Computational Methods for Haplotype Inference with Application to Haplotype Block Characterization in Cattle

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Rafael Villa Angulo
Master of Science
Center of Scientific Research and Higher Education of Ensenada, 2001

Director: John J Grefenstette, Professor
Department of Bioinformatics and Computational Biology

Spring Semester 2009
George Mason University
Fairfax, VA

DEDICATION

This dissertation is specially dedicated to my mother Bertha Alicia, who has given my all her love and support. Who has taught me the real value of family and education in life. Without her there is no way I could possible have accomplished this. “Mamá lo hemos logrado.”

To my brothers Carlos and José Ramón, who have been my partners along the way in my life. For being always there giving me valuable advice and reminders when I had a hard time understanding.

To my uncle Abel, who has been the paternal guide in my life. For letting me to look at his honesty, humility, and humanity, and take him as the standard to follow in order to succeed in my life.

To Gabriela for all her love and support along this PhD experience. To Karina for taking care of my house. And to all my family and friends who are an important part of my life and have shared their caring thoughts.

ACKNOWLEDGEMENTS

It has been an exceptional journey – one that I had never dreamed of. I am the most grateful to Dr. John Grefenstette for agreeing to be my advisor and permitting me to be part of his group. Under his guidance I have been learning from a great scientist and an exceptional human being. Thanks Dr. Grefenstette for letting me to learn from your experience, and for putting me in the way of accomplishing one of my great goals in life, becoming a real scientist.

I would like to especially thank Dr. Lakshmi Kumar for all his advice and comments during my dissertation process. His dedication and courage for acquiring knowledge has taught me that hardworking is one of the keys for becoming a great scientist. Thanks to Dr. Curt P. Van Tassell for agreeing to be part of my dissertation committee, and for all corrections and comments made on my presentations. They have help to clarify and correct fine details in my dissertation work. Thanks to Dr. Donald Seto and Dr. Patrick Gillevet for accepting to be part of my dissertation committee and for all comments made on my presentations, and to Dr. Clare Gill for all comments and corrections on the manuscript of the paper published in BMC Genetics journal.

Finally, I would like to thank the Bovine HapMap Consortium for providing the data used in this dissertation project. Thanks to the Fulbright Foundation, LASPAU, the University of Baja California, and the NRI grant 2007-35605-17870 for supporting me at George Mason University and making this great experience possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

COMPUTATIONAL METHODS FOR HAPLOTYPE INFERENCE WITH APPLICATION TO HAPLOTYPE BLOCK CHARACTERIZATION IN CATTLE

Rafael Villa Angulo, PhD

George Mason University, 2009

Dissertation Director: Prof. John J Grefenstette

Genetic haplotype analysis is important in the identification of DNA variations relevant to several common and complex human diseases, and for the identification of Quantitative Trait Loci genes in animal models. Haplotype analysis is now considered one of the most promising methods for studying gene-disease and gene-phenotype association studies. In this dissertation, we address the problem of haplotype inference from cattle genotypes, which has significant differences with human genotype data. Using data derived by the International Bovine HapMap Consortium, we provide the first high-resolution haplotype block characterization in the cattle genome. In addition, a new genetic algorithm method for haplotype inference in large and complex pedigrees was developed.

Novel results indicate that cattle and humans share high similarity in linkage disequilibrium and haplotype block structure in the scale of 1-100 kb. Effective

populations size estimated from linkage disequilibrium reflects the period of domestication ~12,000 years ago, and the current bottleneck in breeds during the last ~700 years. Analysis of haplotype block density correlation, block boundary discordances, and haplotype sharing show clear differentiation between indicus, African, and composite breed subgroups, but not between dairy and beef subgoups. Our results support the hypothesis that historic geographic ancestry plays a stronger role in explaining genotypic variation, and haplotype block structure in cattle, than does the more recent selection into breeds with specific agriculture function.

Another significant contribution from this dissertation is the development of new method for haplotype inference in large and complex cattle pedigrees. A new representation of the search space for valid haplotype configurations was developed, and a genetic algorithm was used to optimize features of the haplotype assignments. The genetic algorithm includes a novel population initialization method, new crossover and mutation operators, and a fitness function that minimizes the inferred recombinations in the pedigree. The new method outperformed the current available methods capable of handling large and complex pedigrees, and has the advantage of being scalable to larger datasets.

1 Introduction and motivation

1.1 Introduction

Sequencing the human genome provided the starting point for understanding the genetic complexity of man and how genetic variation contributes to diverse phenotypes and diseases. In concert with the rapid expansion of detailed genomic information, a technological revolution for capturing genetic information from DNA samples emerged. High-throughput genotyping technologies (HTGTs) are one of such technologies that have played an invaluable role in capturing genetic diversity and heritable variations among individuals. The availability of HTGTs have permitted the resequencing and genotyping of additional species serving as model organisms for resolving the genetic complexity of human evolution and to effectively extrapolate genetic information from comparative medicine (veterinary) to human medicine.

The bovine genome sequencing project represents a significant new application for HTGTs in breed characterization and genomic selection [1, 2]. In addition to serving as disease model organism, the bovine genome sequencing project has different goals from other model species (mouse/rat/dog/chimp) sequencing projects; it is the first livestock animal to be sequenced with goals that are particular relevant for agriculture, i.e., increasing food productivity and improving animal health by application of genome-

based approaches [3]. HTGTs offer the potential to be used for detecting genes underlying economically important traits (i.e., quantity and quality of milk production) and improve the current dairy cattle breeding schemes which rely on progeny testing to assess the genetic value of bulls [4]. Progeny testing would not be necessary if markers were available that explained a substantial fraction of the genetic variance [5].

The importance of cattle genome sequencing relays in the strong relationship of coexistence they have hold with humans through the history of modern civilization. Cattle have served as valuable sources of food, draft power, dung for fertilizer and fuel, and leather hides [6]. In recent history, with the emergence of genomic era, genetic and genomic analyses are making possible the characterization of cattle genetic structure, permitting high resolution mapping for Quantitative Trait Loci and gene-disease associations [7-11]. Perhaps the most remarkable and promising applications of genomic information analysis in livestock is genomic prediction and selection [12, 13], by which individual animals are evaluated and selected for specific uses (i.e., crossing for reproduction). The accuracy of all mentioned analyses depends strongly on the existence of accurate haplotype information from the individuals being analyzed. These reasons make haplotype inference a very important factor for cattle products improvement.

In 2004, an international Bovine HapMap project was initiated as a component of the whole genome sequencing effort. The main objectives of this international initiative were: (1) to discover single nucleotide polymorphisms (SNP); (2) validate at least 20,000 SNP by genotyping a panel representing diverse Bos taurus and Bos indicus breeds; (3) use the genotypic data to infer common haplotypes; (4) estimate linkage disequilibrium;

and (5) examine diversity among breeds [14]. By the beginning of this dissertation project, objectives one and two were accomplished resulting in 118,000 putative SNPs discovered from the alignment of a reference sequence (Hereford Dominette) with shotgun sequences generated from other six cows (Angus, Brahman, Holstein, Jersey, Limousin, and Norwegian Red breeds). In addition, genotype panels had been obtained from 20-50 individuals from each of 19 different breeds. It was the pertinent time to proceed with the remaining objectives since the genotypic data were available for all participants in the international Hapmap initiative, and the third whole genome sequence assembly was already completed.

1.2 Problem statement

The general objective of this work is to identify the most appropriate methods to infer haplotypes from the available genotype data from cattle, and to characterize the haplotype block structure based on patterns of linkage disequilibrium within different cattle breeds.

For achieving this goal, two important aspects need to be analyzed. First, the majority of the software tools for inferring haplotypes and analyzing patterns of linkage disequilibrium have been developed based on the specific genetic structure of humans. Second, these tools have been implemented to manage small or moderate amounts of data. There are several significant differences between bovine and human genotype data. The effective population size in bovine (particularly in dairy breeds) is very limited due to extensive selection and the widespread use of a few historic sires. Cattle have complex

pedigrees due to artificial insemination (AI) with few selected bulls that have their offspring extended over several generations. Furthermore, the distribution of human genotype data is considerably different from the bovine genotype data due to differences in marker density and number of animals genotyped. In the human HapMap project, genotyping was performed on a small number of individuals from each geographical region, but the number of SNPs genotyped includes over one million sites. In sharp contrast, the bovine genotype data will be from a 50K chip, covering from 250 to 2500 animals in a breed. Hence the algorithmic optimizations performed in analyzing the human data may not be applicable to the bovine dataset. These differences in the nature of data suggest an urgent need to evaluate, adapt and develop new computational tools for handling large data sets of bovine genotypes, and to provide haplotype inference tools specific to the needs of bovine data characteristics.

1.3 Review of computational methods for haplotype analysis

HTGTs have proliferated mostly because the low cost and short operation time, compared to several bio-molecular methods when used for genotyping large-scale DNA samples, and their ability for dense polymorphism discovery in variation studies [15, 16]. However, due to intrinsic characteristics, HTGTs cannot distinguish the source chromosome of each allele. They simply associate the two alleles to the SNP position, producing genotypes. For that reason a complementary process is necessary to elucidate the haplotypes (DNA strings inherited by each parent). The interest in haplotypes analysis in variation studies has been increasing in recent time. They have been

successfully applied to the identification of DNA variations relevant to several common and complex human diseases [7-11], for the identification of Quantitative Trait Loci genes in animal models [17], and recently for genomic prediction in livestock animals [12, 13]. Haplotype anaysis is now considered one of the most promising methods for studying gene-disease and gene-phenotype association studies [18-21].

Haplotype phasing refers to the computational process of deducing haplotypes from genotypes data. Numerous computational and statistical algorithms have been developed for addressing the haplotype phasing problem. We can categorize them into five different approaches: (1) parsimony; (2) phylogeny; (3) maximum-likelihood; (4) Bayesian inference; and (5) Genetic Algorithms based methods. The first two are combinatorial methods; they generally state an explicit objective function that one tries to optimize in order to obtain a solution to the inference problem. The next two are statistical methods; they are usually based on an explicit model of haplotype evolution, and the inference problem is then cast as a maximum-likelihood or Bayesian inference problem [22, 23]. The last is an emergent approach adopted from Artificial Intelligence. It is similar to parsimony approaches in that it attempts to optimize an objective function, but it differs in that it is mainly focused in applying a directed stochastic search in a landscape of candidate configurations and produces a set of feasible solutions. In the next section, we introduce each of the five major Haplotype Phasing Approaches.

1.3.1 Parsimony methods

Parsimony-based approaches assume that a target population shares a relatively small number of common haplotypes due to linkage disequilibrium. Thus, they try to resolve ambiguous genotypes using already identified haplotypes.

The first implemented algorithm was proposed by Clark [23]. The algorithm starts by identifying any genotype vectors with zero or one ambiguous site, since this vectors can be resolved in only one way. These haplotypes are called the *initial resolved haplotypes*. For resolving the remaining ambiguous genotypes, Clark proposed the following rule that infers a new resolved vector *NR* from an ambiguous vector A and an already resolved genotype vector R:

> *Suppose A is an ambiguous genotype vector with h ambiguous sites and R is a resolved vector that is a haplotype in one of the* 2^{h-1} *potential resolutions of vector A. Then infer that A as the conflation of one copy of resolved vector R and another (uniquely determined) resolved vector NR. All of the ambiguous positions in A are set in NR to the opposite of the entry in R. Once inferred, vector NR is added to the set of known resolved vectors, and vector A is removed from the set of ambiguous vectors.*

Clark's algorithm for resolving the set of genotypes is to first identify the initial resolved set, and then repeatedly apply the Inference Rule until either all of the genotypes have been resolved, or no further genotypes can be resolved.

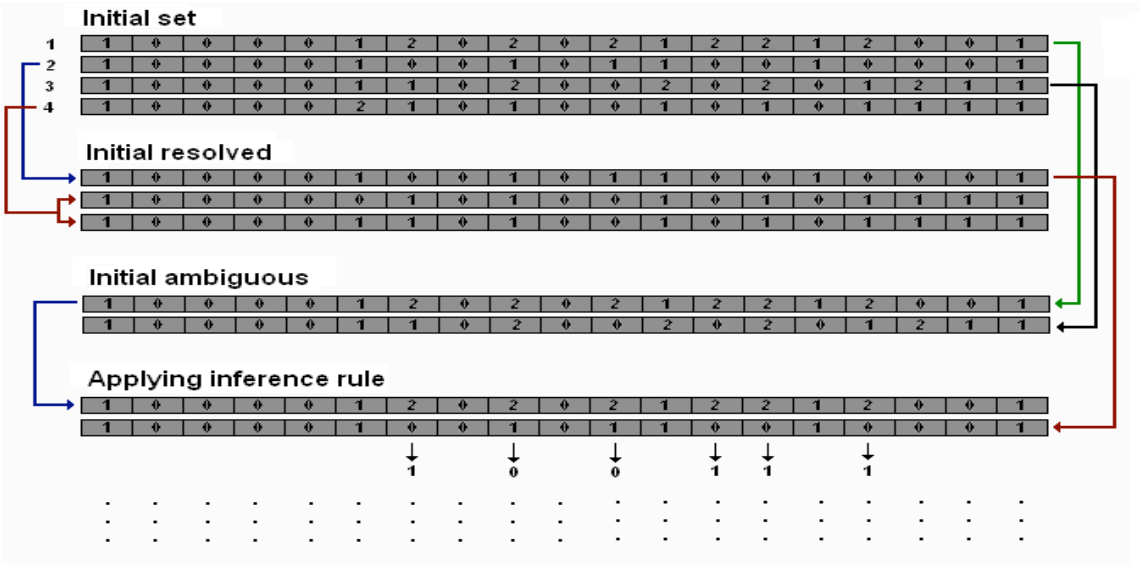


Figure 1.1 Application of Clark's inference rule. The picture shown the initial set, the initial resolved, and the initial ambiguous subsets.

Figure 1.1 shows an example of the application of Clark's rule. At the top of the figure is the initial set from which the initial resolved and ambiguous subsets are elucidated. In the bottom, one member from the resolved and one from the ambiguous subsets are compared in order to elucidate a new haplotype.

Clark's algorithm is simple, intuitive, and has been known to work well in practice [23]. However, it has several limitations: (1) it requires at least one unambiguous genotype, otherwise the algorithm does not start, (2) genotypes may remain unresolved at the end of the procedure; and (3) a different order of iteration may yield a different set of haplotypes. Clark demonstrated through simulation samples that the two first limitations can be overcome if the size of the initial genotype set is large enough. To address the third limitation, he proposed to run the algorithm several times and select the solution

that resolves the largest number of genotypes. This criterion is referred as *maximum-resolution*.

The *maximum-resolution* criterion has been extensively analyzed by Gusfield [24] who proved that it is an NP-hard problem. In addition, several other groups have analyzed the problem from the perspective of finding the minimum set of haplotypes that can resolve all genotypes in a data set [25-28], calling the problem as *maximum-parsimony* (MP) or *pure-parsimony* (PP) problem. However this approach assumes that the observed number of distinct haplotypes in a population is much smaller than the possible number of distinct haplotypes under linkage disequilibrium. Therefore, when the data does not satisfy this condition, the performance of the parsimony-based methods becomes poor [22, 28].

Even with the great efforts that have been made to optimize parsimony methods, to date there is no complete satisfactory solution for the Clark's rule limitations, and they remain as open problems.

1.3.2 Phylogeny methods

Phylogeny methods assume that haplotypes in a population evolved along the coalescent, a rooted tree describing the evolutionary history of a set of DNA sequences. This methods thus aim to find haplotypes that resolve target genotype data and follows the coalescent model as well.

A coalescent is a stochastic process that provides an evolutionary history of a set of sampled haplotypes. This history of the haplotypes is represented as a directed, acyclic graph, where the lengths of the edges represent the passage of time, in number of

generations [22]. In the haplotyping problem we ignore time, so we are only concerned with the fact that the history is represented by a directed, acyclic graph. The perfect phylogeny assumes no recombination. Hence, if we trace back the history of a single haplotype *H* from a given individual *I*, we see that haplotype *H* is a copy of one of the haplotypes in one of the parents of individual *I*. It does not matter that *I* had two parents, or that each parent had two haplotypes. The backward history of a single haplotype in a single individual is a simple path, if there is no recombination. That means the histories of two sampled haplotypes (looking backward in time) from two individuals merge at the most recent common ancestor of those two individuals.

There is an additional element of the basic coalescent model: the *infinite-site mutation*. This assumption states that, at each SNP site, a mutation only occurs once in the evolutionary history. Therefore, a chromosome with mutation at one SNP site must be a descendent of the ancestral chromosome in which the mutation originally occurred. Moreover, any chromosome without this mutation cannot be a descendant of a chromosome that has the mutation.

A *perfect phylogeny* [15] is a computational term referring to a coalescent tree of haplotypes. Let *H* be a set of 2*n* haplotypes $H = \{h_1, \ldots, h_{2n}\}$, where each haplotype h_i consists of *m* SNPs. A perfect phylogeny is defined as a rooted tree *T* with 2*n* leaves that satisfies the following properties:

1. Each of the 2*n* haplotypes labels exactly one leaf of *T*.
2. Each of *m* SNPs labels exactly one edge of *T*.

3. Every internal edge (i.e., one not connected to a leaf) is labeled by at most one SNP.
4. For any haplotype h_i, SNPs labeled on the path from the root to the leaf specify the SNPs whose allele is mutated (i.e., minor) in h_i.

Figure 1.2a shows a perfect phylogeny for a set of four haplotypes. In general, the root of a phylogeny is always assumed to be a haplotype whose alleles are all major (i.e., all 0's). A set of haplotypes has a perfect phylogeny *if and only if* for each pair of SNPs, there are no three haplotypes with values (0,1), (1,0), and (1,1) [29]. Figure 1.2b illustrates a violation of this condition. Haplotype 1, (1,0), has a mutation at the first SNP site, while haplotype 2, (0,1), has a mutation at the second SNP site. Thus, they cannot be descendant of each other, and the two internal edges that denote the mutations at the first SNP and at the second are drawn. Haplotype 3, (1,1), has mutations at both SNP sites, thus it should be the descendant of the subtree that either haplotype (1,0) or (0,1) belongs to. However, to make haplotype 3 belong to either subtree, another edge denoting the mutation at either the first SNP or at the second should be added to the respective subtree. This violates the infinite-site-mutation assumption, that is, at each SNP site, a mutation can occur only once.

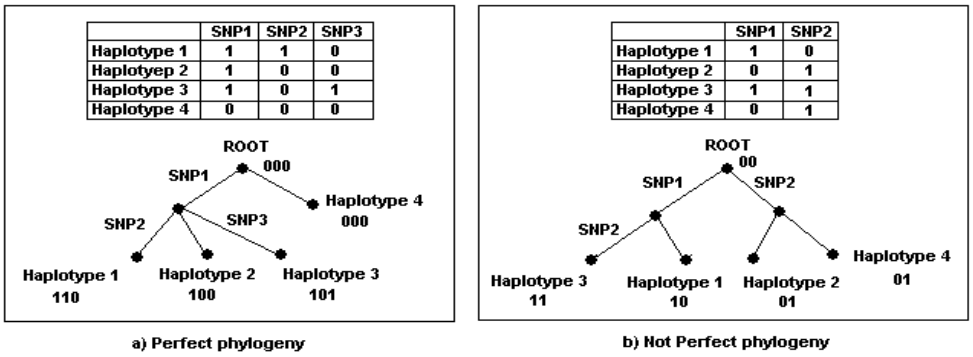


Figure 1.2. Perfect and imperfect phylogeny.

Different methods have been proposed to address the perfect and imperfect phylogeny [30-33]. However, although the performance of the perfect-phylogeny based methods has improved, all of them suffer from their strict conformity to the coalescent model; it is possible that no perfect phylogeny solution exists for a given data set. And if it exists, building the graph becomes a bottleneck due to excessive time required to reconstruct the correct tree. Actually constructing the graph in a reasonable amount of time is still an open problem [31].

Imperfect phylogeny-based methods (e.g., [32]) take a more realistic approach. In principle, the methods assume that most but not all haplotypes will fit the perfect phylogeny model. Thus, they consider a relaxed model that allows for a certain number of recurrent mutations and recombinations. Among multiple candidate solutions satisfying the relaxed model, the one with the maximum-likelihood given a genotype data set is chosen as the solution. However, handling the exponential number of candidate solutions remains an unsolved problem [15].

1.3.3 Maximum-likelihood methods

This approach is based in the idea that it is possible to use maximum likelihood to estimate haplotype frequencies and then to estimate haplotype pairs. It is straightforward to write down the likelihood function associated with any given sample of individuals if one makes an assumption about the process by which mating occurs within the population. The standard and usually reasonable assumption is that there is a process of random mating among individuals. Given this assumption, one can then derive an explicit likelihood function and the goal is to determine its maximum value [22].

Formally, let D be the genotype data of n individuals, where each genotype consists of m SNPs, and the number of distinct genotypes in D is n'. Let g_i denote the i^{th} distinct genotype, and f_i denote the frequency of g_i in the data set D, where $i = 1, ..., n'$. Let H be the set of all haplotypes consisting of the same m SNPs. The possible number of haplotypes in H is 2^m. Let h_j denote the j^{th} distinct haplotype in H, and p_j be the population frequency of haplotype h_j, where $j = 1, ..., 2^m$. Unlike the genotype sample frequencies, f_i, which we can directly calculate from the data set, the haplotype population frequencies, p_j, are unknown, and we need to estimate them.

