

Haplotype Inference Using A Genetic Algorithm

Dongsheng Che, Haibao Tang, and Yinglei Song

Abstract—The *haplotype inference* problem is a computational task to infer haplotype pairs based on the phase-unknown genotypes, and is pivotal in the International Hapmap project. The haplotype inference problem is NP-hard, and exact algorithms become infeasible when the problem sizes are big. Genetic algorithms (GA) are commonly used to approximate optimal solutions for NP-hard problems within reasonable computation time. In this paper, we have proposed a simple genetic algorithm formulation for the haplotype inference problem based on the model of parsimony, which aims to resolve the existing genotypes using as few haplotypes as possible. We applied our GA in the real datasets of the human β_2AR locus and APOE locus, and compared the solutions to the experimentally verified haplotypes; we have found that our approach of inferring haplotypes is very accurate. We believe that our GA is a potentially powerful method for robust haplotype inferences.

I. INTRODUCTION

MOST genetic information between two different human individuals are almost identical. However, the DNA sequences vary $\sim 0.5\%$ for unrelated individuals, and more than 10 million DNA variations have been identified so far [1, 2]. These genetic variations between the individuals are the underlying factors responsible for different traits, susceptibilities to diseases and clinical responses [3]. Among the genetic variations, single nucleotide polymorphisms (SNPs) are the most common type [4]. SNP is the variation that a single nucleotide (A, T, C or G) differs at a particular site, and each of the possibilities is called an *allele*. For diploid organisms like human, the genetic compositions might be different for a particular genetic locus since the two alleles inherited from each of the parents might differ. *Haplotype* is then referred to as a set of associated alleles at different sites and usually inherited from one parent as a unit, while a pair of the haplotypes inherited from both parents comprise a *genotype* [5].

Haplotype is often a better and more compact representation than genotype – while there are large number of genotypes, the number of underlying haplotypes may be significantly less. Therefore, the International HapMap

Project was initiated to develop a haplotype map of the human genome [2, 6]. When the haplotype is known, the identification of the SNP alleles at a few sites can associate other polymorphic sites in the proximal region, paving the way to investigate the genetics behind common disease and clinical responses to pharmaceuticals. Experimental methods for detecting both genotypes and haplotypes are available [4, 7-10]. However, most high-throughput sequencing methods for quantifying haplotypes are still inefficient and less cost-effective [7, 11]. Hence, an attractive way is to determine genotypes first, and then infer haplotypes from genotypes computationally. However, since one genotype usually does not uniquely define a pair of haplotypes, computational techniques need to determine which set of haplotypes probably occurs given a set of genotypes.

The haplotype inference problem can be formulated as follows. Suppose we have n individuals' genotype information and the length (the number of polymorphic sites) of each genotype is m . We assume that most genetic sites are bi-allelic without loss of generality. At each site, there are two possible alleles, a and A , comprising three genotypes -- homozygous aa or AA , or heterozygous Aa . In the case of SNP, a or A might be different nucleotides. The input is $n \times m$ matrix consisting of n genotype vectors each size of m . We encode the data in the following way. Each element in this matrix has a value of 0, 1, or 2. If the position is homozygous (aa or AA), it can have the corresponding genotype value 0 or 1. If the site is heterozygous (Aa), it then has a genotype value 2. The output of this problem is a set of n pairs ($2n$ haplotypes) of binary vectors of size m , with each pair of haplotypes corresponding to a genotype.

It is trivial to find that for any genotype vector g , if the site has value 0 (or 1), *i.e.* homozygous, the associated haplotype pair vectors h_1, h_2 must both have the values of 0 (or 1) at that site since there is no ambiguity. For a heterozygous site where g has the value of 2, h_1, h_2 must be different, but the assignment is now ambiguous. For instance, if the observed genotype g is 2021, then the associated haplotype pairs could be represented by (1011, 0001), or (1001, 0011). More generally, for a genotype containing k heterozygous sites, there are 2^{k-1} unique haplotype pairs. The *haplotype inference* problem is to find out the *optimal* combination of haplotype pairs based on n genotypes, based on the objective function.

Two different genotypes can nonetheless share a common haplotype. For instance, if we have g_1 022100 (001100, 010100) and g_2 220200 (010100, 100000), both g_1 and g_2 can share 010100 as a common haplotype. We can see that

This work was supported in part by the Pennsylvania KISZK grant (C000032549).

