2), fitness(e)) - fitness(p_2)}{best\_of\_size(e.size)} \end{array}$$

In this case, the three individuals e, p_1 and p_2 are of the same size. They can be compared.

We define the improvement for the inter-population crossover by only comparing the improvement between a child e and its parent of the same size:

$$Improve_{Inter}(e, p_1, p_2) = \begin{cases} \frac{Max(fitness(p_1), fitness(c)) - fitness(p_1)}{best\_of\_size(e.size)} & If\ size.p_1 = size.e \\ \frac{Max(fitness(p_2), fitness(c)) - fitness(p_2)}{best\_of\_size(e.size)} & If\ size.p_2 = size.e \end{cases}$$

Hence, the intra-improvement considers two comparisons and the inter-improvement only one. This is the reason why we must divide by two the intra-improvement.

So the global function for the improvement is:

$$Improve(e, p_1, p_2) = Improve_{Intra}(e, p_1, p_2) \ \ OR \ \ Improve_{Inter}(e, p_1, p_2)$$

The progress of the j^{th} application of each crossover C_i, which mates two individuals p_1 and p_2 to obtain two children e_1 and e_2 is:

$$progress_j(C_i) = Improve_j(e_1, p_1, p_2) + Improve_j(e2, p_1, p_2)$$

Then for all the crossover operators C_i, assume $Nb\_cross(C_i)$ applications of the crossover are made during a given generation. Then the profit of a crossover C_k is:

$$Profit(C_k) = \frac{\sum_j progress_j(C_k)/Nb\_cross(C_k)}{\sum_i \left(\sum_j progress_j(C_i)/Nb\_cross(C_i)\right)}$$

We set a minimum rate δ and a global crossover rate $p_{crossover}$ for N crossover operators to apply. The new crossover ratio for each C_i is calculated using the same formula than in the paragraph 3.3 for the mutation by replacing $p_{mutation}$ by $p_{crossover}$.

3.4 Random Immigrant

We use random immigrant to introduce some diversity in the search and to avoid premature convergence. This mechanism replaces all the individuals that have a fitness under the mean fitness of the population by new generated individuals and it is setting off when the best individual has not been improved in a given number of generations.

3.5 Parallel implementation

The evaluation function of the problem is time consuming. In order to run the algorithm in a reasonable time, we have made a synchronous parallel implementation of the evaluation phase.
The implementation is based on a master / slaves model. The slaves are initiated at the beginning and access only once to the data. During the evaluation phase, the master gives each slave an individual to evaluate. Then the slave computes the fitness of this individual and send it back to the master.
The programming environment used is C/PVM (Parallel Virtual Machine) on a network of PC under Linux [4].

4 Results

Table 4 presents the results obtained by different combinations of the mechanisms explain above. For each combination we made five runs and we indicate the best haplotype found over the five runs, its size and fitness. We also report the mean fitness and its standard deviation (Dev.) for the five runs. Finally, we give the minimum number of required evaluations to obtain the solution and the mean over the five runs.
We highlight in grey, the best haplotype found over the five runs when it does

not correspond to the optimal one. So, for combinations 1. and 3., the best haplotype of size four has not been found in any of the five runs.
The column Dev. gives the standard deviation of the fitness found for the five runs. A zero indicates that each run has found the same solution. We can see that for combination 1. to 5., this deviation is not always equal to zero. Only combination 6. allows to find at each run and for the different size the best solution.
Moreover, the last column shows that the number of evaluations required for each combination is of the same range. This indicates that introducing complex mechanisms does not penalize the search time.
All these experiments show that the complete version (combination 6.) is the most interesting (best result in reasonable time).

Table 4. Comparison of results obtained by the GA.