Maximum-likelihood (ML) methods estimate the population haplotype frequencies, $\lambda = \{p_1, p_2, \ldots, p_2{}^m\}$ based on their likelihood, L, given the genotype data D. Initially, the likelihood, L, can be stated as the probability of genotypes comprising D as:

$$L = P_r(D \mid \lambda) \approx \prod_{i=1}^{n'} P_{r\lambda}(g_i)^{f_i} = \prod_{i=1}^{n'} \Big(\sum_{\{\forall <h_k, h_l> \mid h_k \oplus h_l = g_i\}} P_{r\lambda}(h_k, h_l) \Big)^{f_i} \qquad \text{Equation (1)}$$

In brief, the likelihood of the data D is the product of the probabilities of all genotypes in D. Each genotype g_i occurs f_i times in D, and its probability $Pr_\lambda(g_i)$ can be computed by summing the joint probability of each haplotype pair that can resolve the genotype. Under the assumption of random mating *(Hardy-Weinberg equilibrium* assumption), the joint probability $Pr_\lambda(h_k,h_l)$ of two haplotypes can be computed as the product of the two population haplotype frequencies p_k and p_l. When $k = l$, $Pr_\lambda(h_k,h_l) = (p_k)^2$. Otherwise, $Pr_\lambda(h_k,h_l) = 2p_kp_l$. Thus the joint probability $Pr_\lambda(h_k,h_l)$ can be substituted with the product of two population haplotype frequencies accordingly, and the population frequencies that maximize equation (1) are computed. Using the estimated population frequencies, each genotype can be resolved by the haplotype pair with maximum population frequency among all pairs compatible with the genotype.

For estimating the haplotype frequencies, the Expectation-Maximization algorithm (EM) has been the most successful approach [34-36]. The procedure is defined as follows: Initially, arbitrary values are assigned to the target haplotype frequencies $p_1, \ldots, p_2^n$, which are referred to as $p_1^{(0)}, \ldots, p_2^{n\,(0)}$. In the expectation step, the haplotype frequencies are used to estimate the expected genotype frequency $\hat{P}_{r\lambda}(h_k,h_l)^{(t)}$ where (t) denotes the t^{th} iteration. In the maximization step, the expected genotype frequency $\hat{P}_{r\lambda}(h_k,h_l)^{(t)}$, computed in the previous step, is used to re-estimate the haplotype frequencies $p_1^{(t+1)},\ldots,p_q^{(t+1)}$. The expectation maximization steps are repeated until the change in the haplotype frequency in consecutive iterations is less than some predefined value. The

complexity for one iteration of the EM algorithm is $O(n2^k)$ where n is the number of genotypes, and k is the maximum number of heterozygous SNPs in the genotypes.

The main limitation of the EM algorithm lies in the exponential increase in the number of possible haplotypes as the number of heterozygous SNPs in a genotype grows. Recently, partition-ligation strategies and segmentation based on observed linkage disequilibrium patterns in natural populations have been implemented [35, 36] in order to overcome this problem. However, these algorithms predict haplotype configurations regarding just to specific block-based models, due to the partition-ligation approaches they use, and do not attempt to directly relate observed genetic variation to underlying demographic or evolutionary processes, such as population size and recombination [35].

1.3.4 Bayesian inference methods

Bayesian haplotype reconstruction methods aim to solve the inference problem by regarding the unknown haplotypes as unobserved random quantities and evaluate their conditional distribution in light of the genotype data. They combine *prior information* -- beliefs about what sorts of patterns of haplotypes we would expect to observe in population samples-- with the *likelihood* -- the information in the observed data -- in order to calculate the posterior distribution -- the conditional distribution of the unobserved haplotypes (or haplotypes frequencies), given the observed genotype data [37].

In Bayesian approaches to complicated statistical problems, it is helpful, conceptually, to distinguish two separate issues.

I. The *model or prior distribution* for the quantities of interest, in this case for population haplotype frequencies. For a given data set, different prior assumptions will in general lead to different posterior distributions, and hence to different estimates.

II. The *computational algorithm* used. For challenging problems, the posterior distribution cannot be calculated exactly. Instead, computational methods – typically Markov chain Monte Carlo (MCMC) – are used to approximate it. Different tricks, or different number of iterations, will change the quality of approximations to the Bayesian answer [37].

The most successful method [38], implemented in the software PHASE, uses Gibbs sampling, a type of MCMC algorithm, to obtain an approximate sample from the posterior distribution of H given G, $Pr(H/G)$. The algorithm starts with an initial guess $H^{(0)}$ for H, repeatedly chooses an individual at random, and estimates that individual's haplotypes under the assumption that all other haplotypes are correctly reconstructed. Repeating this procedure enough times results in an approximate sample from $Pr(H/G)$. An improved version of PHASE including considerations of linkage disequilibrium decay and recombination is the most accurate method publicly available to date [35]. In [39], another such method (implemented in HAPLOTYPER software) uses the Dirichlet distribution for sampling along with a model of inheritance where parents and children may be independent of each other.

Bayesian methods are stochastic and each execution of the program may result in different solutions since the derivations are dependent on the initial configuration, which is randomly selected. There are reported situations for which the procedure implemented in PHASE may be under-confident in the estimated haplotypes [35]. This occurs when analyzing small data sets (few markers and/or few individuals), because, for such data sets, the method may tend to overestimate the recombination rates. Despite the better accuracy demonstrated by Bayesian methods, the main problem is in the time consumed for inferring haplotypes. They are on average 10 or more times slower than the other methods.

1.3.5 Genetic algorithms based methods

Genetic Algorithms (GAs) are a particular class of the Evolutionary Computation family of Artificial Intelligence techniques applied to optimization problem solving. GAs tries to mimic the *Natural Selection process* (survival of the fitness) and based on some genetic operators as *selection, mutation*, and *crossover* make a group of candidate solutions to evolve and converge to the real solution of a problem in an iterative way.

The generic GA starts with a set of solutions (represented by chromosomes) called population. Solutions from one population are selected and used to generate a new population. This is motivated by the hope that the new population will be better than the previous one. Solutions, which are selected to form new solutions (offspring) are selected according the their fitness – the more suitable they are the more chance the have to

reproduce. This is repeated until some condition (number of iteration or improvement of the best solution) is satisfied [40]. We can summarize the generic GA in the next steps:

1. **[Start]** Generate a random population of n chromosomes (candidate solutions for the problem).
2. **[Fitness]** Evaluate the fitness $f(x)$ of each chromosome x in the population.
3. **[New population]** Create a new population by iterating the next steps:
    a. **[Selection]** Select two parent chromosomes from the population according to their fitness (the better fitness the bigger chance to be selected).
    b. **[Crossover]** With a crossover probability p_c cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    c. **[Mutation]** With a mutation probability p_m mutate offspring at each locus (position in chromosome).
    d. **[Accepting]** Place new offspring in a new population.
4. **[Replace]** Use new generated population for a further run of algorithm.
5. **[Test]** If the ending condition is satisfied, stop, and return the best solution in the current population.
6. **[Loop]** Go to step 2.

The strength of GAs in optimization problems resides in the good balance of crossover and mutation operators that allow the algorithm to *exploit* the best solutions found so far and at the same time to *explore* the solutions landscape [41]. While mutation allows the algorithm exploit local regions of solutions, crossover allows the algorithm to avoid getting stuck in a local minimum (or maximum) and explore different regions in search for better solutions.

In recent years, GAs have been applied to the haplotype inference problem [17, 42, 43] and they have shown to be a good alternative to overcome the time and space algorithmic complexity affecting the other approaches.

Tapadar et al., [42] implemented a genetic algorithm to infer haplotypes in pedigrees. They use as input to the algorithm: (1) the family structure, (2) the genotypes of every member of the family at each of L loci, and (3) the number of recombinations desired. The GA reconstructs the haplotypes for each individual in the pedigree aiming to obtain the haplotypic configuration with the minimum number of recombinations (the fitness function is defined as the number of recombination events required to explain a specific haplotypic configuration of all members of a family).

The runtime and performance of the algorithm depends on the number of loci and the number of candidate pedigree conformations (population size) considered. With small population sizes the algorithm converges to a local minimum, and with large population sizes the algorithm converges to the global minimum. Some of the problems this implementation has are: (1) occasionally, the algorithm converges to an incorrect solution; (2) the algorithm finds a specific configuration based on the number of

recombination desired, but for each number of recombination there exist several configurations satisfying the pedigree structure; (3) the algorithm does not work with missing data, and (4) the time required to find each solution depends on the initialization of the algorithm variables, therefore, for each different initialization it requires a different time to find the solution.

There are different problems that need to be overcome to make this approach generate more realistic solutions. First, it needs to be modified and fitted to a specific biological model. The fact that just recombination is taken into account, leads the algorithm to ignore patterns as linkage disequilibrium, mutation rate and others that make other approaches generate more accurate solutions. Second, the inclusion of more biological parameters would help the algorithms to converge to the best solution in a more biological-based landscape and not on a pure numerical-based landscape at it does in the current implementation. And third, the convergence of the algorithm needs to be more controlled in order to assure that it always tends to more realistic solutions. The genetic operators (crossover, mutation and selection) need to be modified and optimized to reflect a more realistic behavior.

Another recent implementation of a GA for haplotype reconstruction takes advantage of the potentiality of GAs to parallelize computations and divide the algorithm in multiple CPUs [43]. In addition, this method includes an strategy called joint updating scheme for efficiently update segregation indicators (gene flow). The performance of this method was compared to that of SimWalk2, a method that uses simulated annealing for haplotype reconstruction [44]. Overall, the GA method, using 4 processors, increased the

computational efficiency up to ~8 times compared to SimWalk2. In summary, the results from this work are another promising reason for the use of GAs for haplotype reconstruction.

1.3.6 Haplotype inference in pedigrees

Almost all previously discussed methods for haplotype inference focus on inferring haplotypes in unrelated populations. Recently, given the advantages of high throughput genotyping technologies and the sequencing of human co-evolved species genomes, the analysis of pedigrees of individuals from populations has become of great interest. Haplotype inference based on pedigree data has two fundamental assumptions: (1) the given genotype data has a pedigree structure called pedigree graph. That is to say, the individuals in a population are genetically related; (2) the inheritance satisfies the Mendelian law, i.e. out of two alleles in every SNP site of the genotype of a child, one comes from his paternal genome and the other from his maternal genome, and there is no mutation to occur during the inheritance. One can then get a better estimation of haplotypes because the haplotypes of an offspring are constrained by their inheritance from its parents [45].

Three different models, based on the fact that few recombinations occur when the haplotypes of an offspring inherit from parents, have been proposed in the literature in order to formalize the haplotyping problem in pedigrees. We can enumerate them as follows:

1. *Minimum Recombination Haplotype Configuration* (MRHC). In this model, given a valid genotype pedigree graph *G*, we aim to find a realization *H* of *G* involving a minimum number of recombination events.
2. *Zero Recombination Haplotype Configuration* (ZRHC). In this model, given a valid genotype pedigree graph *G*, we aim to find a realization *H* of *G* involving no recombination events or decide that such realization does not exist.
3. *k – minimum Recombination Haplotype Configuration* (k-MRHC). In this model, given a valid genotype pedigree graph *G*, we aim to find a realization *H* of *G* such that the total number of recombinations is minimal and the number of recombinations on ach parent-offspring pair is at most *k*.

Different strategies have been used for solving the haplotyping problem in pedigree data including the previous models and other searching algorithms [32, 42, 46-51]. Recently a review comparing the most widely used methods [45] shows that, in general, the incorporation of pedigree structure can improve the accuracy for haplotype frequency estimation and haplotype reconstruction, but the run-time of the existing programs increases substantially with an increased number of markers. From this study, it is not possible to give a general conclusion that the existing methods perform well in all cases since the data used in the analysis is from only one species (human) in which the genetic structure is substantially different from other species (e.g. cattle) and the pedigree length is restricted to two or three generations. Therefore, it is of great interest to perform a

more extensive study on larger pedigrees, and to develop more efficient and more accurate haplotyping methods for this challenging situation.

1.3.7 Current work in cattle pedigree haplotyping

Recently, different publications have reported analysis of linkage disequilibrium in cattle inferring haplotypes from genotypes using the two-site Expectation Maximization allele frequency estimator described by [34] or estimating LD parameters directly from genotypes [52-55]. The development of two different software programs based on non-human pedigrees data and Half-Sib families are reported in the literature [51, 56]. The first implements a rule-based haplotype reconstruction method and is aimed specially for analysis of large pedigrees for small chromosomal segments, where recombination frequency within the chromosomal segment can be assumed to be zero. The second implements a Monte Carlo approach for estimation of haplotype probabilities within half-sib (paternal) families, based on multilocus genotypes of the half sibs. None of the articles reports extensive validations or comparison with different approaches for support the results.

Given the rapid growth of genetic information from cattle pedigrees and the lack of efficient methods for handling the haplotype inference, it is evident we need to develop new strategies and provide reference methods for comparison with existing approaches.

1.3.8 Haplotype block characterization

With the completion of the human genome sequence, a great effort was initiated by

different groups in order to characterize the genetic variation among individuals and provide a powerful foundation for gene-disease association studies [57-59]. Haplotype-based methods are to date the most promising approach for doing gene-disease studies [15]. Therefore, a good characterization of haplotype structure in genes affected by disease and regions responsible for phenotypic traits is of relevant importance. In the case of cattle, the situation is not less important given that the association of genetic regions with animal diseases and with quantitative traits, as milk and beef quality, have a direct impact in human health.

Gabriel et al., [59] demonstrated that investigating regions for evidence of recombination and linkage disequilibrium patterns it is possible to parse the human genome into haplotype blocks, and that these blocks share just a few common haplotypes. Gabriel's study provided a solid foundation for the construction of the haplotype map of the human genome. In a latter study, Guryev et al., [60] demonstrated that haplotype block structure is conserved across mammals. In addition, it is possible to use common approaches for characterizing block structure in mammal species. Next, a brief review of two basic concepts (linkage disequilibrium and haplotype block definition) is presented. A complete review can be found in [59, 61].

1.3.8.1 Linkage disequilibrium

Linkage disequilibrium refers to the nonrandom association between alleles. It happens when alleles at two or more loci do not segregate independently, and may indicate a functional interaction between loci associated with a phenotype of interest [62].

Finding linkage disequilibrium patterns from the haplotypes is the base for defining haplotype blocks. The most widely used measure of LD is the square correlation coefficient between SNP pairs. It is defined as:

$$r^2 = \frac{D^2}{p_1 q_1 p_2 q_2}$$

where: $D = p_{11} - p_1 q_1$, linkage disequilibrium coefficient.

p_1 = frequency of Minor allele in SNP1,

q_1 = frequency of Minor allele in SNP2,

p_2 = frequency of mayor allele in SNP1,

q_2 = frequency of mayor allele in SNP2, and

p_{11} = frequency of the observed zygote of both minor alleles among all individuals.

1.3.8.2 Haplotype block definition

Given the r^2 values for all possible SNP pairs in the data, we can define haplotype blocks in different ways. One widely used procedures for defining haplotype blocks is as follows [61]:

a) Block definition based on r^2 values:

1. Begin a block by selecting the pair of adjacent SNPs with the highest r^2 value (no less than $\alpha = 0.4$)

2. Repeatedly extend the block if the average r^2 value between an adjacent marker and the current block members is above β (=0.3) and all individual r^2 values are above γ (=0.1).

Applying the above procedure, a set of haplotype blocks for different populations can be found. Then, the distribution of these blocks, along with the size and other statistics can be computed and compared.

1.4 Open problems from literature review

From this literature review we can state that despite the great effort in solving the haplotyping problem, there still exist many specific details that remain unsolved, and even in specific cases in which some algorithms seem to perform well, they have been tested just with data from one species and is not possible to assume they give general solutions.

Some of the specific problems that remain unsolved are:

1. The haplotyping accuracy of all methods decreases as the linkage disequilibrium drops.
2. Current algorithms do not work well for data sets with moderate amount of genotyping errors or missing alleles.
3. Most haplotyping algorithms show poor phasing accuracy for rare haplotypes.
4. All methods incorporate just one or two aspects of realistic population genetic models. Therefore, there is no method that takes into account most of the parameters that characterize the specific genetic of a population (i.e.,

heterozygosity, mutation rate, recombination, LD decay, inbreeding, diversity, demographic-specific, genetic drift, migration, signature of selection, etc).

5. Pedigree data with dense markers and deep generational genotype data is still unsolved in time and space algorithmic complexity.

There is no doubt that as new species are being sequenced more data for validating and adjusting existing methods is becoming available. At the same time, better strategies for accurately inferring haplotypes from genotype data are required. In addition to improving the accuracy of existing strategies, one of the main problems to solve in the near future is the management of very large data sets. In this thesis, we analyze the specific case of cattle pedigrees, and evaluate the suitability of a genetic algorithm to solve the haplotyping problem, improving time and space algorithmic complexity.

1.5 General and specific objectives of this thesis

As stated previously, the general objective of this work is to identify the most appropriate methods to infer haplotypes from the available genotype data from cattle, and to characterize the haplotype block structure based on patterns of linkage disequilibrium within different cattle breeds.

The methods proposed and developed in this dissertation project will be applied and tested in cattle data. The work is mainly focused on methods for performing the inference of haplotypes and characterization of haplotype block structure based on linkage disequilibrium patterns. The specific objectives that are addressed are:

1. Evaluate alternative methods for haplotype inference in related and unrelated individuals from cattle data.
2. Apply haplotype inference to cattle data, inferring haplotypes and performing a characterization of haplotype block structure based on linkage disequilibrium patterns.
3. Develop an improved method for haplotype inference for cattle.

2 Comparing algorithms for haplotype inference in cattle data

2.1 Introduction

This chapter describes a comparison of the runtime and the similarity of inferred haplotypes of three different algorithms applied to unrelated bovine samples, and provides a brief review of the capability of publicly available software for haplotype inference in a typical cattle pedigree. In the case of unrelated individuals, PHASE, a Bayesian method which implements a coalescent-based model for haplotyping, fastPHASE, which implements a Maximum likelihood strategy on a cluster model for haplotyping, and MERLIN, which implements a likelihood estimations in a phylogeny model for haplotyping, were used to infer haplotypes from two different datasets. One set consists of 157 SNPs from chromosome 5 in 32 Holstein cows, and another set consists of 2,465 SNPs from chromosome 6 in 27 unrelated cows from the Angus breed. In the case of related individuals, HAPLORE, MERLIN, and SIMWALK2 software were applied to a Holstein breed pedigree consisting of 79 individuals, from which 40 are founders. This pedigree structure was obtained from the USDA-ARS Bovine Functional Genomics Laboratory (Beltsville, MD, USA) and its size represents approximately the average size of pedigrees used by USDA in association studies.

2.2 Haplotyping unrelated individuals

The **PHASE** algorithm [37, 38, 63] is a Bayesian approach to haplotype inference that uses ideas from population genetics--in particular, coalescent--based models--to improve accuracy of haplotype estimates for unrelated individuals sampled from a population. The algorithm attempts to capture the fact that, over short genome regions, sampled chromosomes tend to cluster together into groups of similar haplotypes. With the explicit incorporation of recombination in the most recent version of the algorithm [63], this clustering of haplotypes may change as one moves along a chromosome. The method uses a flexible model for the decay of LD with distance that can handle both "blocklike" and "nonblocklike" patterns of LD. Algorithmically, the method uses coalescent theory to assign prior predictions about the distribution of haplotypes. It then uses a Markov-Chain-Monte Carlo algorithm to estimate haplotypes from observed genotypes.

The **fastPHASE** algorithm [35] is also based on the idea that, over short regions, haplotypes in a population tend to cluster into groups of similar haplotypes. But it differs from PHAES in that it defines haplotype clusters based on relative frequencies of alleles. It then uses a Hidden Markov Model to reflect the fact that alleles at nearby markers are likely to arise from the same cluster. Algorithmically, it makes use of likelihood calculations to estimate frequencies and sample pairs of haplotypes from their joint distribution given the unphased genotype data. fastPHASE is faster than PHASE because its computation increases only linearly with the number of individuals and it can be applied directly to unphased genotype data, with unknown haplotypic phases integrated out analytically rather than via a time-consuming and tedious-to-implement MCMC

scheme, as used by PHASE. The disadvantage is that fastPHASE is not well adapted to a realistic genetic model for allele clustering, so results are not guarantied to be the best solution.

The **MERLIN** algorithm [32] is a phylogeny approach based on the idea that patterns of gene flow in general pedigrees can be modeled by sparse inheritance trees. Algorithmically, it first constructs trees describing gene flow pattern for SNP markers. Then it uses the Lander-Green algorithm [64] to calculate likelihoods for all gene flow patterns at arbitrary chromosomal locations. Finally, it finds haplotypes by finding the most likely path of gene flow.

2.2.1 Results and evaluation of inferred haplotypes

Haplotypes for the two datasets were inferred using the three programs, PHASE, fastPHASE, and MERLIN. For the purpose of this test, even when individuals share common ancestors several generations back, for haplotype inference we took them as unrelated. Three different aspects were evaluated: (1) Algorithm runtime, which is time taken by each program to infer haplotypes; (2) Similarity between haplotypes generated by the three algorithms; and (3) Agreement graphs, which consists in finding the most frequently predicted allele for each marker.

2.2.1.1 Algorithm runtime

The time taken by each program to infer haplotypes in both datasets was measured in order to make a comparison of inference speed. Table 1 summarizes the results. The

computer used for this work was an Mac server running OS X, with 2 x 3 GHz Dual-Core Intel Xeon processors, and with 4GB 667 MHz DDR2 FB-DIMM. As we can see from table 1, runtime from fastPHASE and MERLIN are comparable, while PHASE is much slower.

Table 2.1 Runtime for haplotype inference from the three algorithms.

Algorithm	Runtime	
	Set of 154 SNPs	Set of 2,465 SNPs
PHASE 2.1	~ 228 hrs	> 300 hrs (out of time)
fastPHASE 1.14	~ 11 minutes	~ 20 minutes
MERLIN 9.8	~ 9 minutes	~ 15 minutes

2.2.1.2 Haplotype similarity

For computing haplotype similarity, we took, for each individual, the results from two different programs, and counted, comparing marker by marker, how many markers have the same allele assigned. Then we divided by the total of markers. We repeated the comparison for the four possible combinations of haplotype comparisons (two from each individual). Figure 2.1 shows the percent of similarity for all individuals in the Holstein dataset.