D. Che is with the Computer Science Department, East Stroudsburg University, East Stroudsburg, PA 18301 USA (phone: 570-422-2731; fax: 570-422-3490; e-mail: dche@po-box.esu.edu).

H. Tang is with the Plant Genome Mapping Laboratory, University of Georgia, Athens, GA 30602 USA (e-mail: bao@uga.edu).

Y. Song is with the Mathematics and Computer Science Department, University of Maryland Eastern Shore, Princess Anne, MD 21853 USA (e-mail: ysong@umes.edu).

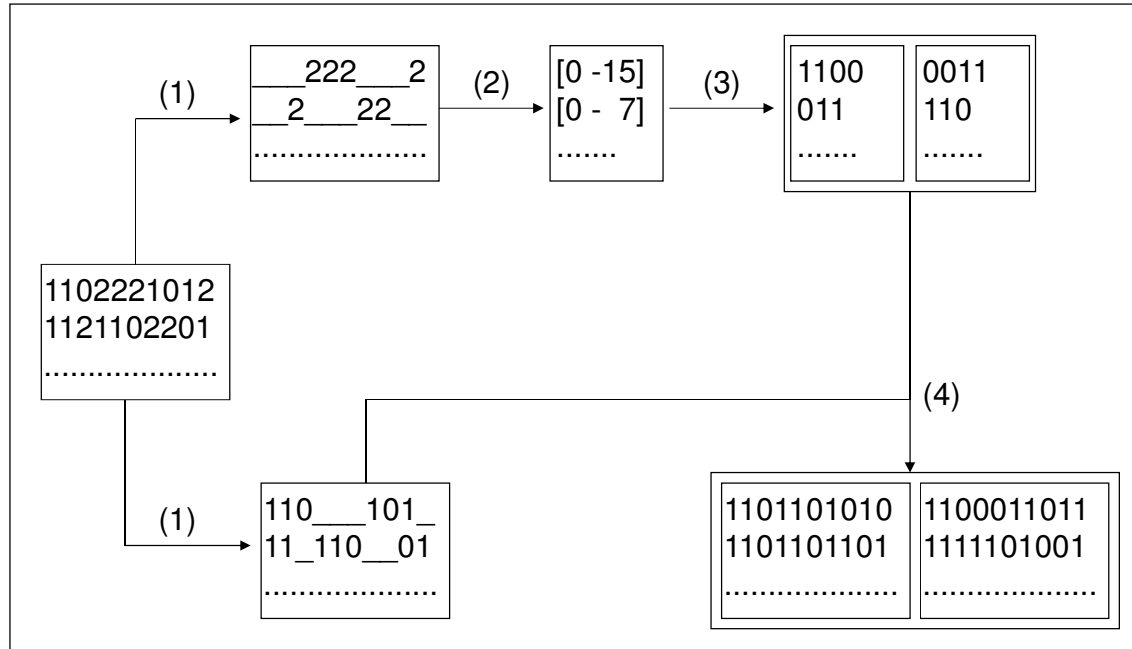


Fig. 1. A schematic view of the haplotyping procedure. In Step 1, values of 2's are extracted from the genotype matrix. The numbers of 2's for each genotype are recorded, thus corresponding ranges for integer representation are determined in Step 2. Candidate solutions generated in the genetic algorithm can be converted into corresponding binary representations in Step 3. The final haplotypes are determined by combining fixed values (0 or 1) from the genotype matrix and generated candidate solutions from step 3.

the number of distinct haplotypes is often less than $2n$ although there are $2n$ haplotypes used for resolving n genotypes. In general, there are $2^{n(k-1)}$ combinations of n haplotype pairs for n genotypes with k heterozygous sites, the problem of finding the minimal number of distinct haplotypes is NP-hard [12].

The difficulty of haplotype inference problem is partly due to the fact that the objective function is not well defined. Some methods are based on *pure parsimony*, i.e., choosing a set of fewest haplotypes to resolve the observed genotypes [11]. Many other algorithms are available to approximate the optimal solution, including rule-based methods [13, 14], combinatorial optimization techniques [15], and statistical inferences based on likelihood functions such as expectation-maximization algorithm (EM), Gibbs sampling or Markov chain Monte Carlo (MCMC) [16-21]. In addition, PHASE implements a Bayesian statistical method to calculate the posterior distributions of the unknown haplotypes [22].