Scheme	Best Haplotype	Size	Fitness	Mean	Dev	Min Nb. of Eval.	Mean
1. Simple GA without cooperative operators	8 12 15	3	58.814	58.8146	0	175	766
	6 8 16 31	4	80.631	78.942	5.908	1938	2714.6
	8 12 16 33 43	5	123.108	123.108	0	2581	4271
	8 12 15 21 32 43	6	161.252	158.818	2.444	3762	11307.8
2. Simple GA + Random Immigrant	8 12 15	3	58.814	58.814	0	167	354.6
	8 18 26 50	4	84.856	82.478	2.372	1287	2254.8
	8 12 16 33 43	5	123.108	123.108	0	1375	4410.6
	8 12 15 21 32 43	6	161.252	160.282	0.98	6051	9074
3. Simple GA with cooperative non adaptive operators	8 12 15	3	58.814	58.8146	0	187	425.8
	6 8 16 31	4	80.631	79.2622	5.58	644	3584
	8 12 16 33 43	5	123.108	123.108	0	1778	6749
	8 12 15 21 32 43	6	161.252	161.252	0	6676	11620
4. Adaptive Mutation + Crossover intra population	8 12 15	3	58.814	58.814	0	302	603.6
	8 18 26 50	4	84.856	82.478	2.371	375	3164.8
	8 12 16 33 43	5	123.108	123.108	0	2397	4737.8
	8 12 15 21 32 43	6	161.252	161.252	0	3364	10074
5. Adaptive Mutation + Adaptive crossover	8 12 15	3	58.814	58.814	0	207	1110.2
	8 18 26 50	4	84.856	83.162	1.688	426	2809.6
	8 12 16 33 43	5	123.108	123.108	0	2227	3474.2
	8 12 15 21 32 43	6	161.252	161.252	0	4455	9851.2
6. Adaptive Mutation + Adaptive crossover + Random Immigrant	8 12 15	3	58.814	58.814	0	317	587.4
	8 18 26 50	4	84.856	84.856	0	1111	3238.2
	8 12 16 33 43	5	123.108	123.108	0	2994	5615.2
	8 12 15 21 32 43	6	161.252	161.252	0	11573	15464.6

5 Conclusion

In this paper, we have presented a parallel multi-populations genetic algorithm for the search of candidate haplotypes in linkage disequilibrium study of multifactorial diseases. Our aim was to respect the constraints given by the biologists and in particular to use a dedicated evaluation function. This function and its particularity of no possible comparison between haplotypes of different size lead us to design a multi-populations genetic algorithm. We have implemented cooperative operators (mutations and crossovers) that allow individual of different sizes to cooperate whose rates are set adaptively and used also a random immigrant strategy. We have shown their performances on real datasets by comparing different combinations of operators and mechanisms. The complexity of the evaluation function has lead us to a parallel implementation of the algorithm which is based on a master / slave model.
Thanks to this method, biologists are able to test different data sets and to formulate hypotheses on genetic factors involved in diseases under study.

References

1. Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell.* Garland Pub, 4 edition, March 2002.
2. L. Davis. Adapting operator probabilities in genetic algorithms. In J. D. Schaffer, editor, *Third International Conference on Genetic Algorithms*, pages 61–69. Morgan Kaufmann, 1989. San Mateo, CA.
3. P.C. Sham et D. Curtis. Monte carlo tests for associations between disease and alleles at hightly polymorphic loci. *Annal Human Genetic*, pages 97–105, 1995.
4. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Mancbek, and V. Sunderam. *PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing,*. MIT PRess, 1994.
5. Vermeersch Grégory. Algorithmes génétiques pour la bio-informatique. Master's thesis, University of Lille 1, LIFL, 2001.
6. Francisco Herrera and Manuel Lozano. Adaptation of genetic algorithm parameters based on fuzzy logic controllers. In F. Herrera and J. L. Verdegay, editors, *Genetic Algorithms and Soft Computing*, pages 95–125. Physica-Verlag, Heidelberg, 1996.
7. T. P. Hong, H.S. Wang, and W.C. Chen. Simultaneosly applying multiple mutation operators in genetic algorithms. *Journal of Heuristics*, 6:439 – 455, 2000.
8. Bryant A. Julstrom. What have you done for me lately? adapting operator probabilities in a steady-state genetic algorithm. In L. J. Eshelman, editor, *Proceedings of the sixth International Conference on Genetic Algorithms*, pages 81–87. Morgan Kaufmann, 1995. San Francisco, CA.
9. Department of Energy. The human genome program of the U.S. URL : http://www.ornl.gov/hgmis/, 2002.
10. J.D. Terwilliger and J. Ott. *Handbook of human genetic linkage.* Johns Hopkins University Press, Baltimore, June 1994. ISBN: 0801848032.