Figure 2.1 Haplotype similarity graph from Holstein dataset

From figure 2.1 we observe that the similarity of inferred haplotypes between PHASE and fastPHASE is ~80%. Similarity between fastPHASE and MERLIN is ~70%, and between PHASE and MERLIN is ~65%.

Figure 2.2 shows a graph of the percentage of haplotype similarity between fastPHASE and MERLIN for all individuals in the Angus set.

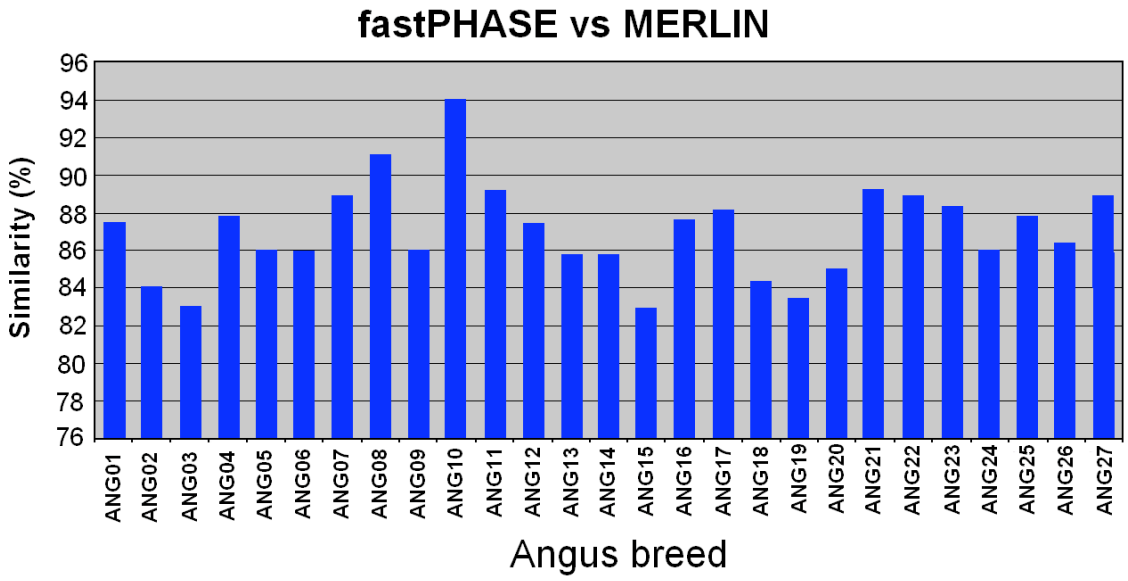


Figure 2.2 Haplotype similarity graph for Angus dataset

From figure 2.2 we observe that the similarity between haplotypes inferred from fastPHASE and MERLIN for the set of Angus is between 82% and 92%.

2.2.1.3 Agreement graphs

The general idea for the agreement graphs is as follows: having identified (from the results) the haplotypes inherited from the mother and from the father corresponding to

each initial genotype, find for each marker, making a consensus from the three algorithms, the allele that appears the most for each parent haplotype. Repeat this for all individuals and make a consensus with all the population haplotypes. Make a graph of marker against number of the most frequently predicted allele.

In our case, since we do not know (from the inferred haplotypes) which haplotype was inherited from the father, and which from the mother, we compared the first haplotype from the inferred pair (we called them first haplotype rows), all second haplotypes (we called them second haplotype rows) and randomly taking either first or second (we called them random haplotype rows), and we obtained three different measures which are plotted in the agreement graphs. An example of how we find the most predicted allele is as follows:

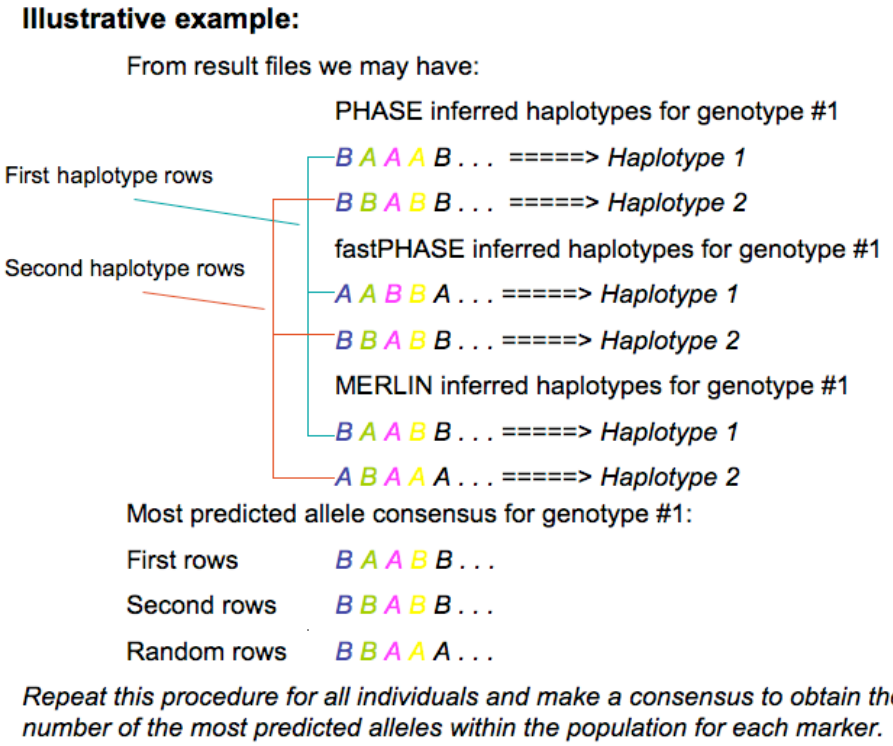


Figure 2.3 shows the agreement plot for the Holstein set, and figure 2.4 shows the disagreement plot for the first 100 SNPs in the Angus set. In both sets, even taking agreement or disagreement, we notice that from the algorithms analyzed, comparing the first row haplotypes, the second row haplotypes, and taking randomly either the first or second row haplotypes, the most predicted allele is very similar times predicted.

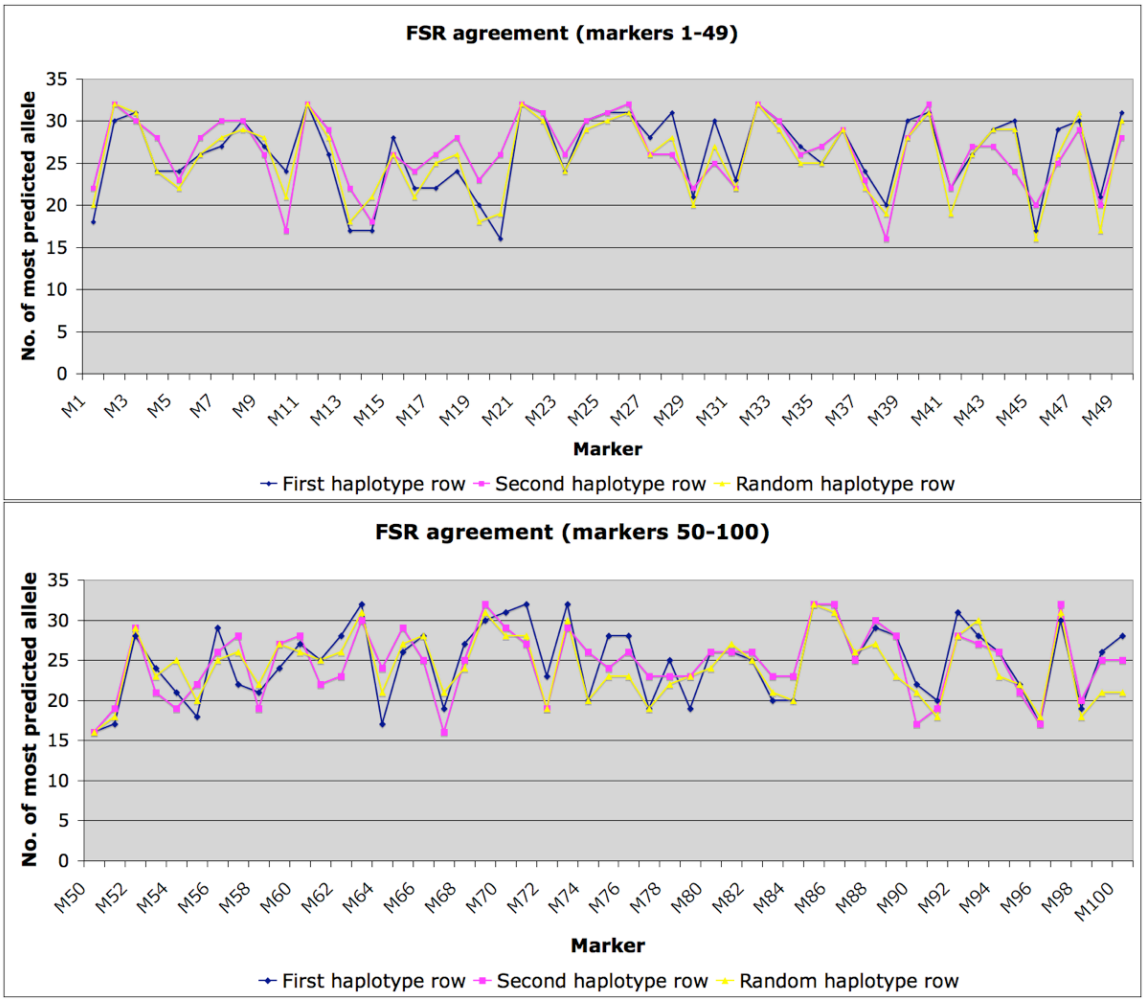


Figure 2.3 Agreement plot for the Holstein set shows that the most predicted allele is very similar times taking either the first, the second and random haplotypes.

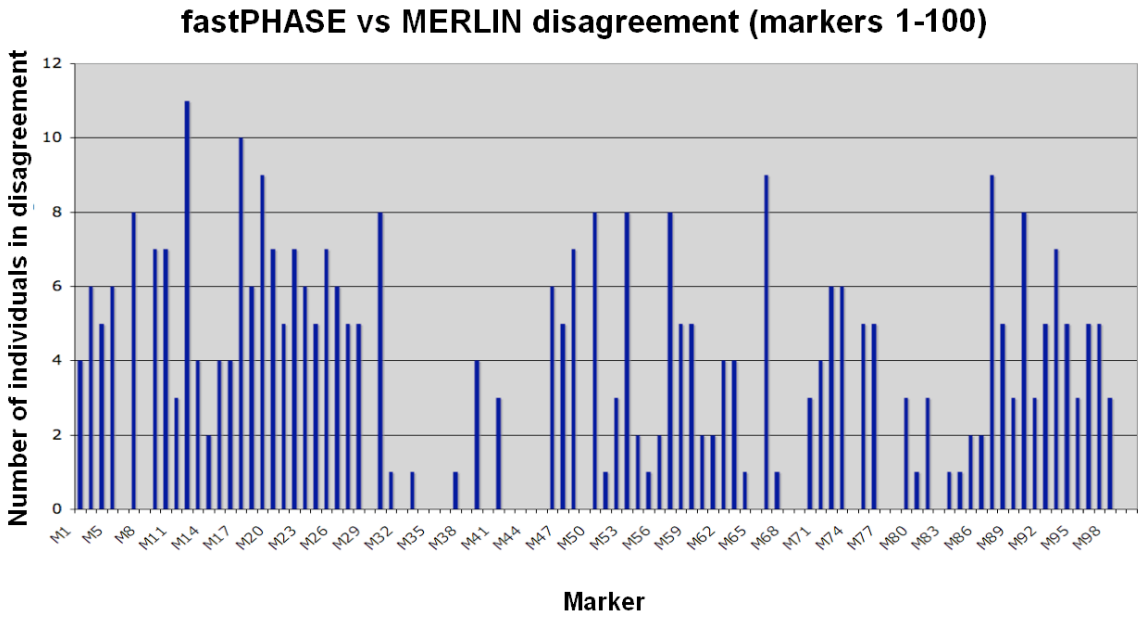


Figure 2.4 Disagreement plot for first 100 markers in the Angus set.

2.3 Haplotyping pedigrees

In the last fifty years, with the refinement of artificial insemination and inbreeding techniques, high selection pressure has been applied to cattle reproduction in the search and maintenance of meat, milk, and fat QTLs. This selection has forced cattle pedigrees to be complex with large inbreeding loops and multigenerational structures. A typical pedigree can have hundreds or thousands members with tens of generation. Due to the high cost and time consuming requirements of molecular methods for haplotyping, genotyping approaches have become the standard methods for capturing the genetic structure in large population samples. This has made computational methods for haplotyping become highly relevant in the process of genetic structure analysis. In the

case of cattle, algorithms need to be able to manage large samples, with a high degree of complexity.

Several software programs for haplotype reconstruction in general pedigrees are publicly available [65], however many have been designed and tested in small or moderate-sized pedigrees. We evaluated three of the most widely used programs for general pedigrees (HAPLORE, MERLIN, and SIMWALK2) and attempted to reconstruct haplotypes for a cattle pedigree consisting of 79 individuals, from which 40 are founders.

2.3.1 Results from cattle pedigree haplotype inference

HAPLORE [46] implements an Expectation-Maximization algorithm to estimate haplotype frequencies. Previous to the application of the EM algorithm, a set of logic rules are applied to the initial genotypes in order to perform a linkage analysis and reduce genotypes and haplotypes to configurations without any recombinations. This limitation made our dataset unsuitable for analysis by HAPLORE, given that it contains recombinations.

The second software we considered was MERLIN [32], described in subsection 2.2. MERLIN was not able to analyze our data either because it is limited to pedigrees containing fewer than approximately 27 non-founders. The reason is that it makes use of the Lander-Green algorithm to calculate likelihood of feasible gene flow paths. This algorithm is of complexity proportional to $m * 4^n$, where m is the number of markers, and n is the number of non-founders in the pedigree. For pedigrees with approximately 27 non-founders, the algorithm becomes computationally infeasible.

The third software we evaluated was SIMWALK2 [44, 66, 67], which turned out to be the only public available software capable of handling large and complex pedigrees. It implements a MCMC algorithm capable of searching valid haplotype configurations in proportion to their likelihood. It makes use of a simulated annealing algorithm [68] in conjunction with a random walk approach [69] to find candidates and develop consecutive solutions to reach the optimal haplotypes. The analysis of performance and haplotypes inferred by SIMWALK2 are presented next:

To evaluate the accuracy of Simwalk2 inferring haplotypes, we simulated a set of 50 SNPs for the 79 individuals Holstein pedigree. We used SIMPED [70], a program to generate haplotype and genotype data for pedigree structures. We assumed a constant recombination rate of 0.005 across all the pedigree. We evaluated three different measures from simwalk2: (1) run time, (2) number of recombinations in the inferred haplotypes compared to the number of recombinations in the real (simulated) haplotypes, and (3) the number of switch errors, computed as the proportion of heterozygote SNPs whos phase is wrongly inferred relative to the previous heterozygote SNP [37]. Table 2.2 presents the results.

Table 2.2 Evaluation of simwalk2 inferring haplotypes for 50 SNPs in the Holstein pedigree.

Program evaluated	Runtime	Number of recombinations	Switch errors
Simwalk2	~10 hrs	600	601

The real haplotypes contained a total of 37 recombinations. Then the difference in recombiantions from the inferred haplotypes compared to the real ones was 563.

2.4 Summary

For unrelated individuals, MERLIN and fastPHASE are fast and comparable while PHASE is very slow. However, the literature reports that PHASE is the most accurate software for haplotype inference in unrelated individuals. We found that PHASE and fastPHASE produce the most similar haplotypes, with an average of ~80% of similarity. From the agreement graphs we can conclude that, regardless of the order in which resulting haplotypes are taken as paternal or maternal, the most frequently predicted allele is consistent. As a final conclusion for the analysis of unrelated individuals, in the case of cattle data which generally consist of large samples in individuals and SNPs, fastPHASE seems to be the most adequate method to infer haplotypes. Even when MERLIN is faster than fastPHASE, it was designed for analysis of pedigrees and computes gene flow trees, which are not present in unrelated individuals. In addition to being fast, fastPHASE produces very similar results to those from PHASE, which has been reported as the most accurate software so far. Of course, when the sample is small and the number of SNPs is not large, PHASE would be preferred over fastPHASE.

For related individuals, the only publicly available software capable of handling large and complex pedigrees typical in cattle datasets appears to be SIMWALK2. However, it is slow and its accuracy has not been extensively tested. For this reason, we address the

challenges to develop a new approach for haplotype inference capable of managing large and complex pedigrees with improved computational and time complexity.

3 High-resolution haplotype block characterization in cattle

3.1 Introduction

The rapid improvement in high-throughput single nucleotide polymorphism (SNP) discovery and genotyping technologies is making possible the availability of many thousands of SNP markers for genome-wide association studies [16, 71-74]. High-resolution linkage disequilibrium (LD) maps and characterizations of haplotype block structure are being generated for different organisms, confirming that elucidating in the fine-scale the structure of LD at the population level is crucial for understanding the nature of the highly non-linear association between genes and phenotypic traits, such as complex diseases and quantitative trait loci (QTL) [17, 75, 76].

Initial studies in humans [57, 59] demonstrated that, by investigating regions for evidence of recombination and LD patterns, it was possible to parse the human genome into haplotype blocks, and that those blocks shared just a few common haplotypes. This result provided impetus for the construction of LD and haplotype maps of the human genome. Furthermore, haplotype block structure appears to be conserved across mammals [60].

Recently, high resolution LD and haplotype block maps were generated for humans using a set of 3.1 million SNPs genotyped in 270 individuals from four geographically diverse populations [77]. Overall, 98.6% of the assembled genome is within 5 kb of the nearest

polymorphic SNP. The analysis of these high-resolution data is helping to infer with great precision, information about population history, recombination and mutation rates, evidence of positive selection, and is providing invaluable information for gene-disease association studies [78].

An initial bovine study [52] reported characterization of haplotype blocks in Holstein-Friesian cattle using a 15K SNP chip with an average intermarker spacing of 251.8 kb. Another study [53] reported haplotype block structure for 14 European and African cattle breeds using 1536 SNPs. This study had an average resolution of 311 kb intermarker distance and was focused mainly on chromosome 3. Recently, the Bovine HapMap Consortium [6] generated an assay of 30K SNPs and genotyped 501 animals sampled from 19 worldwide taurine (*Bos taurus*) and indicine (*Bos indicus*) breeds, plus two outgroup species (Anoa and Water Buffalo). In this chapter we present the characterization of LD and haplotype block structure across 101 high-density targeted regions from the bovine HapMap data, spanning 7.6 Mb of the genome with an average intermarker distance of ~4 kb. The extent of LD is presented along with the estimation of ancestral population size for different generations. In a first level of analysis, haplotype block characterization allowed us to elucidate the breed-specific block structure and its variability compared with all other breeds. In a second level of analysis, haplotype block density correlation, haplotype block boundary comparison, and haplotype sharing between breeds and subgroups helped us to elucidate high-resolution similarities between breeds, and also permitted us to differentiate breeds by geographic separation versus

those related by shared ancestry. Finally, breeds were clustered given computed genetic distances based on haplotype block analysis.

3.2 Bovine Hap Map data

The Bovine HapMap consortium provided the data for this project. The breeds belong to *Bos Taurus*, *Bos ind*icus, and *composite* (hybrid breed composed from crossing two different breeds) breeds. A total of 501 animals from 19 breeds form the data set. Table 3.1 lists the breeds and show number of individuals genotyped. Around 24 animals were genotypes per breeds, with the exception of Holstein, Limousin, and Red Angus where 53, 42, and 12 animals were genotyped, respectively.

Table 3.1 Number of animals per breed in the initial HapMap database

Breed	No. of individuals	Breed	No. of individuals
Charolais	24	Jersey	28
Limousin	42	Norwegian Red	25
Piedmontese	24	Gir	24
Romagnola	24	Nelore	24
Hereford	27	Brahman	25
Angus	27	Beef Master	24
Red Angus	12	Santa Gertrudis	24
Brown Swiss	24	Sheko	20

Guernsay	21	N'Dame	25
Holstein	53	Buffalo and Anoa	2 each

Table 3.2 lists the chromosomes and number markers genotyped. Chromosomes 6, 14 and 25 contain more markers than the rest. In the case of chromosomes 6 and 14 where selected to be denser in markers because evidences of dairy QTLs due to selection pressure to which some breeds have been exposed in the last years. Chromosome 25 was selected as control chromosome due to absence of known QTLs.

Table 3.2 Initial number of markers in the HapMap data.

Chromosome	Markers	Chromosome	Markers	Chromosome	Markers
1	1537	11	1242	21	669
2	1512	12	879	22	698
3	1316	13	999	23	588
4	1320	14	2794	24	694
5	1246	15	851	25	1208
6	2485	16	885	26	602
7	1101	17	828	27	498
8	1224	18	660	28	520
9	1018	19	690	29	479
10	1139	20	882	X	573

3.3 Animal samples

All breeds in this study belong to the taurus and indicus subspecies of *Bos taurus*, and represented several different geographical regions: N'Dama and Sheko are African breeds; Angus, Hereford, and Red Angus are British beef breeds; Charolais, Limousin, Piedmontese, and Romagnola are European beef breeds; Guernsey and Jersey are British dairy breeds; Brown Swiss, Holstein, and Norwegian Red are European dairy breeds; Brahman, Nelore, and Gir are indicus breeds; Beefmaster, and Santa Gertrudis are composites of taurine-indicine origin. Individuals were selected to be unrelated at least for 4-5 ancestral generations, with the exception of 44 trios of sire, dam and offspring included to allow quality control of the data and to assist in the determination of allelic phase relationships. The DNA samples were taken from whole blood or cryopreserved semen.