Other criteria for haplotype inference are available. For example, the *perfect phylogeny* criterion assumes that the haplotypes are derived from a common ancestor based on the observation that recombination is rare within the haplotype block [5, 23-25]. However, that is sometimes unsatisfactory as we now know that recombination does exist within the haplotype thus violating the assumption [26]. In this paper, our proposed genetic algorithm (GA) method is intended to solve the haplotype inference problem on the basis of

parsimony.

The remainder of the paper is organized as follows. Section 2 describes the proposed GA approach in detail. Section 3 describes the experimental results from two studies. The paper is concluded in Section 4 with a perspective of future work.

II. METHODS

A. Representation

As mentioned above, we should only deal with the polymorphic sites, or the sites with genotype value of 2. Given a set of m genotype vectors, with each genotype containing k_i number of 2's, an individual can be represented with a list of m integer strings. The integer value at position i is between 0 and $2^{k_i-1} - 1$ since each genotype vector has its own number of 2's. Step 1 of Fig. 1 shows the splitting of 2's from 0 or 1's in genotypes, and step 2 shows the ranges for integer representation used in the GA.

B. Fitness

The major contribution of the fitness value comes from the number of distinct haplotypes, which is the objective we are trying to minimize using the parsimony criterion. However, interpreting these distinct haplotypes based on the integer representation is not straightforward since the representation is only a part of genotype information (step 2 of Fig. 1). In

TABLE I
AN EXAMPLE OF TWO GENOTYPES WITH PARTIAL CANDIDATE SOLUTIONS AND THEIR VALUES FOR THE FITNESS FUNCTION

Genotype	Integer Representation	Binary Representation	Haplotype Pair	Number of Distinct Haplotypes (<i>Hap</i>)	Number of Compatibles (<i>S_{comp}</i>)
$g_1(10221)$ $g_2(12112)$	(0, 0)	(00, 00)	(10001, 10111) (10110, 11111)	4	1
	(0, 1)	(00, 01)	(10001, 10111) (10111, 11110)	3	2
	(1, 0)	(01, 00)	(10011, 10101) (10110, 11111)	4	0

the example given in Fig. 1, 12 and 3 are the candidate solutions (integer representation of the GA) for genotype 1 and 2. Their equivalent binary representations are 1100 for genotype 1 and 011 for genotype 2 (shown in step 3 of Fig. 1). The positions of these four binary digits must match to the positions of 2's in the original genotype, *i.e.*, the first three bits of 1100 must replace the first three 2's of genotype 1 (1102221012) and the last bit of 1100 must replace the last bit of 2 (assuming that the left-most digit is the first bit). Thus, one haplotype can be inferred to be 1101101010 (shown in step 4 of Fig. 1), and the other haplotype is then resolved based on the definition of genotype. After all m pairs of haplotypes have been deciphered as described above, the next step is to count how many distinct haplotypes in $2m$ haplotypes.

In practice, many individuals in the GA population will have the same fitness value if the fitness function only depends on the number of distinct haplotypes. This in turn will make the GA hard to find the best solution in an efficient way. To resolve this problem, we add additional information in the fitness function, which is the total number of *compatibles*. For each inferred haplotype, we can compare it with $m-1$ other genotypes, excluding the genotype that it tries to resolve. We call the haplotype is compatible with the genotype if and only if all 0 or 1 in the genotype perfectly matches the inferred haplotype, otherwise, it is incompatible. For example, haplotype '10001' is not compatible with genotype 12112 since the third and fourth position does not match (00 in haplotype, 11 in the genotype). On the other hand, haplotype 10111 is compatible with 12112. TABLE I shows an example of two genotypes, with three candidate solutions (integer representation), corresponding binary representation and their haplotypes are also shown. Both integer representation of (0, 0) and (1, 0) have the same number of distinct haplotypes (*i.e.*, 4), but the first candidate solution is closer to the optimal solution (the second solution in this example) than the third solution since the first solution is half correct (the first haplotype pair of the first solution matches that of the best solution). We formulate the total number of compatibles as follows:

$$S_{comp} = \sum_{i=1}^m (comp_{i,1} + comp_{i,2})$$

where $comp_{i,1}$ is the compatible number of haplotype $_{i,1}$ with $(m-1)$ other genotypes, and similarly $comp_{i,2}$ is the compatible number of haplotype $_{i,2}$ with $(m-1)$ other genotypes. The parameter m is the number of genotypes.