3.4 Data filtering

For accessing the overall quality of samples and work with a consistent set of genotypes, some filters were applied to the initial data. It is important to mention that this quality control procedure was taken from the International HapMap consortium. The filters included removal of all genotypes that had >20% missing genotypes, that violated Hardy-Weinberg frequency distribution, or that violated Mendelian inheritance. Data were also removed for all animals with genotype completeness <98%, for markers with estimated genotyping error >5% and at least one breed out of Hardy-Weinberg equilibrium, as well as markers that were monomorphic for all breeds, markers with minor allele frequency

<0.05 among all breeds, markers containing >2 discordant trios, and markers assigned to unknown chromosome. After this QC procedure, the data set contained 31,857 markers from 487 animals, and excluded Anoa and Water Buffalo.

In addition to previous QC filters, we removed monomorphic SNPs breed by breed in order to avoid the analysis of uninformative data.

3.5 Selection of high-density regions

In order to facilitate the study of haplotypes extended over multiple markers, we focused on the regions of the bovine genome that had the highest density of markers in the HapMap data set. We focused exclusively on chromosomes 6, 14, and 25, which were selected for additional genotyping due to the presence of known QTL of interest in chromosomes 6 and 14, and the absence of known QTL on chromosome 25. Chromosome 25 therefore served as a control for studies focusing on high-density regions. For this study, we defined high-density regions as non-overlapping genomic windows of 100 kb containing 10 or more markers and a maximum gap between markers of 20 kb. This definition identified 101 high-density regions covering a total genomic distance of 10.1 Mb. The effective region (regions within markers) covered is 7.6 Mb and contains a total of 1,981 markers with an average of one marker each ~4 kb. The average markers per region were 19.61. And, the average distance between adjacent high-density regions on the same chromosome was 1.46 Mb, but they were not evenly spaced. There were 31 instances in which two adjacent high-density regions were contiguous on the chromosome. Table 3.3 presents the results by chromosome.

Table 3.3 Structural details of the 101 high-density regions selected on chromosomes 6, **14, and 25.**

	BTA 6	BTA 14	BTA 25	Summary
High-density regions	30	57	14	101
Markers in regions	547	1228	208	1,981
Average markers per region	18.17	21.54	14.86	19.61
Distance scanned	3 Mb	5.7 Mb	1.4 Mb	10.1 Mb
Effective distance	2,276,304	4,465,915	896,479	7,638,698
Max gap between markers (kb)	19.47	19.7	19.35	19.7

3.6 SNP allele frequencies across population samples in high-density regions

In order to investigate how informative the SNPs occurring in the targeted regions were, we computed the allele frequency distribution and the average minor allele frequency (MAF) across all markers in the targeted regions. Figure 3.1 presents the average by breed, and figure 3.2 presents values by group.

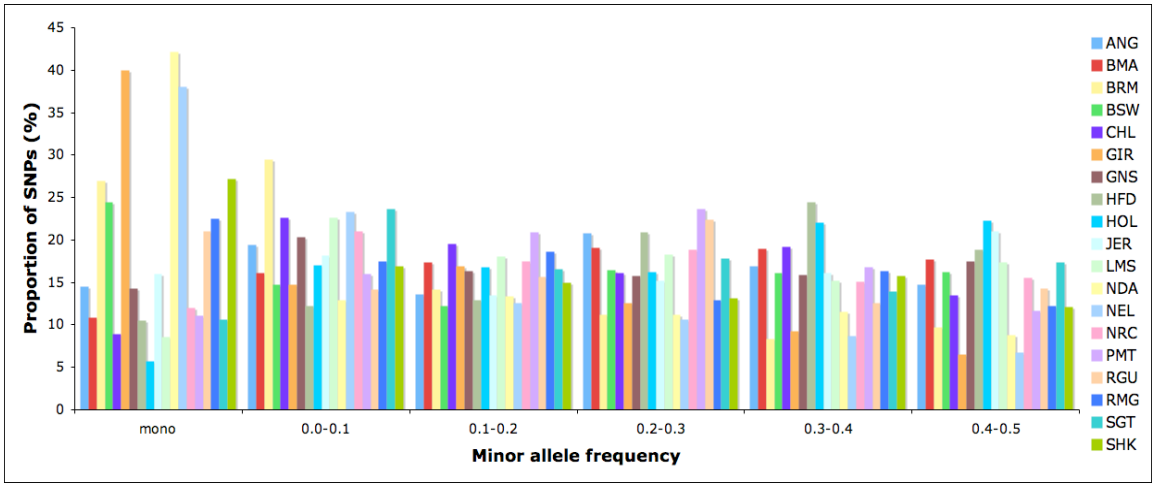


Figure 3.1 Average proportions of SNPs of various frequencies by breed in high-density regions (intervals' upper limit inclusive).

The breeds Nelore, N'Dama, and Gir exhibited the lowest proportion of polymorphic SNPs, between 57% and 62%, compared to the remaining breeds, which exhibited 77% to 95%. Thus a substantial fraction of loci in the targeted regions are informative for all breeds. Figure 3.2 presents all SNPs (including monomorphic and polymorphic SNPs) but for all subsequent analyses monomorphic SNPs were removed from the study.

In general, African and indicine breeds exhibited lower MAF values. It could be thought that this is due to an ascertainment bias in the SNP discovery because all targeted SNPs in this study were originally derived by comparison between a Hereford assembly and sequence reads from a series of bacterial artificial chromosomes (BACs) constructed from Holstein DNA. However, analysis of variation from among the major cattle breeds free from SNP ascertainment bias demonstrated a higher genetic diversity in indicine compared to taurine breeds [6].

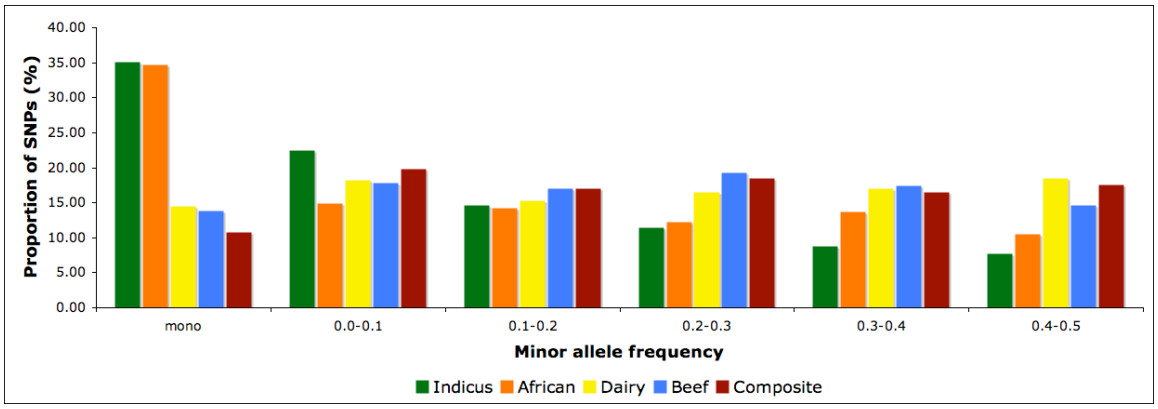


Figure 3.2 MAF distribution in high-density regions. Average proportions of SNPs of various frequencies by cattle group in high-density regions (intervals' upper limit inclusive).

As shown in table 3.4, in the targeted regions, MAF values ranged from a maximum of 0.253 (Holstein) to 0.116 (Nelore), which is a difference of about 28% in the full scale of 0.0 to 0.5. The average decay in MAF between breeds was 1.51%. Furthermore, we compared the proportion of polymorphic SNPs in the selected regions with the proportion of polymorphic SNPs in the entire HapMap data set and found a 20% higher proportion in the complete HapMap data than the selected regions.

Table 3.4 Average minor allele frequencies (MAF) per breed across the high density regions in the study

Breed	Average MAF	Value in the scale 0.0 – 0.5 (%)	Decay with respect to the previous breed (%)
Holstein (Dairy)	0.253	50.6	0
Hereford (Beef)	0.250	50	0.6
Beefmaster (Composite)	0.227	45.5	4.5
Jersey (Dairy)	0.216	43.2	2.3
Limousin (Beef)	0.215	43	0.2
Charolais (Beef)	0.210	42	1
Norwegian Red (Dairy)	0.210	42	0
Santa Gertrudis (Composite)	0.210	42	0
Piedmontese (Beef)	0.209	41.8	0.2
Guernsey (Dairy)	0.208	41.6	0.2
Angus (Beef)	0.206	41.2	0.4
Brown Swiss (Dairy)	0.196	39.2	2
Red Angus (Beef)	0.193	38.6	0.6

Romagnola (Beef)	0.181	36.2	2.4
Sheko (African)	0.180	36	0.2
Brahman (Indicus)	0.140	28	8
N'Dama (African)	0.133	26.6	1.4
Gir (Indicus)	0.125	25	1.6
Nelore (Indicus)	0.116	23.2	1.8

3.7 Linkage Disequilibrium analysis

We used the 1,981 SNPs in the high-density regions to evaluate the extent of pairwise LD as a function of physical distance. A pair of haplotypes was estimated for each animal in the sample using fastPHASE Version 1.2.3 [35]. The LD measure we adopted was the squared correlation coefficient between SNP pairs (r^2), computed as:

$$r^2 = \frac{\left(p_{11} - p_1 q_1\right)^2}{p_1 q_1 p_2 q_2}$$

where, p_1 and p_2 are the minor and major allele frequencies in SNP 1 respectively, q_1 and q_2 are the minor and major allele frequencies in SNP 2 respectively, and p_{11} is the frequency of observing both minor alleles in the same individual across all population.

Figure 3.3 shows the average of r^2 value using bins of 5 kb. Consistent with previous analyses in cattle [76, 79], the decline of LD as a function of distance was rapid, such that r^2 averaged ~ 0.1 at 100 kb. Hereford, Jersey, and Brown Swiss had consistently higher r^2 values relative to the other breeds. In the case of Hereford and Jersey, this result is consistent with a lower resolution analysis (10 kb) previously performed using the same data [6]. In the case of Brown Swiss, the higher resolution inspection permitted us to

elucidate its similarity in LD extent with the two previous breeds. As also shown previously [6], at the smaller distances N'Dama had the highest r^2 values while the *Bos indicus* breeds (Brahman, Nelore, and Gir) had the lowest values. In contrast, analyzing r^2 values at longer distances, Santa Gertrudis and Sheko were the breeds with the highest r^2 values while Angus and Beefmaster were the breeds with the smallest r^2 values. Table 3.5 shows the average r^2 value for each breed, computed as the mean r^2 value across all possible SNP pairs within each targeted region.

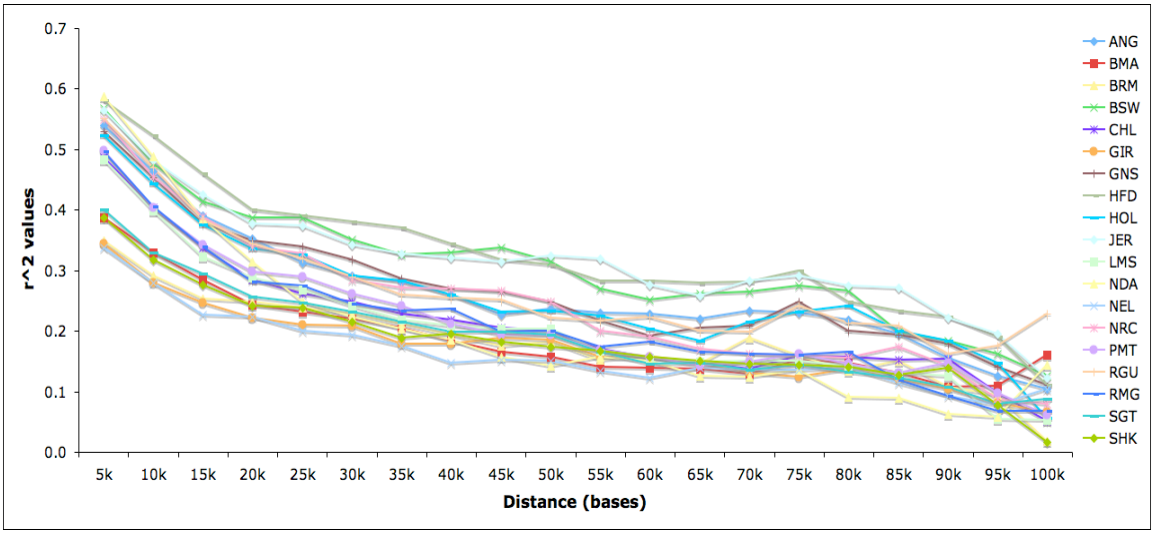


Figure 3.3 LD in high-density regions. LD shows a rapid decline, such that r^2 averages ~0.1 at 100 kb. r^2 values are averaged using bins of 5 kb.

Table 3.5 Total average of r^2 per breed across high-density reions

Breed	r^2 average	Breed	r^2 average
Hereford	0.397	Romagnola	0.283
Jersey	0.380	Charolais	0.278
Brown Swiss	0.377	Limousin	0.274

Guernsey	0.333	Santa Gertrudis	0.246
Angus	0.332	Sheko	0.236
Red Angus	0.330	Beefmaster	0.234
Norwegian Red	0.324	Brahman	0.230
Holstein	0.323	Gir	0.218
N'Dama	0.299	Nelore	0.204
Piedmontese	0.284	Total r^2 average	0.294

3.8 Effective population size estimation

We used the complete set of SNPs (31,857) to estimate the effective population size in the previous 10,000 generations for each breed. This estimation was based on the observation that in a population with constant effective population size *N*, the approximate expectation of r^2 is: $E(r^2) = \frac{1}{4Nc + 1}$, where *N* is the effective population size 1/(2*c)* generations in the past, $E(r^2)$ is the average of r^2 values for all SNPs within a specified range, and *c* is the median of the range in Morgans (we assumed 1 cM ~ 1 Mb) [53, 80-83].

A correction for sampling error was made to all computed r^2 values as:

$$r^2_{corrected} = \frac{r^2_{computed} - \frac{1}{n}}{1 - \frac{1}{n}},$$

where *n* is the number of sampled haplotypes [79].

The results show a persistent decline in effective population size through the period considered, but suggest two distinctive time points (Figure 3.4a). The first distinction is

~2,000 generations ago, at which time all population sizes seem to converge, compared to previous periods. The time associated with this convergence is approximately the early Neolithic period (~12,000 years ago) when domestication of cattle by humans began [6].

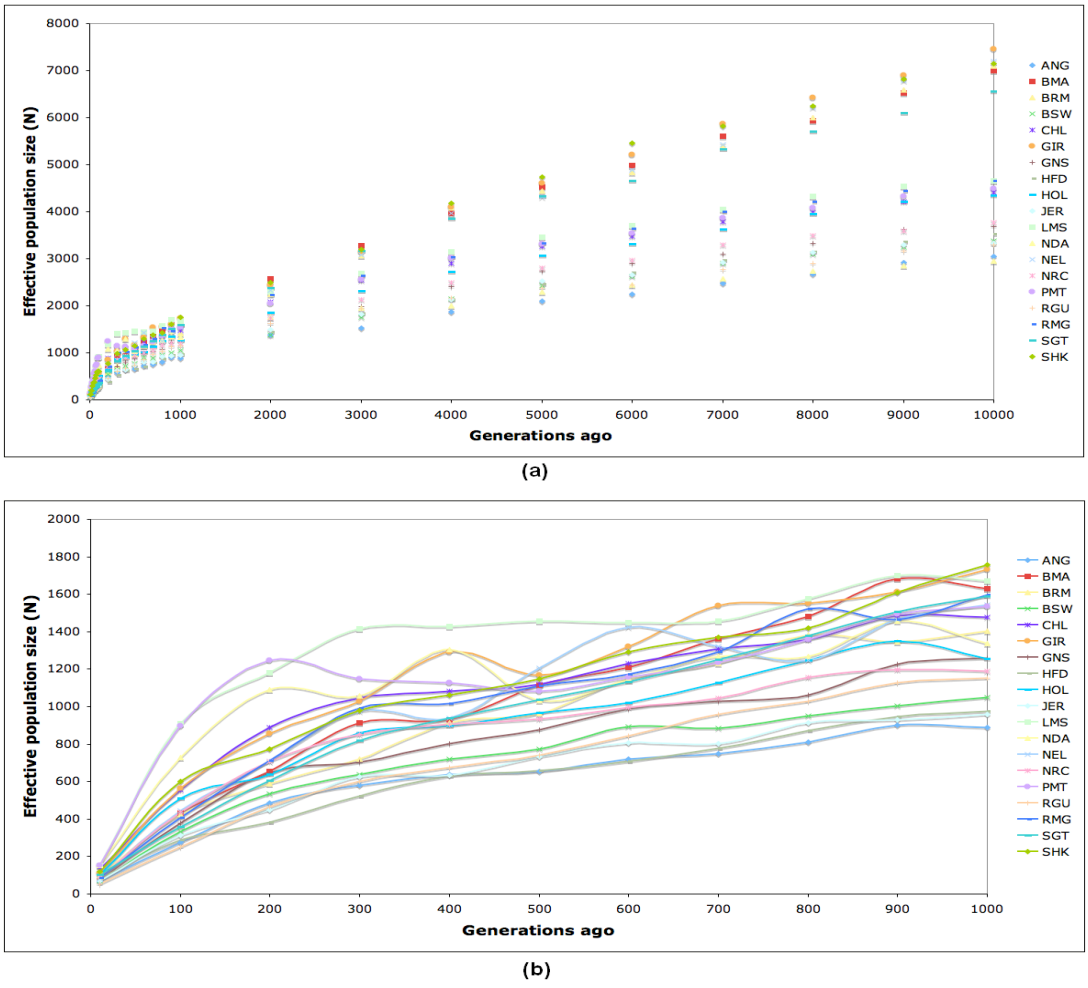


Figure 3.4 Estimated effective population size in previous 10,000 generations. Results suggest two distinct time points: the initiation of cattle domestication ~2,000 generations ago (a), and a population bottleneck in the most recent 100 generations (b).

The second distinctive point is the most recent 100 generations, which show a sharp decline in population size (Figure 3.4b), suggesting that all breeds in this study are experiencing a population bottleneck. Two events may have contributed substantially to this reduction in effective population size: First, approximately 100 generations ago an intensification of population isolation was experienced principally in Europe, starting with the Great Famine of 1315-1322 followed by a series of large scale crises that struck Europe early in the 14th century, which caused significant reductions in the human population due to a great dearth of all victuals, and a dramatic reduction in livestock population sizes mainly due to a plague of murrain [53, 84]. Second, the high selection pressure for specific traits and the use of artificial insemination have reduced dramatically the number of sires within the last ~50 years [82]. The estimated effective population size *N* for the most recent time point (10 generations ago) gave an average value of about 100 individuals across all populations. This result is similar to the average *N* of 116 reported in [6] in an analysis of these same samples. Table 3.6 presents the estimated effective population size for 10, 100, 1000, 5000, and 10000 generations ago for each breed in the study. We recognize that most breeds have originated more recently than 10,000 generations ago, but we assume that the estimates of effective population size in those cases should reflect the average historical population size of their ancestors.

Table 3.6 Effective population size for each breed, estimated from r^2.

Generations Ago	10	100	1000	5000	10000
Angus	64	275	890	2091	3042
Beefmaster	92	432	1629	4525	7008
Brahman	99	424	1402	4439	7095

Brown Swiss	68	335	1048	2430	3382
Charolais	130	554	1478	3263	4404
Gir	112	562	1732	4604	7460
Guernsey	76	378	1259	2737	3693
Hereford	83	288	974	2467	3520
Holstein	103	510	1256	3061	4350
Jersey	72	311	958	2523	3320
Limousin	154	911	1671	3456	4659
N'Dama	152	730	1336	2296	2946
Nelore	83	444	1545	4306	7199
Norwegian Red	89	437	1191	2808	3769
Piedmontese	151	898	1533	3310	4495
Red Angus	55	251	1151	2371	3303
Romagnola	87	408	1595	3321	4649
Santa Gertrudis	102	358	1587	4326	6562
Sheko	120	599	1759	4731	7145

3.9 Haplotype block structure

We estimated haplotype blocks based on r^2 using the following algorithm [61]: (i) Begin a block by selecting the pair of adjacent SNPs with the highest r^2 value (no less than α = 0.4); (ii) Repeatedly extend the block if the average r^2 value between an adjacent marker and current block members is at least β (= 0.3) and all the pairwise r^2 values within the block are at least γ (= 0.1).

For each breed, we estimated the haplotype blocks along with some statistics as follows: first, we counted the number of blocks, then we computed the percentage of region covered in blocks by dividing the total distance within blocks over the total effective distance comprised in the 101 targeted regions, then we counted the number of markers per block and the block size mean. Finally we estimated the 95% Confidence Interval (α

= 0.95) for the block mean size, assuming that block size follows a normal distribution, as:

$$(\overline{X} - t_{n-1,1-\frac{\alpha}{2}} \frac{s}{\sqrt{n}}, \overline{X} + t_{n-1,1-\frac{\alpha}{2}} \frac{s}{\sqrt{n}}),$$

where $\overline{X}$ denotes the sample average mean size, *s* denotes the sample standard deviation, *n* denotes the sample size, and $t_{n-1,1-\frac{\alpha}{2}}$ denotes the $(1-\frac{\alpha}{2})th$ percentile of a *t* distribution with *n-1* degrees of freedom [85].

Table 3.7 summarizes the haplotpye block structure in high-density regions across all breeds. In summary, the average maximum number of markers per block was 27.16. Across all breeds, 34.7% of the high-density regions were covered by haplotype blocks. We found that mean block size varied from 5.7 to 15.67 kb across breeds (with a mean block size of 10.3 kb over all breeds) and an average of 3.8 markers per block. These results are similar to those in a recent study of human haplotype blocks [86], which reported haplotype block sizes averaging 7.3, 13.2, and 16.3 kb in three human populations when analysing ten 500-kilobase regions with a density of one SNP per ~5 kb. The human data showed a marked decline in LD over the range of 1-100 kb, again similar to our observed decline in cattle LD from 0.6 to 0.1 over the range 1-100 kb.

Table 3.7 Haplotype block structure across high-density regions in all breeds.

Breed	No of blocks	Regions in blocks (%)	Markers per block (max)	Markers per block (average)	Min block size (kb)	Max block size (kb)	Block size mean (std) in kb	Block mean size 95 % Confidence Interval (min, max) in kb
ANG	282	41.61	38	4.21	0.25	68.08	11.28 (11.82)	9.89 , 12.66

BMA	299	34.86	19	3.53	0.33	57.21	8.62	(8.86)	7.61 , 9.63
BRM	233	19.61	16	2.98	0.25	30.38	6.72	(6.53)	5.88 , 7.56
BSW	257	35.97	41	4.04	0.09	74.71	11.39	(12.61)	9.84 , 12.94
CHL	302	40.7	20	4.01	0.07	67.26	10.48	(11.24)	9.21 , 11.75
GIR	191	13.88	11	2.78	0.03	20.50	5.38	(4.84)	4.69 , 6.07
GNS	289	39.71	16	4.08	0.51	54.28	10.79	(11.18)	9.50 , 12.08
HFD	280	54.3	41	5.01	0.65	70.64	14.13	(13.35)	12.56 , 15.70
HOL	307	47.46	47	4.36	0.38	63.81	11.45	(11.49)	10.16 , 12.74
JER	302	40.77	41	4.01	0.31	65.65	11.11	(11.09)	9.85 , 12.37
LMS	296	42.85	35	4.14	0.15	65.99	10.47	(11.52)	9.15 , 11.79
NDA	211	22.15	14	3.29	0.30	37.21	8.39	(8.05)	7.30 , 9,48
NEL	192	14.83	7	2.73	0.10	23.54	6.51	(5.36)	5.75 , 7.27
NRC	298	43.61	21	4.12	0.53	70.29	12.28	(11.82)	10.93 , 13.63
PMT	288	40.32	30	4.08	0.25	52.10	10.38	(9.71)	9.25 , 11.50
RGU	264	38.23	32	4.07	1.14	58.20	11.58	(10.22)	10.34 , 12.82
RMG	273	31.35	30	3.67	0.13	52.62	9.40	(9.01)	8.33 , 10.47
SGT	298	34.58	24	3.54	0.04	49.85	8.37	(8.49)	7.42 , 9.34
SHK	225	22.35	33	3.28	0.08	55.50	7.79	(8.93)	6.62 , 8.96

From this and the results in the previous section, if we assume that the elucidated average of r^2 of ~0.1 in 100 kb, and that the haplotype block average size of ~10 kb with one informative SNP each ~5 kb are homogeneously distributed across the bovine genome, then, for constructing an LD map for association studies we should tag at least a SNP in each 100 kb. Therefore, we can estimate that it would be necessary to successfully assay at least 28,700 SNPs for a LD map for association studies. In the same way, it would be necessary to assay at least 574,000 SNPs to characterize the haplotype block structure across the entire bovine genome (assuming a bovine genome size of 2.87 Gb).

3.9.1 Haplotype block density correlation

To determine if the haplotype block structure in high-density regions is conserved among breeds, we counted the number of haplotype blocks occurring in each of the high-density regions for each breed, producing a 101-element *haplotype block density vector* for each

breed. Following [60], we computed Pearson product moment correlation coefficient, *r*, between each pair of breeds using the formula:

$$r_{i,j} = \frac{\sum (x_{i,k} - \overline{x_i})(y_{j,k} - \overline{y_j})}{\sqrt{\sum (x_{i,k} - \overline{x_i})^2 \sum (y_{j,k} - \overline{y_j})^2}}$$

where i and j represent two breeds, k represents a high density region, $x_{i,k}$ and $y_{j,k}$ represents the number of haplotype blocks found in region k for breeds i and j respectively, and $\overline{x_i}$ and $\overline{y_j}$ represents the mean number of haplotype blocks found across all regions for breeds i and j respectively.

Figure 3.5 shows the block density correlation between breeds, and table 3.8 presents the block density correlation within the group and outside the group.

	ANG	BMA	BRM	BSW	CHL	GIR	GNS	HFD	HOL	JER	LMS	NDA	NEL	NRC	PMT	RGU	RMG	SGT	SHK
ANG	1.00																		
BMA	0.55	1.00																	
BRM	0.44	0.54	1.00																
BSW	0.58	0.61	0.53	1.00															
CHL	0.69	0.68	0.52	0.59	1.00														
GIR	0.29	0.51	0.64	0.38	0.62	1.00													
GNS	0.64	0.57	0.45	0.66	0.67	0.33	1.00												
HFD	0.66	0.51	0.35	0.49	0.54	0.27	0.57	1.00											
HOL	0.69	0.55	0.37	0.45	0.61	0.22	0.67	0.62	1.00										
JER	0.74	0.46	0.47	0.59	0.57	0.22	0.58	0.60	0.62	1.00									
LMS	0.71	0.70	0.55	0.70	0.73	0.47	0.65	0.65	0.59	0.59	1.00								
NDA	0.64	0.52	0.36	0.45	0.49	0.27	0.49	0.57	0.61	0.52	0.56	1.00							
NEL	0.22	0.32	0.68	0.26	0.49	0.67	0.33	0.07	0.22	0.29	0.31	0.14	1.00						
NRC	0.73	0.66	0.49	0.62	0.73	0.42	0.66	0.58	0.63	0.61	0.69	0.60	0.25	1.00					
PMT	0.68	0.64	0.52	0.70	0.77	0.50	0.69	0.62	0.62	0.56	0.77	0.49	0.37	0.67	1.00				
RGU	0.57	0.60	0.50	0.54	0.75	0.52	0.69	0.47	0.63	0.54	0.63	0.44	0.45	0.66	0.72	1.00			
RMG	0.48	0.65	0.57	0.71	0.65	0.48	0.58	0.49	0.46	0.51	0.71	0.46	0.37	0.62	0.67	0.54	1.00		
SGT	0.57	0.74	0.66	0.60	0.70	0.59	0.57	0.55	0.47	0.45	0.68	0.48	0.43	0.61	0.63	0.59	0.61	1.00	
SHK	0.44	0.55	0.68	0.54	0.58	0.64	0.42	0.43	0.40	0.55	0.57	0.44	0.57	0.51	0.59	0.52	0.60	0.59	1.00

Composite
Indicus
African
Dairy
Beef

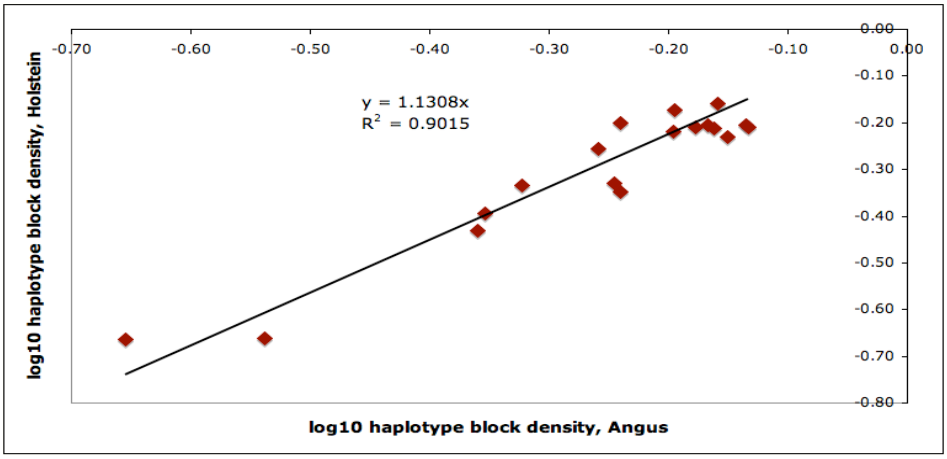
Figure 3.5 Block density correlation across high-density regions shows the level of conservation in haplotype block structure among breeds from the same group.

Figure 3.5 is color-coded to highlight the correlation between pairs of breeds in the same subgroup. The largest observed pair-wise correlations among breeds (0.77) occurred among Piedmontese, Charolais and Limousin (all three continental beef breeds). The smallest observed correlation was 0.07 between Hereford, a beef breed, and Nelore, an indicus breed.

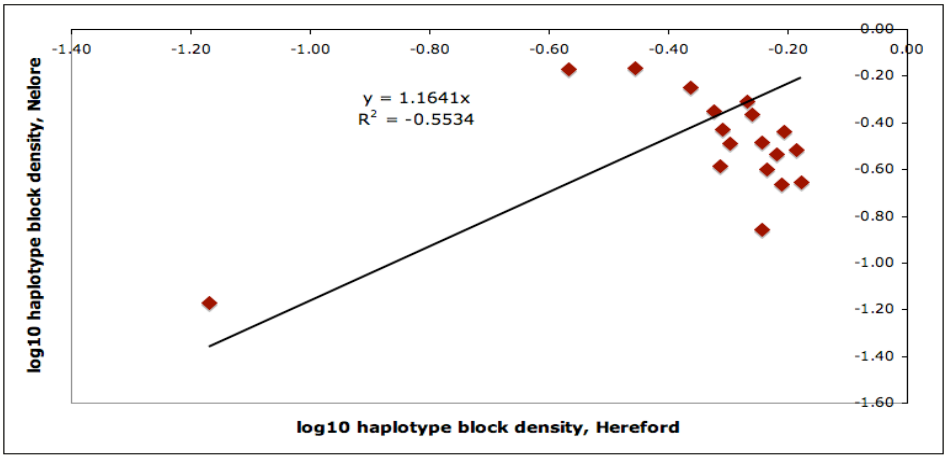
Table 3.8 Average haplotype block density correlations from all breeds within the group and outside the group.

Cattle group	Within group	Outside group
Beef	0.64	0.56
Dairy	0.61	0.54
African	0.44	0.51
Composite	0.74	0.57
Bos indicus	0.67	0.41

In general, indicus breeds showed small correlation with taurus breeds. For all subgroups, except African, the average within-group correlation was greater than the correlation with other subgroups. In the case of the African and composite breeds, the results may be biased by the sample size, having only two breeds from each subgroup. We observed a surprising degree of correlation between some subgroups, such as beef and dairy breeds. For example, Figure 3.6(a) presents the scatter plot of the density values ($\log_{10}$ values) of Holstein (a dairy breed) against Angus (a beef breed). Figure 3.6(b) shows a scatter plot for the lowest-correlation pair of breeds, Hereford and Nelore.



(a)



(b)

Figure 3.6 Comparison of Haplotype Block Densities between high-density regions of Holstein-a dairy breed- against Angus-a beef breed (both taurine) shows a high degree of correlation (a). Comparison of Nelore-an indicus breed against Hereford-a dairy breed (indicus against taurus) shows a low degree of correlation (b). The scatter plots show $\log_{10}$ of the amount of haplotype blocks for the same region in each breed pair.

3.9.2 Haplotype block boundary discordances

We examined the consistency in block boundaries across breeds and subgroups by looking at adjacent pairs of SNPs in the high-density regions. Following the strategy of [59]: for each breed, if the SNP pair was inside a block, we termed it NR (having no

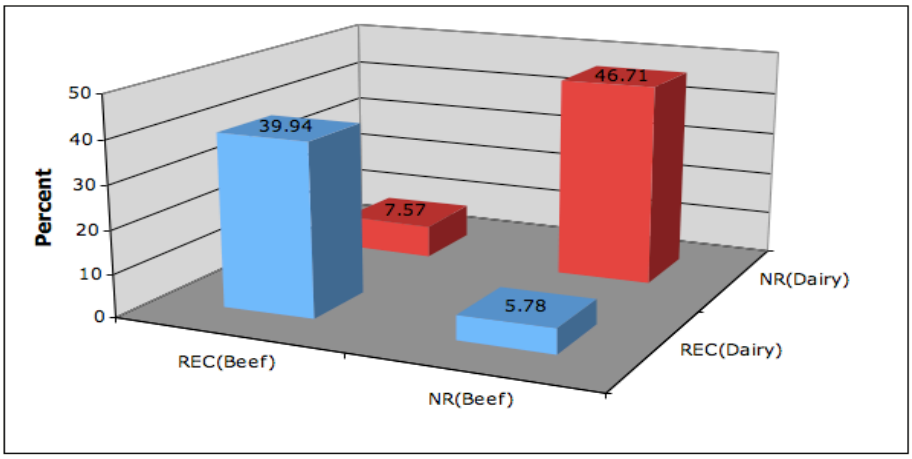
evidence of recombination), and if the SNP pair was outside a block, we termed it REC (having evidence of recombination). Then, for a given pair of breeds or subgroups, a SNP pair was called *concordant* if the assignment was the same in both breeds (or subgroups) and *discordant* if the assignment disagreed. Results from comparing several groups of breeds are presented in Table 3.9

Table 3.9 Proportions of block boundary discordances and concordances among cattle subgroups.

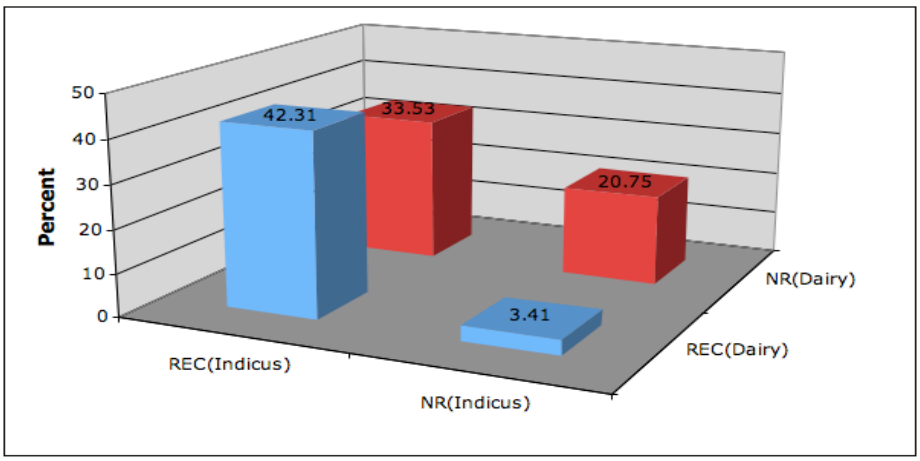
Comparison	Concordant NR (%)	Concordant REC (%)	Discordant NR – REC (%)	Discordant REC – NR (%)
Beef vs Dairy	46.71	39.94	5.78	7.57
Beef vs Indicus	20.92	44.28	31.56	3.24
Beef vs Composite	37.11	42.31	15.38	5.20
Beef vs African	27.57	42.83	24.91	4.68
Dairy vs Indicus	20.75	42.31	33.53	3.41
Dairy vs Composite	36.47	39.88	17.80	5.84
Dairy vs African	26.94	40.40	27.34	5.32
Indicus vs Composite	19.88	53.41	4.28	22.43
Indicus vd African	17.23	60.81	6.94	15.03
Composite vs African	24.45	49.88	17.86	7.80

Figure 3.7(a) shows that approximately 13% of adjacent markers have discordant assignment in beef and dairy breeds when analyzed as subgroups. This level of discordance indicates a high degree of similarity in the fine-scale haplotype block

structure between beef and dairy breeds, which suggests that a very detailed analysis of block discordance needs to be performed in order to differentiate between these two subgroups. On the other hand, Figure 3.7(b) shows that approximately 37% of marker pairs have discordant assignment when comparing the dairy subgroup against the indicus subgroup. This level of discordance indicates a fairly high degree of dissimilarity in haplotype structure between these two subgroups.



(a)



(b)

Figure 3.7 Concordance and discordance of block assignments for adjacent SNP pairs (within SNP pair distance <10 kb) in high-density regions. (a) dairy against beef breeds (both taurine), (b) dairy against indicus breeds (indicus against taurus).

3.10 Haplotype sharing

We examined the multi-marker haplotypes associated within the high-density regions to provide further insight into relationships among breeds. We analyzed the degree of sharing among the 19 breeds of phased haplotypes extending over multiple markers in the 101 high density regions. Each high-density region defined a locus for the purpose of the analysis. Haplotype segments were defined as the highest-probability haplotypes inferred by fastPHASE for each animal at each locus. The proportion of shared haplotypes between two populations P_1 and P_2 at locus k was defined as

$$S(P_1,P_2,k) = \frac{\sum_{i,j} S_a(i,j,k)}{2n_1n_2}$$

where i and j range over the individuals in populations P_1 and P_2, respectively, $S_a(i, j, k)$ is the number of shared haplotypes between individuals i and j at locus k, and n_1 and n_2 are the number of samples in P_1 and P_2. The raw proportions were normalized to take into account the proportion of shared haplotypes within each of the individual populations, as follows:

$$S'(P_1,P_2,k) = \frac{2 * S(P_1,P_2,k)}{S(P_1,P_1,k) + S(P_2,P_2,k)}$$

$S'(P_1, P_2, k)$ has value 1.0 if the proportional of shared haplotypes between populations P_1 and P_2 at locus k is equal to the average of the proportional of shared haplotypes within the two populations P_1 and P_2. If $S'(P_1, P_2, k) << 1.0$, then the proportion of shared haplotypes between the two populations is much less than the average within the two populations.

Table 3.10 shows the normalized proportion of shared haplotypes, averaged over all high-density regions, between various clusters of breeds. The most dramatic dissimilarity, as expected, is between all taurine and indicine populations.

Table 3.10 Normalized proportion of shared haplotypes.

	All regions	BTA 6	BTA 14	BTA 25
ANG / HOL	0.47	0.59	0.40	0.48
Beef / Dairy	0.73	0.84	0.68	0.70
Taurus / Indicus	0.17	0.19	0.14	0.21

Figure 3.8 shows a dendrogram based on using the proportion of shared haplotypes within the high-density regions as a distance measure for clustering breeds. The dendrogram shows a clear differentiation for breeds of African origin (N'Dama and Sheko), for *Bos taurus/Bos indicus* composite (Beefmaster and Santa Gertrudis), and for indicus breeds (Gir, Nelore, and Brahman).

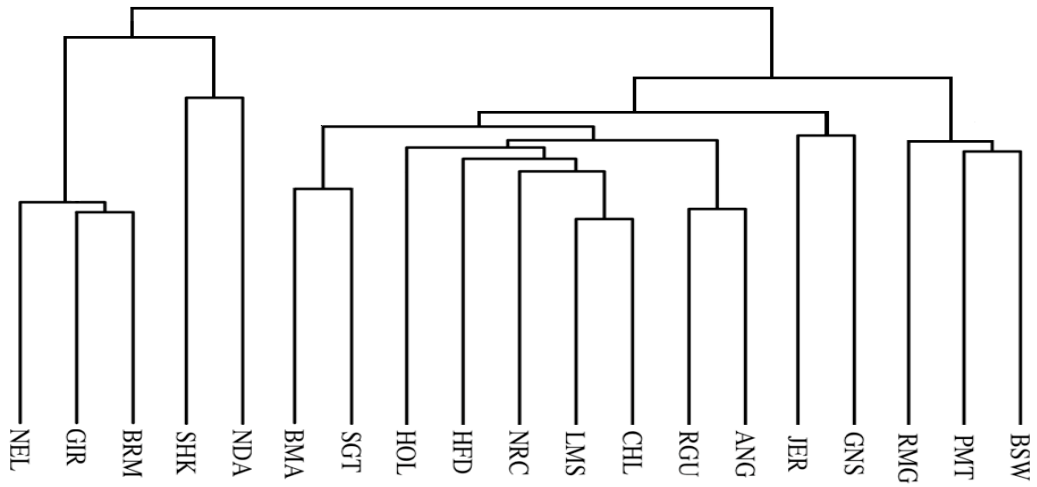


Figure 3.8 Dendrogram based on genetic distance calculated from haplotype sharing.

3.11 Breeds grouping

For each breed, we generated a *discordance vector* consisting of the percentage discordance found with all of the other breeds. Then, we used these vectors to perform a Principal Component Analysis (PCA) and look for differentiation between cattle subgroups. We used R software to perform this analysis. The central idea of PCA is to reduce the dimensionality of a data set which consists of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all the original variables [87].

Formally, PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second

greatest variance on the second coordinate, and so on. PCA is theoretically the optimum transform for a given data in least square terms. The procedure for obtaining PCAs can be summarized as follows:

Given a vector $\mathbf{X^T}$ of n dimensions, $X = [x_1, x_2, ..., x_n]^T$, whose mean vector **M** and covariance **C** are described by:

$$M = E(X) = [m_1, m_2, ..., m_n]^T$$

$$C = E[(X - M)(X - M)^T]$$

Calculate the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, and the eigenvectors $P_1, P_2, ..., P_n$; arrange them according to their magnitude.

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$$

Select d eigenvectors to represent the n variables, $d < n$. Then the $P_1, P_2, ..., P_d$ are called the principal components.

For the subgroups we investigated, PCA shows the best cluster separation in the subspace defined by the second principal component, PC2 (see Figure 3.9). For PC2, indicus, African, and composite breeds have negative loadings, while the beef and dairy breeds, all *Bos taurus* breeds of British and European origin, have positive loadings. Santa Gertrudis and Beefmaster, known to be *Bos indicus/Bos taurus* composites, appear as intermediate between the two main subgroups. This result is consistent with previous PCA analysis performed directly on genotypes for the complete set of markers [6]. Both PCA analyses define a strong axis of variation separating taurine from indicine subgroups and placing composites as intermediates. However, the analyses differ in the principal

component defining this relationship (PC1 for the genotype analysis, and PC2 for the block boundary discordances).