The fitness function for the GA is formulated as follows:

$$fitness = Hap + \frac{1}{S_{comp} + 1}$$

where Hap is the total number of distinct haplotypes, and S_{comp} is the total number of compatibles. Since the goal is to minimize the fitness, it is a minimization problem.

C. Population Initialization and Parent Selection

The GA population is randomly initialized. 500 individuals are used for the datasets tested in this study.

Parents are selected from the population to build the mating pool for further crossover and mutation. In this study, tournament selection with the size of 2 is used for parent selection, *i.e.*, running a "tournament" between 2 individuals chosen at random from the population, and selecting the best individual for crossover.

D. Crossover and Mutation

Uniform crossover is one of crossover techniques. It creates new generation individuals by randomly selecting sequence elements from either one of the two parent sequences. Uniform crossover is used, with multiple crossover probabilities (0.3, 0.4, 0.5, 0.6, and 0.7) tested.

The most common mutation operator for integer encoding is random resetting, in which a new value is randomly chosen from the set of permissible values. Random resetting is used, with multiple mutation probabilities (0.01, 0.033, 0.667, and 0.1) tested.

E. Replacement Strategy

To prevent the loss of the fittest member of the current population, an elitism strategy is used to keep this best individual in the next generation. Therefore, the GA with an elitism scheme replaces all old individuals with new children, except that the best old individual will remain in the next population. The elitism strategy is used in this study.

TABLE II
SUMMARY OF GA SETUP FOR THE HAPLOTYPE INFERENCE PROBLEM

Representation	Integer strings of length m (the number of genotypes)
Population initialization	Random
Population size	500
Parent selection	Tournament selection with size 2
Crossover	Uniform crossover
Crossover probability	0.3-0.6
Mutation	Random resetting
Mutation probability	0.01-0.1
Replacement strategy	Elitism
Generation	100

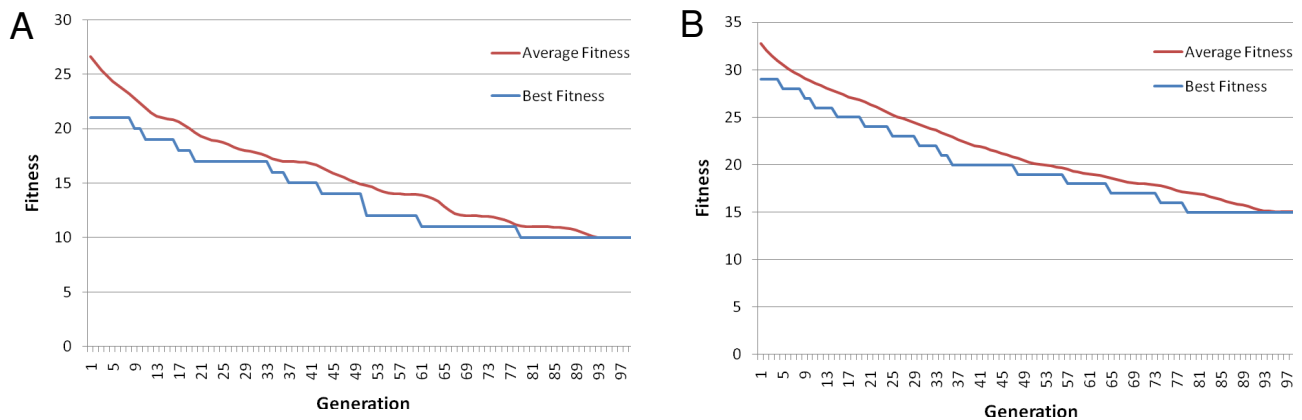


Fig. 2. Fitness values of the best individuals and the whole population throughout 100 generations for (A) β_2AR locus and (B) APOE locus.

In addition, the number of generations is used to terminate the whole process. The overall GA setup for the two studies in this paper is summarized in TABLE II.

F. Performance Evaluation

Results of haplotypes inferred from the GA are evaluated by the *error rate*, which is the proportion of genotypes whose original haplotype pairs (experimentally confirmed) were incorrectly inferred by the computational method [17, 19].

III. EXPERIMENTAL RESULTS

We used our GA algorithm for the haplotype inference problem in two test cases. In both cases, the haplotypes are already known, so we can effectively evaluate the performance and error rate.