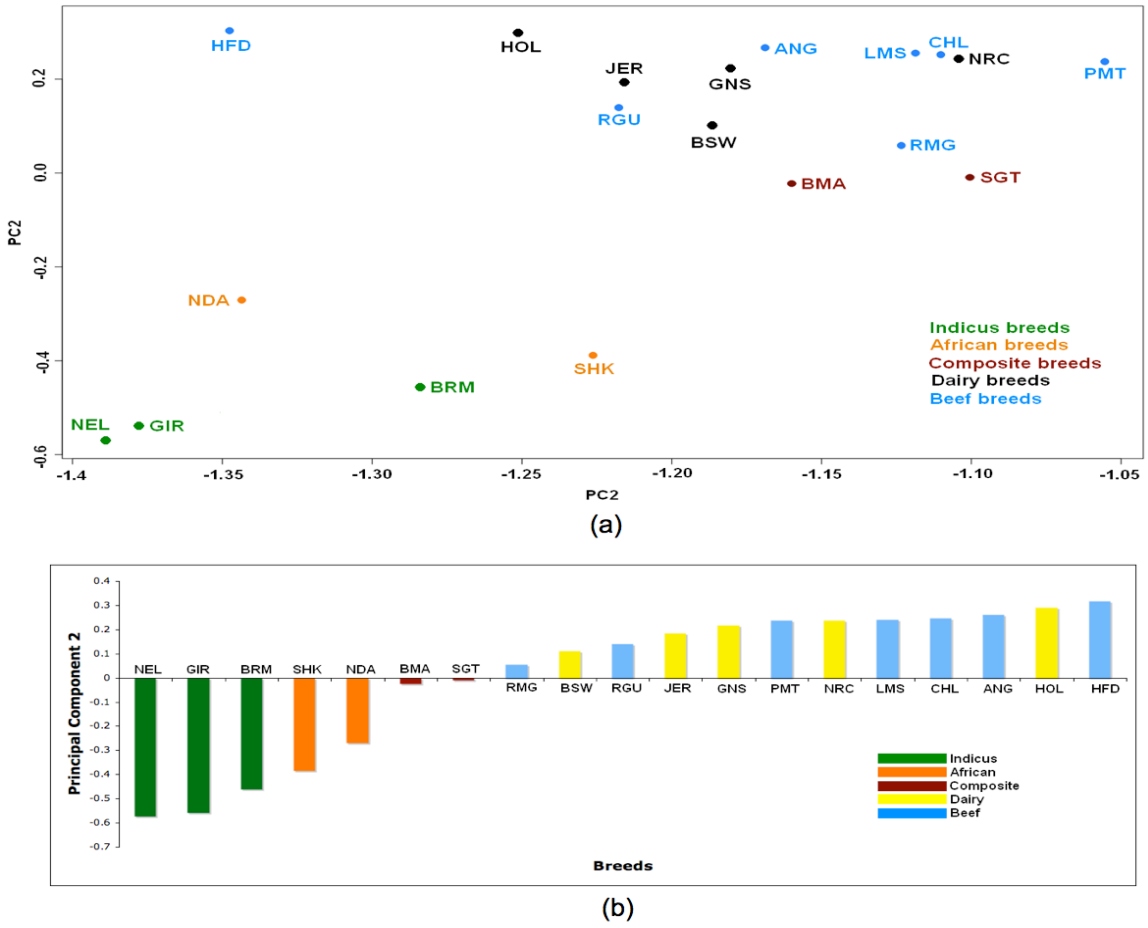


Figure 3.9 Principal Component Analysis on block boundary discordance vectors shows how different breed subgroups as indicus, African, and Composite cluster together, but there is no clear separation between dairy and beef breeds. (a) Plot of PCA1 vs PCA2. (b) Plot of PCA2.

In general, we consider that this analysis confirms that results obtained by analyzing the 1,981 SNPs in the selected high-density regions are consistent with the results obtained

by analyzing all of the initial 30k SNPs. We further observed that dairy and beef breeds cannot be clearly differentiated from this analysis. This result supports the hypothesis stated in the previous analysis [6] that historic geographic ancestry plays a stronger role in explaining genotypic variation (and haplotype block structure) in cattle than does their more recent selection into breeds with specific agriculture functions.

3.12 Summary

In this chapter we presented a high-resolution characterization of haplotype block structure in cattle. The analysis was performed on 101 targeted genomic regions spanning 7.6 Mb with an average density of one SNP each ~4 kb, sampled from 19 worldwide breeds. We studied LD and elucidated the block structure for each specific breed. Consistent with previous analyses in cattle, and in high agreement with observation in humans, we observed that LD declines rapidly, such that r^2 averages ~0.1 at 100 kb, and haplotype blocks exhibit an overall mean size of 10.3 kb (varying from 5.7 kb to 15.57 kb across all breeds) with an average of 3.8 markers per block. Estimation of effective population size in previous generations reflects the period of domestication ~12,000 years ago, as well as the current population bottleneck that breeds have experienced worldwide (last ~700 years) as a result of population isolation and selective breeding. In addition, an analysis of block density correlations, block boundary discordances, and haplotype sharing across all breeds and between subgroups were consistent in exhibiting a clear differentiation between indicus, African, and composite subgroups, but not between dairy and beef subgroups.

In summary, this work presents the first high-resolution analysis of haplotype block structure in worldwide cattle samples. First, novel results show that cattle and human share a high similarity in LD and haplotype block structure in the scale of 1-100 kb. Second, unexpected similarities in haplotype block structure between dairy and beef breeds make them non-differentiable. Finally, our results suggest that it would be necessary to successfully assay ~30,000 SNPs to construct an LD map for association studies, and ~580,000 SNPs to characterize the haplotype block structure across the entire bovine genome.

4 Haplotype inference in cattle pedigrees

4.1 Modeling cattle pedigree structure

Cattle may have complex pedigrees especially due to artificial insemination (AI) with few selected bulls that have their offspring extended over several generations [88]. An example pedigree structure is shown in Fig 4.1.

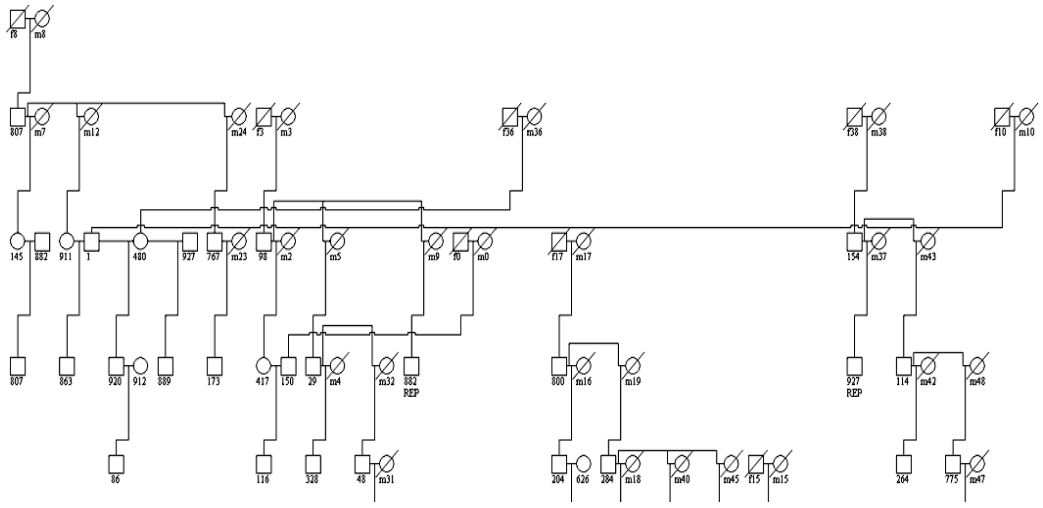


Fig 4.1 Small portion of pedigree from Holstein population (Data courtesy Dr. Curt Van Tassell).

From Figure 4.1 we can get a sense of how complex the genetic structure of cattle pedigrees can be. That kind of specific genetic structure is not present in human pedigrees or other species that have coevolved with humans.

The usual approach for modeling pedigrees is using descent graphs with nodes representing individuals and edges representing generational genetic linkage [44, 88-91]. Figure 4.2(a) shows a classical representation of a looped pedigree. A loop occurs when the graph has a cycle, defined as a path such that the start node and end node are the same. A pedigree can also be represented as a connected graph with nodes of two types (see fig 4.2(b)): the nodes or vertices that represent the individuals (the v_i's), and the nodes of matings (the m_i's). As shown in figure 4.2(b), there are edges (the e_i's) that connect the nodes. In general, a graph G is represented under the form $G = (N, E)$ where N is the set of nodes and E is the set of edges of G. Note that each arc e_i of the second graph representation links an individual and a mating node. In figure 3.2(b), we can define a loop as a sequence of no duplicated adjacent (i.e., linked by an edge) nodes $n_1 ... n_k$, except for $n_1 = n_k$ [88].

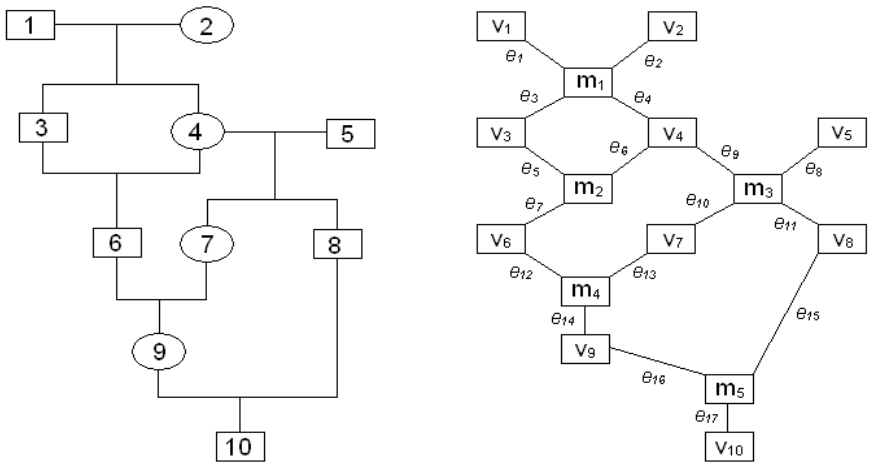


Figure 4.2 Classical representation of pedigrees. 2(a) representation using nodes for individuals and edges for genetic linkage. 2(b) representation using nodes for individuals and matings, and edges for genetic linkage.

The use of graphs to model pedigrees permits the development of computational methods for systematically estimate haplotypic information from genotypic data. But, it often can be quite difficult because (a) the computation involves the analysis of every possible underlying combination of multilocus genotypes, (b) the computational time of haplotype estimation grows exponentially with the number of genotypes, and (c) the presence of loops in the pedigree makes difficult to identify and eliminate superfluous genotypes, which are genotypes that are not compatible with all members in the pedigree.

One common strategy when computing some statistics, including haplotype estimation, in pedigrees, is to start by eliminating superfluous genotypes and avoid the processing of unnecessary data. Then, perform the rest of the analysis. In the next subsections we describe a computational method for haplotype estimation from genotype data. The method starts by eliminating superfluous genotypes using a genotype elimination

algorithm. Then, it makes use of graphs to reduce the complexity of the pedigree and improve the data analysis. Next, it implements an algorithm to search for valid haplotype configurations, marker by marker, for the complete pedigree. Finally, it makes use of a machine learning algorithm in order to find highly suitable haplotype configurations for the complete set of markers. Figure 4.3 shows a flow chart of the steps for estimating haploypes from pedigrees genotypes.

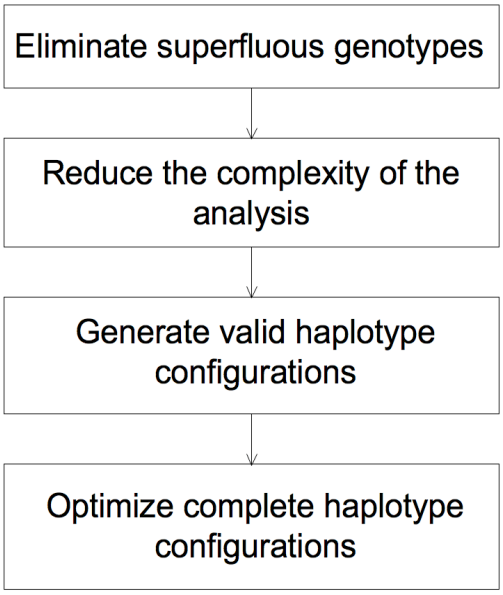


Figure 4.3 Flow chart of the steps for inferring haplotypes in pedigrees.

4.2 Genotype elimination

The first step in our method is to eliminate superfluous genotypes in order to avoid processing unnecessary data. Before we explain how we remove those genotypes let

make a definition: from [90]; given a connected pedigree with *n* individuals, a genotype vector $G = (G_1, G_2, ..., G_n)$, where G_i is the genotype of the *ith* individual, is *complete* if it is consistent with the observed phenotypes of the pedigree members. (Note that, for codominant markers, the observed phenotype is simply the observed genotype). A genotype-elimination algorithm typically aims to construct, for each member, a minimal genotype list that contains only genotypes that are the members of at least one compatible genotype vector. Genotypes that are not members of any compatible genotype vector are the superfluous genotypes, and need to be removed.

For eliminating superfluous genotypes we use the automatic genotype elimination algorithm proposed by Lange and Goradia (1987) [91]. This algorithm is an improvement of that proposed by Lange and Boehnke (1983) [92], and aims, on a locus by locus basis, to identify those genotypes that are not consistent with the observed phenotype information in the pedigree and that, because they contribute no information, therefore can be eliminated from the computations. The steps of the algorithm are as follows:

A. *For each pedigree member, list only those genotypes compatible with his or her phenotype*
B. *For each nuclear family:*
    1. *Consider each mother-father genotype pair.*
        a. *Determine which zygote genotypes can result.*
        b. *If each child in the nuclear family has one or more of the zygote genotypes among his current list of genotypes, then save the parental genotypes. Also save any child genotype matching one of the listed zygote genotypes.*
        c. *If any child has none of these zygote genotypes among his current list of genotypes-i.e., is incompatible with the current parental pair of genotypes-take no action to save any genotypes.*
    2. *For each individual in the nuclear family, exclude any genotype not saved during step 1 above.*
C. *Repeat part B until no more genotypes can be excluded.*

Figure 4.4 shows an example of genotype elimination applied to a trio. In the example, one marker genotype is given for each member. Genotypes for the father, mother and offspring are AB, AC, and BA respectively. In the first step we list, for each member, all compatible genotypes, which result to be AB BA, AC CA, and BA AB for father, mother and offspring respectively. In the second step we enumerate all father-mother genotype pairs, which are: AB AC, AB CA, BA AC, and BA CA. Next, we compute all possible zygotes that can result from each father-mother genotype pair. The resultant zygotes for the genotype pair AB AC are: AA AC BA BC. For the genotype pair AB CA are: AC AA BC BA. For the genotype pair BA AC are: BA BC AA AC. And, for the genotype pair BA CA are: BC BA AC AA. Then, for each zygote list we check if at least one of the zygotes appears to be in the list of the offspring genotypes. If it is the case, we save the parental genotypes along with the offspring genotype. In the example, we can observe that the only genotype from the offspring that appears in the zygote lists is BA; therefore, we save both genotypes from the father and the mother, and just BA genotype from the offspring. This procedure is repeated iteratively until no more genotypes can be excluded. The resultant list of non-superfluous genotypes for the complete trio is:

Father	AB BA
Mother	AC CA
Offspring	BA

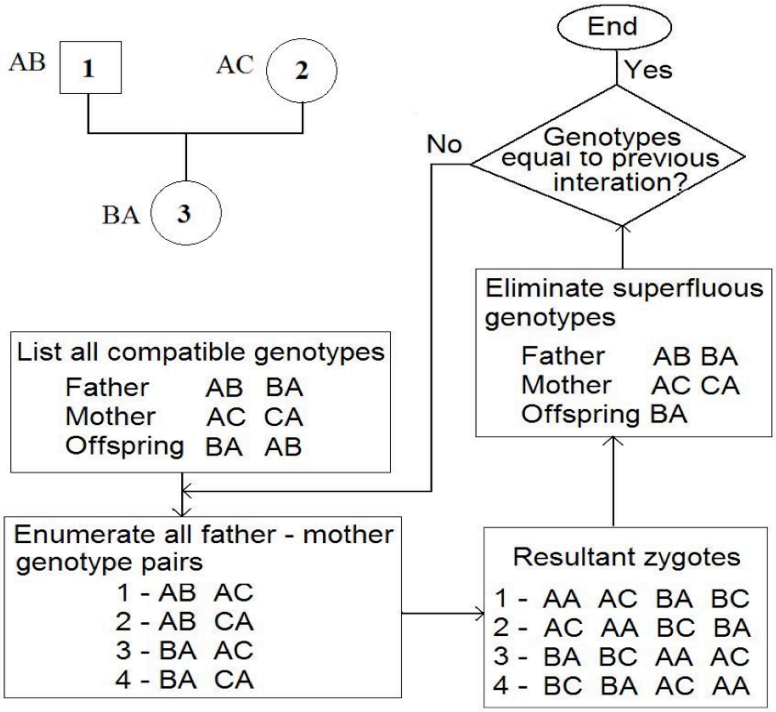


Figure 4.4 Example of Genotype elimination applied to a trio.

As demonstrated by Lange and Goradia in [91], this algorithm is guaranteed to eliminate all superfluous genotypes on any connected pedigree without loops. However, it is not optimal because it can fail to eliminate some superfluous genotypes from pedigrees with loops. As shown in subsection 4.4.2.2, this problem is implicitly solved by a haplotype compatibility constraint imposed when searching for valid haplotype configurations in the pedigree.

4.3 Feasible haplotype configuration by trio

After applying genotype elimination to the data from the pedigree, we obtain for each SNP of each individual, a list with no superfluous genotypes in the case of pedigrees

without loops, and possibly a few superfluous genotypes for pedigrees with loops. The next step is to systematically enumerate all feasible haplotypes for each individual, and then, under certain optimization criteria, select the two most suitable haplotypes as the inferred ones. To perform the haplotype enumeration we may model the pedigree using a descent graph with nodes representing individuals and edges representing genetic linkage. Figure 4.5 shows an example of a graph modeling an 11 member pedigree, in which individuals 1, 2, and 3 are founders and the rest are non-founders. Allele values are A, B, C, and D. The observed genotypes for one SNP are the values at the left of each pedigree member in the graph.

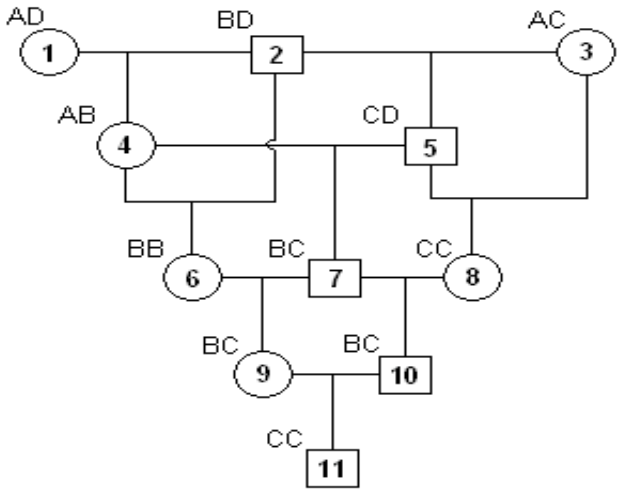


Figure 4.5 Example of a graph modeling a pedigree with 11 members, consisting of 3 founders and 8 non-founders.

As mentioned in section 4.1, traversing pedigrees from the top to the bottom searching for all feasible haplotype configurations has been proven to be a hard problem because in

additions to the analysis of every possible underlying combination of feasible genotype, the computational time of the analysis grows exponentially with the number of genotypes. In addition, the problem becomes even more complicated as the size of the pedigree and the type and number of loops increases. Henshall et. al., [89], proposed an strategy using inheritance constraints to perform the search and select valid configurations in a random form. Abecasis et. al., [32], proposed the use of binary trees for representing the pedigree and performed the search in the space of sparse trees. In this work, we propose a new strategy that consists of enumerating all trios in the pedigree, then elucidating all feasible haplotype configurations (FHC) – defined as an assignment of haplotypes to each individual in a trio, that satisfies Mendelian inheritance - for each trio, and finally performing the search in the space of all compatible FHCs between trios. It has two important advantages: First, elucidating FHCs for each trio helps eliminate some superfluous genotypes from the beginning of the analysis. Second, when performing the search, it is necessary to examine only those FHCs in which common members between trios share the same haplotype, and we may ignore the remaining configurations. These two advantages help to improve the speed of the search and make it easier to elucidate haplotypes for several individuals at the same time, since they are grouped in trios.

Following the example from figure 4.5, after applying genotype elimination we obtain for each individual the following list of feasible genotype:

Individual 1	A,D D,A
Individual 2	B,D D,B
Individual 3	A,C C,A
Individual 4	B,A

Individual 5	D,C
Individual 6	B,B
Individual 7	C,B
Individual 8	C,C
Individual 9	C,B
Individual 10	B,C
Individual 11	C,C

Enumerating all trios in the pedigree we have:

	Father	Mother	Offspring
Trio 1	2	1	4
Trio 2	2	3	5
Trio 3	2	4	6
Trio 4	5	4	7
Trio 5	5	3	8
Trio 6	7	6	9
Trio 7	7	8	10
Trio 8	10	9	11

Then, elucidating all FHCs for each trio we have:

Trio 1:	Father	Mother	Offpring
FHC 1	B,D	A,D	B,A
FHC 2	B,D	D,A	B,A
FHC 3	D,B	A,D	B,A
FHC 4	D,B	D,A	B,A
Trio 2:			
FHC 1	B,D	A,C	D,C
FHC 2	B,D	C,A	D,C
FHC 3	D,B	A,C	D,C
FHC 4	D,B	C,A	D,C
Trio 3:			
FHC 1	B,D	B,A	B,B
FHC 2	D,B	B,A	B,B
Trio 4:			
FHC 1	D,C	B,A	C,B
Trio 5:			
FHC 1	D,C	A,C	C,C
FHC 2	D,C	C,A	C,C
Trio 6:			
FHC 1	C,B	B,B	C,B
Trio 7:			
FHC 1	C,B	C,C	B,C

Trio 8:

FHC 1	B,C	C,B	C,C

Figure 4.6 presents a trio relational graph in which nodes represent trios and edges represent links between common individuals between trios.