A. β_2AR

Our first genotype dataset is the β_2 -adrenergic receptors (β_2AR) locus, which are G protein-coupled receptors that mediate the actions of catecholamines [27]. A total of 13 SNP sites within a span of 1.6kb were found in the β_2AR gene. Out of 121 sampled individuals, 18 distinct genotypes were identified as shown in TABLE III.

Since there are 80 heterozygous sites in all 18 genotypes, there will be 2^{62} (2^{80-18}) possible combinations of haplotype pairs. Hence, it is very time-consuming to exhaustively search the global optimal solution by using exact algorithms which look for the fewest distinct haplotypes. Our GA could find the best solution within one minute on a desktop computer. Fitness values for both the best individual and the whole population throughout of 100 generations are shown in Fig. 2A. The inferred haplotype pairs corresponding to their genotypes are listed in Table III, where 18 genotypes are resolved using 10 distinct haplotypes. We compared our predicted 10 haplotype pairs with those of experimentally verified ones. We found that our inferred haplotype pairs perfectly match the original pairs, achieving a 0% error rate on this dataset. Interestingly, the set of haplotypes was also correctly inferred in HAPAR [11], indicating that the minimum number of haplotypes perfectly explains the genotype samples on this dataset.

B. APOE

We also tested our inference of the haplotypes for a set of 80 genotypes at the human apolipoprotein E (APOE) locus [13, 28]. The experimental results revealed that a total of 17 haplotypes are responsible for the 80 genotypes. We applied

TABLE III
A LIST OF EIGHTEEN GENOTYPES OF B₂AR GENES AND THEIR CORRESPONDING HAPLOTYPE PAIRS INFERRED FROM THE GA.

Genotype No#	Genotype Value	Haplotype 1	Haplotype 2
g_1	20222222000	001000010000	100111101000
g_2	100111101000	100111101000	100111101000
g_3	200222202202	000000000101	100111101000
g_4	001000010000	001000010000	001000010000
g_5	002000020202	000000000101	001000010000
g_6	202222202000	001000000000	100111101000
g_7	002000020200	000000000100	001000010000
g_8	202000010000	001000010000	100000010000
g_9	200000020202	000000000101	100000010000
g_{10}	200222202000	000000000000	100111101000
g_{11}	22222222000	011000010000	100111101000
g_{12}	200222202222	000000000111	100111101000
g_{13}	202222222202	001000010101	100111101000
g_{14}	021000010000	001000010000	011000010000
g_{15}	001000020000	001000000000	001000010000
g_{16}	002000020222	000000000111	001000010000
g_{17}	001000010202	001000010000	001000010101
g_{18}	00000000121	000000000101	000000000111

TABLE IV
SUMMARY OF REAL AND PREDICTED HAPLOTYPES FROM 80 GENOTYPES OF APOE DATASET.

Haplotype ID	Haplotype	Haplotype Representation	Real Frequency	GA Predicted Frequency	PHASE Predicted Frequency
h_1	ATGGGGTCC	000000000	53	52	50
h_2	ATTCGGTCC	001100000	42	42	45
h_3	TTGGGGTCC	100000000	12	13	15
h_4	ATTGAGCCC	001010100	10	10	10
h_5	ACGGGGTCT	010000001	9	7	9
h_6	TTTCGGTCC	101100000	8	6	5
h_7	TTTGGGCC	101000100	5	3	4
h_8	ACTCGGTCC	011100000	5	7	5
h_9	TTGGGGTCT	100000001	3	5	3
h_{10}	ATGGGGTTC	000000010	3	3	3
h_{11}	ATTGGGTCC	001000000	2	2	2
h_{12}	ATGCGGTCC	000100000	2	3	3
h_{13}	ATGGGGCCC	000000100	2	0	0
h_{14}	TTGGGGCCC	100000100	1	3	3
h_{15}	ATTGGGCC	001000100	1	3	2
h_{16}	TTGCGGTCC	100100000	1	0	0
h_{17}	ATTCGATCC	001101000	1	1	1

our GA algorithm on this dataset, and the tracking of the fitness values for the GA populations are presented in Fig. 2B.

The inferred haplotypes we found is a subset of the real haplotypes (numbered h_1 through h_{17}) with two haplotypes h_{13} and h_{16} missing (TABLE IV). Additionally, 5 out of 80 genotypes are incorrectly resolved by our solution, giving an error rate of 6.25%. In comparison, the haplotype reconstruction software PHASE [22] also finds a solution with the same set of haplotypes as our GA algorithm with the same two haplotypes (h_{13} , h_{16}) missing, and has the similar error rate (Table 4).