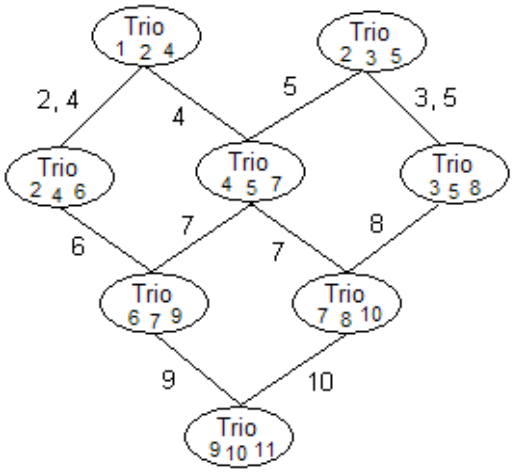


Figure 4.6 Trio relational graph. Nodes represent trios and edges represent links between trios. Numbers in links represent common individuals between trios.

This representation with trios defines a search space in which traversing the graph going from one node to another adjacent node is constrained by haplotype compatibility between common members of all trios in the pedigree. In the case of cattle pedigrees, this representation is advantageous because, due to artificial insemination, sires can appear in several trios and different generations and the graph does not take a complicated structure because each trio containing individuals in common with other trios just repeats the individual in its node. In the case of individual-based graphs, individuals appearing in several trios and generations generate complicated loops that make the graph very

complex and difficult to analyze. In the next subsection, we present a strategy for performing the search.

4.4 Complete pedigree valid haplotype configuration

Finding and enumerating all the complete pedigree valid haplotype configurations (CPVHC) in a fast and efficient way is crucial for time and space complexity of the algorithm. For doing this, we take advantage of the Trio Relational Graph (TRG) we have generated previously. In the following subsections we present the problem of enumerating all CPVHCs and propose a solution based on a backtracking strategy.

4.4.1 Problem representation

Given that we already have all FHCs for each trio in the pedigree, it is possible to select trio one, and for each of its FHCs, to perform a search dropping down trio by trio through the complete pedigree, looking for all paths containing compatible haplotype configuration between individuals linking one trio to another, and selecting one by one all FHCs for those trios with no liked individuals. To perform this search, we define – formally – compatibility between FHCs from different trios as follows: Given trios T_i and T_j, FHC_{ik} is compatible with FHC_{il} if and only if for each individual h in both trios, h has an identical haplotype in FHC_{ik} and FHC_{jl}. Traversing the trio relational graph with the constraint of FHCs compatibility produces all complete pedigree valid haplotype configurations (CPVHC). The search can be well-represented as a problem of searching for valid paths in a descent graph, as shown in figure 4.7.

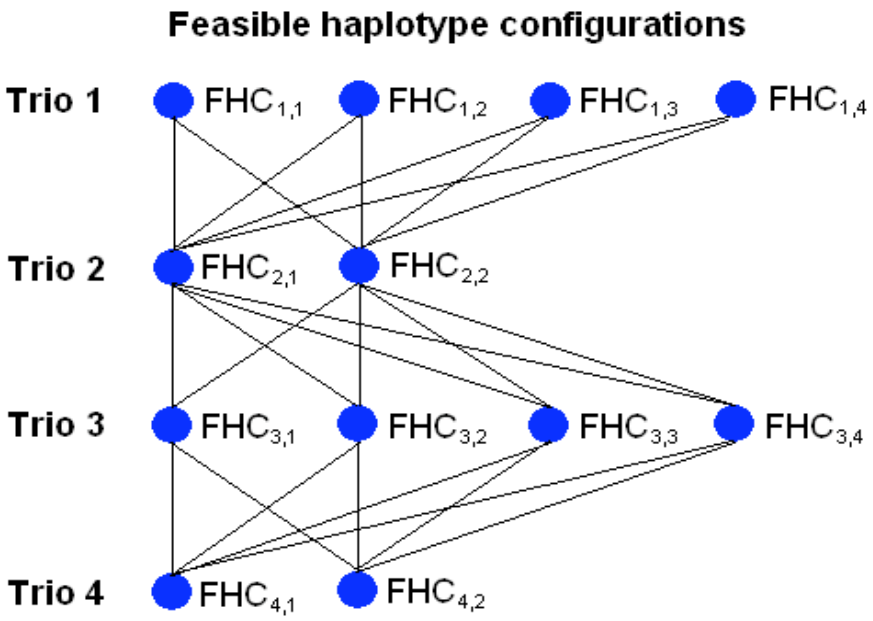


Figure 4.7 Searching for all complete pedigree valid haplotype configurations represented as a problem of searching for valid paths in a descent graph.

In the example from figure 4.7, the first trio (trio 1) contains four FHCs. For finding all valid CPVHCs, we select the first FHC ($H_{1,1}$), then we go to the next trio (trio 2) and ask which of its FHCs are compatible. For each compatible FHC in the second trio, we perform the same search on the third trio (trio 3). We repeat this search up to the last trio (trio 4, the bottom of the descent graph). In order to find all valid CPVHCs, we need to repeat the previous steps for each FHC belonging to trio 1.

We represent a CPVHC as a vector of integers $c = (FHC_{1,j_1}, FHC_{2,j_2}, FHC_{3,j_3}, ..., FHC_{n,j_n})$, where each FHC_{i,j_k} represents the $j_k th$ FHC of trio i, and n is the total number of trios in the pedigree, and all pairs of FHCs in the vector are mutually compatible. This

constraint of mutually compatibility between all FHCs in the vector is necessary in order to solve the problem of loops in the pedigree. Every time a new FHC is considered to be included to the CPVHC vector, it is checked if it has common members with all FHCs already in the vector, and it is included *if and only if* it shares a common haplotype with all common members in FHCs already in the CPVHC vector. Then the problem of enumerating all valid CPVHCs consists in finding the set of all *c*s that satisfy the constraint of haplotype compatibility between trio members. Figure 4.8 shows an example of a CPVHC. The red links show that from trio one, the first FHC is compatible with the second FHC in trio two and with the first FHCs in trios three and four. The set of all FHCs make a CPVHC.

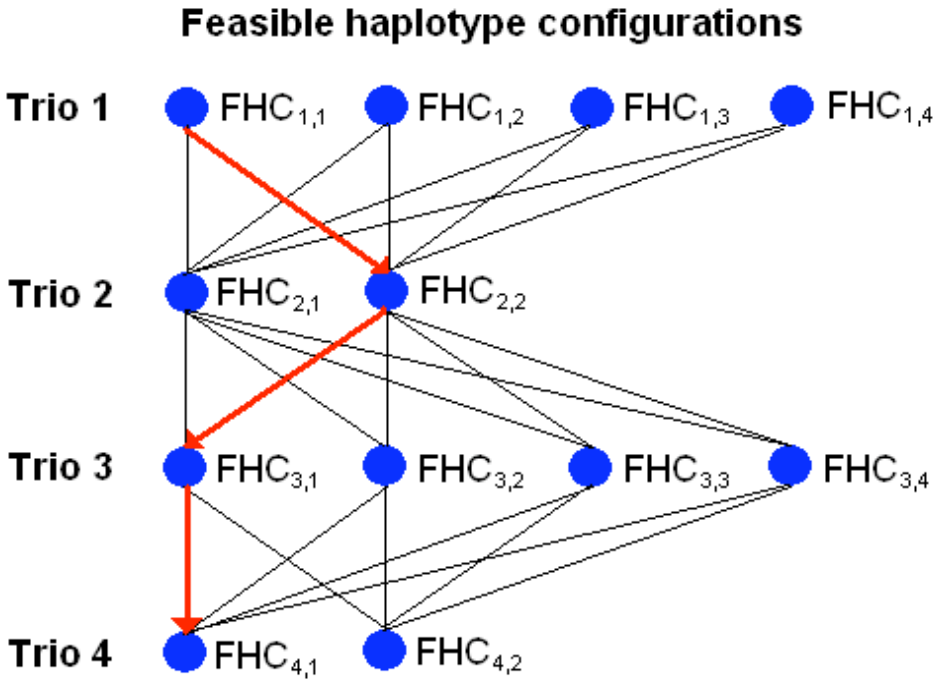


Figure 4.8 Example of a CPVHC

This problem is analogous to the general constraint satisfaction problem which in the general case has been proven to be a NP-complete problem [93]. In the next section, we present an implementation of a backtracking algorithm for solving the problem of enumerating all valid CPVHCs.

4.4.2 Backtracking algorithm for enumerating all CPVHCs.

Representing the problem of finding all CPVHCs as a problem of searching for valid paths in a descent graph permits the implementation of a backtracking algorithm for performing the search. Backtracking is a refinement of brute force approach, which systematically searches for solution to a problem among all variable options. It does so by assuming that the solutions are represented by vectors $(v_1,...,v_m)$ of values and by traversing, in a depth first manner, the domains of the vectors until the solutions are found [94, 95]. When invoked, the algorithm starts with an empty vector. At each stage it extends the partial vector with a new value. Upon reaching a partial vector $(v_1,...,v_i)$ which cannot represent a partial solution, the algorithm backtracks by removing the trailing value from the vector, and then proceeds by trying to extend the vector with alternative values.

4.4.2.1 Backtracking generic procedure.

One of the most common and efficient implementation of backtracking algorithms uses a recursive procedure and its pseudocode is as follows [96]:

```
procedure bt(c)
        if reject (P,c) then return
        if accept (P,c) then output (P,c)
        S ←---------- first (P,c)
        While s ≠ Λ do
                bt(s)
                s ←--------- next (P,s)
end procedure

function first (P,c)
        k ←---------- length (c)
        if k = n
                then return Λ
                else return (c[1], c[2], … , c[k], 1)
end function

function next (P,s)
        k ←-------- length (s)
        if k[s] = m
        then return Λ
                else return (s[1], s[2], … , s[k-1], 1+s[k])
end function
```

In this pseudocode, *P* is the data for the particular instance of the problem that is to be solved, *c* is a vector of values representing a partial or complete solution, *n* is the total number of variables in vector *c* (number of nodes in the descent graph), *m* is the domain of each variable in the solution vector *c*, and Λ is the null symbol.

As we observe in the pseudocode, the procedure takes the instance data *P* as a parameter and calls five different procedures in the code. *reject(P,c)* is the most problem specific procedure, and it should return *true* only if the partial candidate *c* is not worth completing. *accept(P,c)* should return *true* if *c* is a solution of *P*, and *false* otherwise. *first(P,c)* should generate the first extension of candidate *c*. *next(P,c)* should generate the next alternative extension of a candidate, after the extension *s*. And, *output(P,c)* uses the solution *c* of *P*, as appropriate to the application. In the next subsection we describe the

way we implemented Backtracking algorithm to solve the problem of enumerating all CPVHCs.

4.4.2.2 Backtracking to search for CPVHCs.

Formally, in order to find all valid CPVHCs we need to systematically enumerate all vectors $c = (FHC_{1,j_1}, FHC_{2,j_2}, FHC_{3,j_3}, ..., FHC_{n,j_n})$ such that they satisfy the constraint of haplotype compatibility between common individuals between all trios. For performing such systematic search we are following basically the generic Backtracking algorithm but we put special emphasis in two procedures, *reject(P,c)* and *next(P,s),* which are the most problem-specific parts of the algorithm.

We define *c* as a vector of values in which the first position corresponds to trio 1, the second position corresponds to trio 2, and so on. The values in each position corresponds to a pointer indicating a $FHC_{i,j}$, where, *i* and *j* are the trio and haplotype configuration number, respectively. From this definition, if in some step of the algorithm we have a vector $c = (2,3,7)$, it means that the 2nd FHC from trio 1, the 3rd FHC from trio 2, and the 7th FHC from trio 3 make a valid partial or complete pedigree haplotype configuration.

Before calling *bt(c)* procedure, we first extract from trio 1 all FHCs, then, for each FHC we call *bt(c)* giving as input the list *c* with an element corresponding to a FHC from trio 1. *bt(c)* procedure starts by generating a variable called *rt* to which it is assigned the actual length of *c*, which corresponds to the pointer to the last value in the list (the last analyzed trio), as shown next:

```
procedure bt(c)
        rt ←---------- length (c)
```

Next, we call *reject(P,c)* procedure, which function is to point to the next trio (pointer called *ct*) and check for each of its FHC if common individuals share the same haplotype, In this way find out if previous and current trios are compatible. The *reject(P,c)* pseudocode is as follows:

```
Procedure reject(P,c)
        ct = rt +1
        if exists ct data
                H_ct = (FHC_i,1, FHC_i,2, FHC_i,3, …)
                l = Λ
                for each FHC_i,j in H_ct do
                        if common members between rt and ct share same haplotype
                                y ←------- length (l)
                                l = (l[1], l[2], l[3], …, l[y], FHC_i,j)
                if l = Λ
                        then return Λ
                        else
                                y ←------- length (l)
                                m = l[y]
                                return l, m
        else return Λ
```

In order to explain how this procedure works, let us analyze an example. Suppose we have two trios, as shown in figure 4.9. Trio 1 contains two FHCs, while trio1 contains three FHCs.

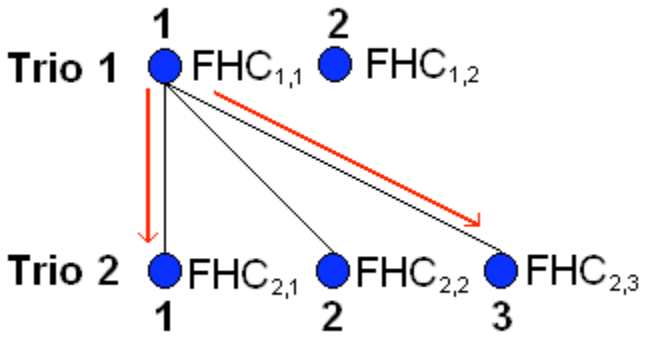


Figure 4.9 Example of two trios containing two compatible FHCs from $FHC_{1,1}$ to trio 2. Red lines indicate compatible FHCs.

Now, suppose we call *bt(c)* giving $c = (FHC_{1,1})$. *reject(P,c)* procedure will then be called after making *rt* = 1, and it will do the following: first, it will generate the pointer to the next trio making *ct* = *rt* +1 = 2. Then, the *if exists* clause will verify if trio 2 exists. In the case is does not exist, it will return Λ, meaning that there is no node beyond *rt*. Since in this case it exists, it will execute its internal code, generating the vector $H_{ct} = (FHC_{2,1}, FHC_{2,2}, FHC_{2,3}) = (1,2,3)$ containing the pointers to all FHCs in trio 2, and making $l = \Lambda$. Next, the *for each* clause will check, for each FHC in H_{ct}, if all common members between trios *rt* and *ct* share the same haplotype. In the case they do, it will add the FHC pointer to the list *l*. Suppose that $FHC_{2,1}$ and $FHC_{2,3}$ are compatible, then *l* = (1,3). In case there is no compatible FHC, then $l = \Lambda$. The last *if* clause checks if there were found no compatible FHCs between *rt* and *ct* trios. In the case $l \neq \Lambda$, it will make *m* to get the pointer value of the last compatible FHC from trio *ct*, and will return *l* and *m*. Finally, from this example, *reject(P,c)* would return $l = (1,3)$ and *m* = 3.

The *accept(P,c)* procedure is a straightforward procedure that asks whether *length(c)* is equal to the total number of trios in the pedigree. If it is the case, then returns *c* as output, indicating that a CPVHC has been reached. The pseudocode is as follows:

```
Procedure accept(P,c)
        k ←---------- length (c)
        if k = n
                then return output (c)
```

The procedure *first(P,c)* is another straightforward procedure that returns the same list *c* but including the fist pointer value from the list of compatible haplotype configurations *l*. The pseudocode is as follows:

```
Procedure first(P,c)
        k ←--------- length (c)
        if k = n
                then return Λ
                else return (c[1], c[2], c[3],…, c[k, l[1])
```

Finally, the *next(P,s)* procedure is called in order to traverse all compatible FHCs in the current node *ct*. These compatible nodes are stored in the list *l*. Then, the procedure needs to ask if the current FCH is the last in the list *l* (*m* contains the last value in *l*), and, if it is not the case, then add the next value in *l* to the input list *s*, and return the list. The pseudocode is as follows:

```
Procedure next(P,s)
        k ←---------- length (s)
        if s[k] = m
```

```
then return Λ
else
    for (x=1; x<=l; x++) do
                k = l[x]
                if k = s[k]
                        then return (s[1], s[2], s[3], …, s[k-1], l[x+1])
                        last
```

If we follow the previous example from figure 4.9, *reject(P,c)* returned $l = (1,3)$ and $m = 3$. When *next(P,s)* is called, *k* is made to point to the last value in the input list *s*. Then, the *if* clause asks if the last value in the list *s* is equal to *m*. if it is the case, returns Λ indicating that there is no more values in *l*. Otherwise, executes a *for* loop comparing each value in *l* with the last value in *s*. In the example, when *next(P,s)* is called by the first time, $s = l[1] = 1$, whose value was updated to *s* through *fist(P,c)* procedure. Then, the *for* loop compares $k = l[x]$ with $s[k] = l[1]$. In the first loop both values are equal to 1 since $x = 1$. Then, the *s* is updated replacing its last value by $l[x+1]$, which corresponds to $l[2] = 3$. It is returned and the *for* loop is terminated.

4.5 Inheritance matrix

From the backtracking algorithm, we obtain a list of vectors indicating a CPVHC each. The next step is to make a representation of these vectors in such a way that it captures the genetic parameters we want to measure in order to search for the most suitable haplotype configurations. In this subsection we describe a representation which captures the allelic inheritance (gene flow) through the pedigree generations.

4.5.1 Inheritance vectors

In general, if we consider a pedigree with f founders and n nonfounders, and, if each individual has either both parents or else none at all in the pedigree there will be $2n$ parent-offspring pairs. Each parent-offspring pair corresponds to a single meiotic event. For each chromosomal location e, we define an *inheritance vector*, Ve, representing gene flow in the pedigree through a sequence of $2n$ binary digits, such that the i^{th} digit, $V_e(i)$, is 0 if the grand-paternal allele is transmitted in the meiosis connecting offspring O_i and parent P_i, and 1 if the grand-maternal allele is transmitted. Define the index of the first digit in any such vector to be 1 so that i ranges between 1 and $2n$. Figure 4.10 shows an example of an inheritance vector.

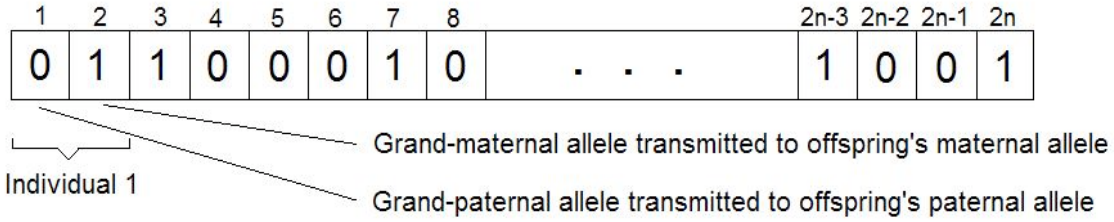


Figure 4.10 Example of an inheritance vector.

Analyzing pedigree data could be to enumerate all possible inheritance vectors (Ve) and calculate likelihoods and/or linkage statistics for the pedigree conditional in each Ve. Next we describe the form we enumerate and accommodate inheritance vectors in order to capture the number of recombination by marker pairs

4.5.2 Inheritance matrix

Once having all inheritance vectors, we represent the complete set of markers in the complete pedigree by aligning the inheritance vectors and thereby constructing a matrix, which we call the *inheritance matrix*. Figure 4.11 shows an example of an inheritance matrix for a pedigree with ten individuals and ten markers.

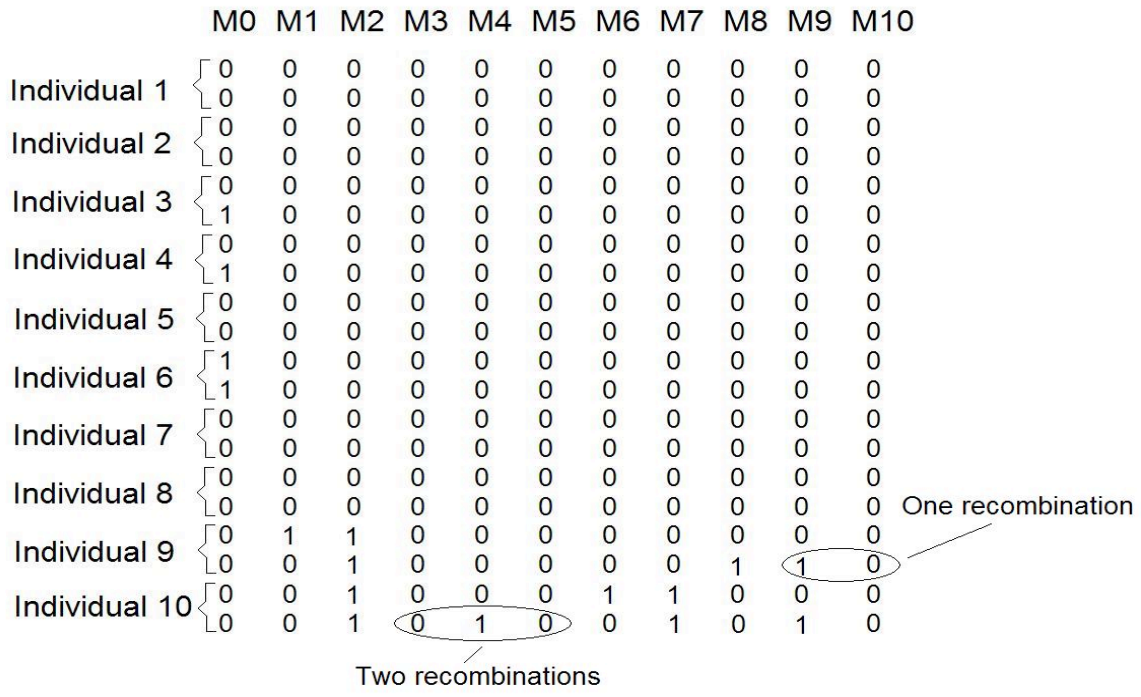


Figure 4.11 Example of an inheritance matrix from a pedigree with ten individuals and ten markers.