Interestingly, both our GA solution and PHASE gives a solution that include only 15 haplotypes, two haplotypes less than the haplotypes revealed in the experiment. Based on the criterion of parsimony, our solution is indeed more parsimonious than the real set. This calls into the question of whether parsimony is a truly valid assumption for this particular dataset, and to what extent the assumption is affected by the intrinsic sampling error.

IV. CONCLUSIONS

In this study, a simple GA was applied in the haplotype inference problem. Preliminary tests of the β_2 AR and APOE dataset using this GA shows that our method performs very well on these two datasets. This indicates that the GA has the potential to infer haplotypes correctly and efficiently.

In the future, the reliability of the GA can be further tested on additional genotype datasets. It now appears that the parameters to set up the GA depend on the number of

genotypes and the number of heterozygous sites for each genotype (Data not shown). In particular, we can dynamically determine what values will be suitable for the population size and the number of generations after seeing more datasets. Recently, probabilistic model building genetic algorithm (PMBGA) has been proposed to preserve the building blocks in the population in the crossover and mutation [29]. This new technique might be incorporated into the GA for the haplotype inference problem in our future work.

Furthermore, if the assumed model of maximum parsimony does not fully explain the real data, as seen in the APOE dataset, the fitness function might be changed accordingly. For example, we might calculate the frequencies of the haplotype types, and assign higher weights for those common haplotypes in the fitness function. Alternatively, we can use the consensus methods for the inferred haplotypes generated by our simple GA. The consensus method coupled with the rule-based algorithms has shown to be effective for improving genotype resolution in the APOE dataset [13]. We believe that our GA is a powerful tool to infer haplotypes, and will be more powerful if the underlying model can be more correctly represented.

ACKNOWLEDGMENT

The authors would like to thank Dr. Don Potter at the University of Georgia for his helpful discussions during the course of this project. The authors are also grateful to anonymous referees for their helpful insights during the review process.

REFERENCES

- [1] D. C. Crawford, D. T. Akey, and D. A. Nickerson, "The patterns of natural variation in human genes," *Annu Rev Genomics Hum Genet*, vol. 6, pp. 287-312, 2005.
- [2] D. Altshuler, L. D. Brooks, A. Chakravarti *et al.*, "A haplotype map of the human genome," *Nature*, vol. 437, no. 7063, pp. 1299-320, Oct 27, 2005.
- [3] B. S. Shastri, "Genetic diversity and new therapeutic concepts," *J Hum Genet*, vol. 50, no. 7, pp. 321-8, 2005.
- [4] Y. Suh, and J. Vijg, "SNP discovery in associating genetic variation with human disease phenotypes," *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, vol. 573, no. 1-2, pp. 41, 2005.
- [5] Z. Ding, V. Filkov, and D. Gusfield, "A Linear-Time Algorithm for the Perfect Phylogeny Haplotyping (PPH) Problem," *J Comput Biol*, vol. 13, no. 2, pp. 522-53, Mar, 2006.
- [6] "The International HapMap Project," *Nature*, vol. 426, no. 6968, pp. 789-96, Dec 18, 2003.
- [7] M. Orita, H. Iwahana, H. Kanazawa *et al.*, "Detection of polymorphisms of human DNA by gel electrophoresis as single-strand conformation polymorphisms," *Proceedings Of The National Academy Of Sciences Of The United States Of America*, vol. 86, no. 8, pp. 2766, 1989.
- [8] S. L. Takala, D. L. Smith, O. C. Stine *et al.*, "A high-throughput method for quantifying alleles and haplotypes of the malaria vaccine candidate Plasmodium falciparum merozoite surface protein-1 19kDa," *Malar J*, vol. 5, no. 1, pp. 31, Apr 20, 2006.
- [9] L. Beaudet, J. Bedard, B. Breton *et al.*, "Homogeneous Assays for Single-Nucleotide Polymorphism Typing Using AlphaScreen," *Genome Res.*, vol. 11, no. 4, pp. 600-608, April 1, 2001, 2001.
- [10] G. Orti, M. P. Hare, and J. C. Avise, "Detection and isolation of nuclear haplotypes by PCR-SSCP," *Mol Ecol*, vol. 6, no. 6, pp. 575-80, Jun, 1997.

- [11] L. Wang, and Y. Xu, "Haplotype inference by maximum parsimony," *Bioinformatics*, vol. 19, no. 14, pp. 1773-1780, September 22, 2003, 2003.
- [12] R. Sharan, B. V. Halldosson, and S. Istrail, "Islands of tractability for parsimony haplotyping," *Proc IEEE Comput Syst Bioinform Conf*, pp. 65-72, 2005.
- [13] S. H. Orzack, D. Gusfield, J. Olson *et al.*, "Analysis and exploration of the use of rule-based algorithms and consensus methods for the inferral of haplotypes," *Genetics*, vol. 165, no. 2, pp. 915-28, Oct, 2003.
- [14] A. G. Clark, "Inference of haplotypes from PCR-amplified samples of diploid populations," *Mol Biol Evol*, vol. 7, no. 2, pp. 111-22, Mar, 1990.
- [15] J. Neigenfind, G. Gyetvai, R. Basekow *et al.*, "Haplotype inference from unphased SNP data in heterozygous polyploids based on SAT," *BMC Genomics*, vol. 9, pp. 356, 2008.
- [16] J. C. Long, R. C. Williams, and M. Urbanek, "An E-M algorithm and testing strategy for multiple-locus haplotypes," *Am J Hum Genet*, vol. 56, no. 3, pp. 799-810, Mar, 1995.
- [17] T. Niu, Z. S. Qin, X. Xu *et al.*, "Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms," *Am J Hum Genet*, vol. 70, no. 1, pp. 157-69, Jan, 2002.
- [18] J. S. Liu, C. Sabatti, J. Teng *et al.*, "Bayesian analysis of haplotypes for linkage disequilibrium mapping," *Genome Res*, vol. 11, no. 10, pp. 1716-24, Oct, 2001.
- [19] M. Stephens, N. J. Smith, and P. Donnelly, "A new statistical method for haplotype reconstruction from population data," *Am J Hum Genet*, vol. 68, no. 4, pp. 978-89, Apr, 2001.
- [20] L. Excoffier, and M. Slatkin, "Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population," *Mol Biol Evol*, vol. 12, no. 5, pp. 921-7, Sep, 1995.
- [21] D. Fallin, and N. J. Schork, "Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data," *Am J Hum Genet*, vol. 67, no. 4, pp. 947-59, Oct, 2000.
- [22] M. Stephens, and P. Donnelly, "A comparison of bayesian methods for haplotype reconstruction from population genotype data," *Am J Hum Genet*, vol. 73, no. 5, pp. 1162-9, Nov, 2003.
- [23] V. Bafna, D. Gusfield, G. Lancia *et al.*, "Haplotyping as perfect phylogeny: a direct approach," *J Comput Biol*, vol. 10, no. 3-4, pp. 323-40, 2003.
- [24] R. H. Chung, and D. Gusfield, "Perfect phylogeny haplotyper: haplotype inferral using a tree model," *Bioinformatics*, vol. 19, no. 6, pp. 780-1, Apr 12, 2003.
- [25] D. Gusfield, "Haplotyping as perfect phylogeny: conceptual framework and efficient solutions," *Proceedings of the Sixth Annual Conference on Computational Biology*, pp. 166-175, 2002.
- [26] D. Posada, and K. A. Crandall, "Intraspecific gene genealogies: trees grafting into networks," *Trends In Ecology And Evolution*, vol. 16, no. 1, pp. 37-45, Jan 1, 2001.
- [27] C. M. Drysdale, D. W. McGraw, C. B. Stack *et al.*, "Complex promoter and coding region beta 2-adrenergic receptor haplotypes alter receptor expression and predict in vivo responsiveness," *Proc Natl Acad Sci U S A*, vol. 97, no. 19, pp. 10483-8, Sep 12, 2000.
- [28] S. M. Fullerton, A. G. Clark, K. M. Weiss *et al.*, "Apolipoprotein E variation at the sequence haplotype level: implications for the origin and maintenance of a major human polymorphism," *Am J Hum Genet*, vol. 67, no. 4, pp. 881-900, Oct, 2000.
- [29] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A Survey of Optimization by Building and Using Probabilistic Models," *Computational Optimization and Applications* vol. 21, no. 1, pp. 5 - 20, 2002.