Inspecting the inheritance matrix, it is possible to find patterns of gene flow, and it permits us to compute the number of recombinations, marker by marker.

4.6 Number of recombinations

The *minimum recombination haplotype configuration* principle states that genetic recombinants are rare and thus haplotypes with fewer recombinants should be preferred in a haplotype reconstruction [31, 50, 97].

From the inheritance matrix it is possible to compute the number of recombination by just traversing each individual and counting the transitions from 0 to 1 or vice versa. The total number of transitions in the complete matrix is the total number of recombinations for that specific haplotype configuration of the pedigree. In the example from figure 4.11, the total number of recombinations is 22. The following subsections present an approach based on genetic algorithms to infer haplotypes in pedigrees searching for CPVHCs and minimizing the number of recombinations.

4.7 Genetic Algorithm for haplotyping

In previous subsections we described how to generate CPVHCs and how we implemented a backtracking strategy for enumerating all feasible CPVHCs. Then we showed how to construct an inheritance matrix and compute the number of recombinations. One strategy to infer haplotypes for a complete pedigree would be to enumerate all inheritance matrices and select that with the minimum number of recombinations. But, the problem with this strategy is that, even when we reduced the complexity of the search for CPVHCs by generating trio relational graphs, the search space increases exponentially with number of individuals in the pedigree, and for

moderated size of pedigrees it becomes computationally infeasible. Therefore, it is necessary to perform the search in a smarter way.

To solve this searching problem we make use of a genetic algorithm for finding sets of inheritance matrices and optimizing the number of recombinations. We perform this optimization by taking advantage of the GA evolutionary strategies for evolving solutions. Some advantages of using a GA for solving this search problem are that we can find sets of different optimal solutions, we can find optimal solutions without inspecting all the search space, and we are able to integrate and remove different genetic parameters when evaluating solutions. Figure 4.12 shows a flow chart of the genetic algorithm.

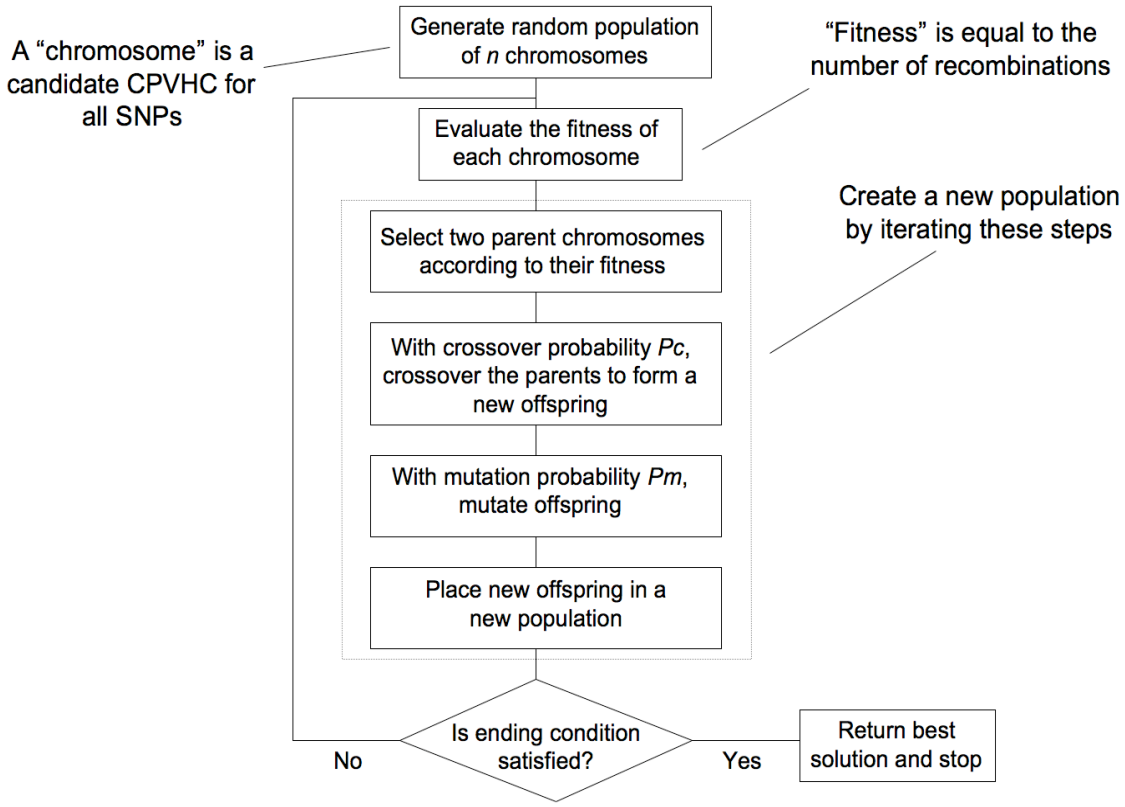


Figure 4.12 Flow chart of the genetic algorithm used for haplotype inference.

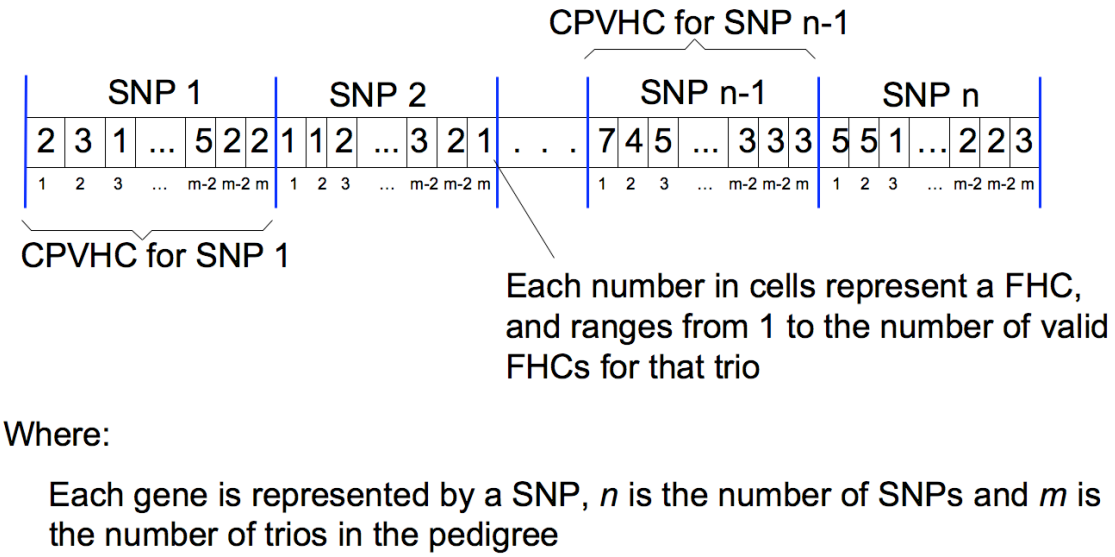
The GA starts by generating randomly an initial population of candidate solutions for the haplotype configuration of the complete pedigree. Then, it evaluates the fitness of each candidate by counting the number of recombinations in each individual. From the resulting fitness of each individual, pairs of parents are selected to generate the next population generation. Next, new offspring are generated crossing over selected parents. Some individuals are randomly mutated in order to keep diversity in the population. These steps are executed in an iterative form until a stop condition is reached. The result is a set of candidate solutions with optimized number of recombinations. Finally, we select the haplotype configuration with the minimum number of recombinations as the final solution. In the following subsections we describe some important aspects of the implementation of the GA.

4.7.1 Search space

The search space corresponds to the total number of feasible solutions for the haplotype configuration of the complete pedigree. From the inheritance vector structure we have that each cell has two possible outcomes, 0 or 1, and the number of cells is $2n$ where n is the number of individuals in the pedigree. Then, the total number of inheritance vectors is 2^{2n}. For a set of m SNPs, the total number of haplotype configurations (including consistent and non-consistent configurations) is $m*2^{2n}$.

4.7.2 Representation

One of the most important factors (if not the most important) in the success of a genetic algorithm is how we represent (encode) a candidate solution [98-103]. In our haplotyping problem we have a set of SNPs that conform our data. When constructing a solution, for each SNP we generate a CPVHC, which is a vector of integers. Then, the set of all CPVHCs, one for each SNP, conform a candidate solution. To represent this in the GA we juxtapose all CPVHC and construct a vector. Figure 4.13 shows a representation example.



Where:

Each gene is represented by a SNP, *n* is the number of SNPs and *m* is the number of trios in the pedigree

Figure 4.13 Representation of a candidate solution. CPVHCs from each SNP is juxtaposed to construct the representative chromosome in the GA.

From figure 4.13 we observe that the size of the representation vector is *n*m*, where *n* is the number of SNPs, and *m* is the number of trios in the pedigree. In this representation, each gene in the GA corresponds to a substring of size *m*, which is equivalent to a CPVHC in its corresponding SNP.

4.7.3 Fitness function

In order to know how well a GA chromosome (a candidate solution) solves the hapotyping problem, we need to define a fitness function to assign scores (fitness) to each solution according to the genetic parameter being optimized. In our implementation we aim to optimize the number of recombinations. Therefore, we defined our fitness function according to the number of recombinations computed from the candidate solution. Figure 4.14 shows the fitness function.

$$fitness = 1 - \frac{computed\ \ number\ of\ \ recombinations}{\max imum\ \ number\ of\ \ recombinations}$$

Where *maximum number of recombinations* is:
(number of SNPs -1)(2*number of individuals)*

Figure 4.14 Fitness function. The optimization consists in maximizing the fitness value.

For computing the fitness of a candidate solution, we generate it's inheritance matrix and compute the number of recombinations. This number if divided by the maximum number

of recombinations that occur in the pedigree, and subtracted to 1. The maximum number of recombinations is computed by multiplying the *number of SNPs* -1 by two times the number of individuals in the pedigree. The GA aims to maximize the fitness in its optimization process.

4.7.4 Initial population

The initial population is generated by randomly constructing a set of candidate solutions. For each candidate solution, a CPVHC is sampled for each SNP. These CPVHCs are sampled using the backtracking algorithm in a depth-first search mode to find the first CPVHC and stop. For randomizing the sampling we sort the order of FHCs in each trio, such that every time the backtracking algorithm searches for the depth-first search solution, it finds a different CPVHC. Figure 4.15 shows an example of a random CPVHC.

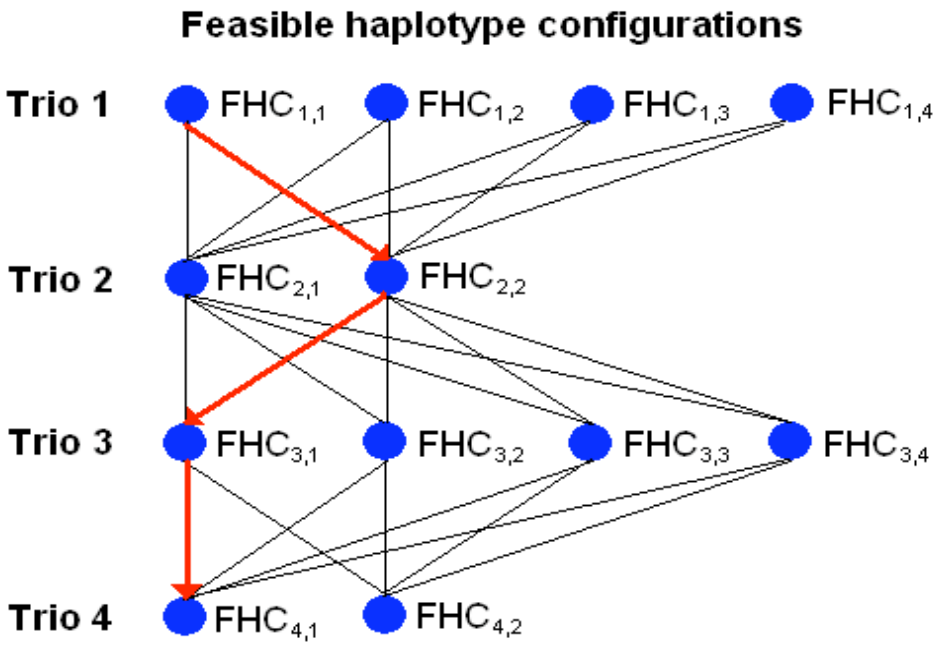


Figure 4.15 Randomized Backtracking for generating initial population.

4.7.5 Crossover

Crossover operator is used to generate offsprings by mating parents. It can be done by selecting randomly a crossover point (more than one point can be selected) and exchanging resulting substrings, as shown in figure 4.16. In our current implementation, we select two candidate solutions from the current population and generate a new candidate solution selecting randomly for each SNP (gene in the GA) one of both CPVHCs, the paternal o the maternal, and adding it to the new offspring (candidate solution). This operation always generates consistent haplotype configurations for the complete pedigree. Figure 2.16 shows an example of crossover.

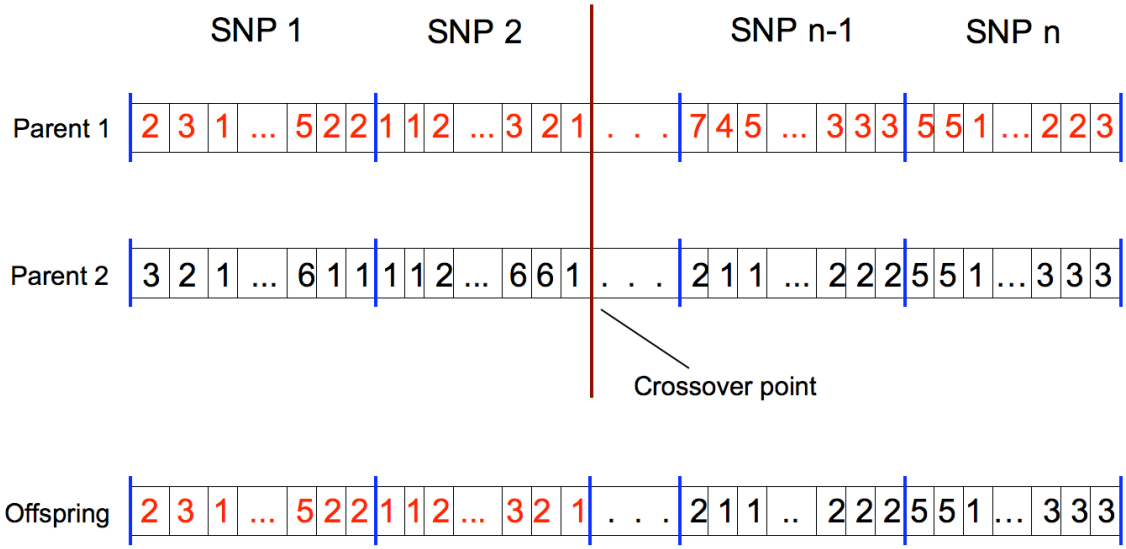
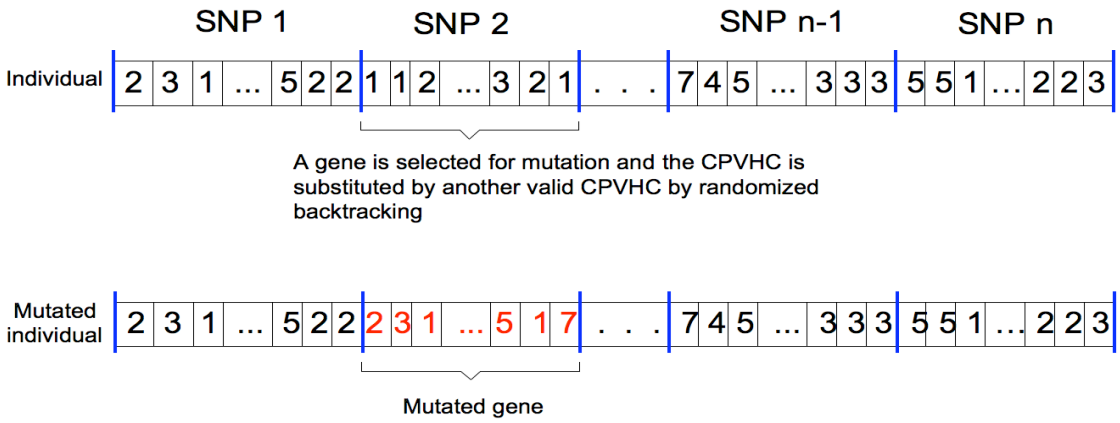


Figure 4.16 Crossover operator to generate new candidate solutions

In the current implementation we use a crossover rate of 1, which means that all individuals selected to contribute to the next generation are mated and recombined to generate new candidate solutions.

4.7.6 Mutation

Mutation operator is usually applied to new offspring and randomly flips some genes in a chromosome. Mutation can occur at each gene position in a string with some probability (mutation rate), usually very small. In our current implementation we use a mutation rate of 1/n where n is the number of SNPs. This guarantee that in average one gene is going to be mutated in each new candidate solution. Figure 4. 17 shows an example of mutation.



- Mutation always generate a valid CPVHC compared with Lee et all., 2008, and Tapadar et al., 1999

Figure 4.17 Mutation operator. A CPVHC is substituted by another with a probability 1/n.

A selected gene for mutation corresponds to a CPVHC. The mutation is generated by randomly sampling another CPVHC using the randomized backtracking and substituting its previous value. This mutation strategy guarantees that the resulting candidate solution is always consistent.

4.8 Analysis of performance of the developed GA-based method

In order to analyze the performance of the GA-based method we used the same Holstein pedigree used for evaluating the performance of Simwalk2 in chapter 2. This pedigree consists of 79 individuals, from which 40 are founders. We used the same set of 50 SNP haplotypes and genotypes generated from the simulator SimPed. Then, we applied the

GA approach to infer the haplotypes. We made the GA to run 10 different times during 100 generations with a population size of 100 individuals. From the results evaluated the runtime, number of recombinations, and the number switch errors and compared them with Simwalk2. Table 4.1 presents the results from both approaches.

Table 4.1 Comparison of performance inferring haplotypes for the Holstein pedigree between the GA-based developed method and Simwalk2

Method	Runtime	Number of recombinations	Switch errors
Genetic Algorithm	~ 6 hrs	500	459
Simwalk2	~ 10 hrs	600	601

From the results we can see that the GA outperformed the results from Simwalk2 in all measured parameter. The GA took ~6 hrs to run 100 generations and generate a set of haplotype configurations from which the best solution contained 500 recombinations and generated 459 switch errors compared to the SimPed solution, while Simwalk2 took ~100 hrs to generate one solution containing 600 recombinations and 601 switch errors compared to the real solution.

Figure 4.18 shows the decay of number of recombinations as the number of evaluated individuals by the GA increases.

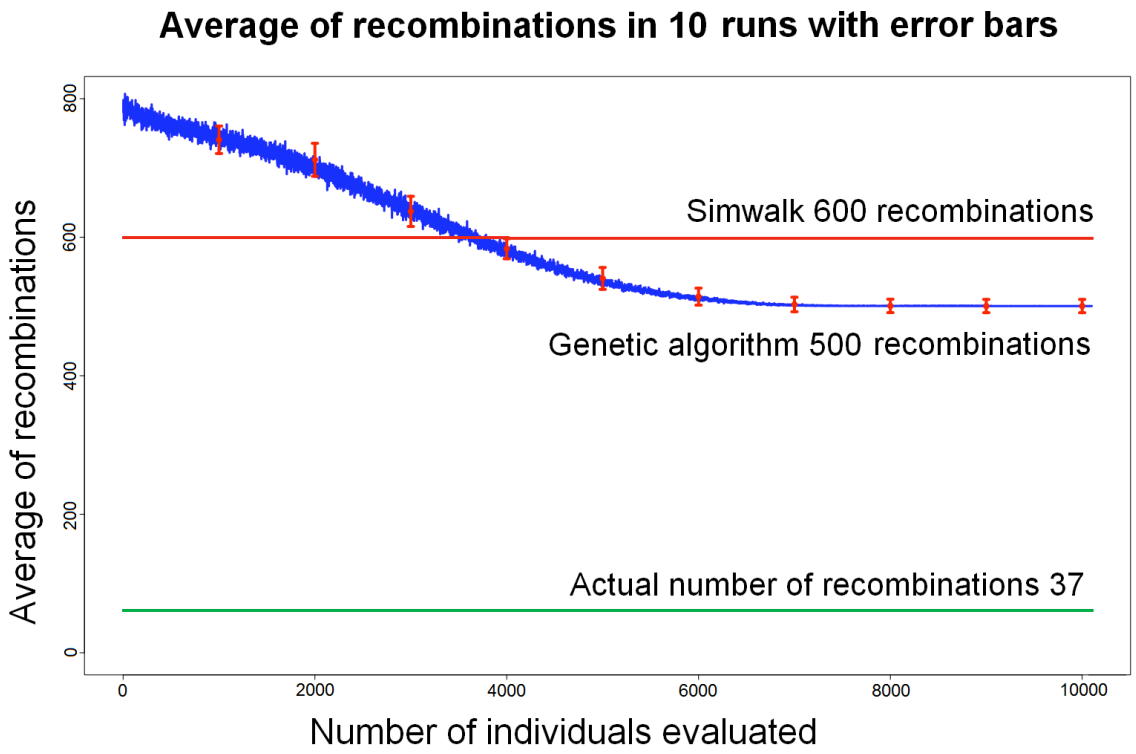


Figure 4.18 Decay of number of recombinations as the number of evaluated individuals by the GA increases.

The blue line shows the average number of recombinations from the 10 runs with error bars showing the standard deviation every 1000 individuals. We can observe that the average number of recombinations in the first individuals was ~800 but decreased to ~500 recombinations after 10000 individuals evaluated. The red line in the figure presents the number of recombinations generated by Simwalk, and the green line shows the actual number of recombinations. This plot shows how the GA generated fewer recombinations compared to Simwalk2, in the specific case of the Holstein pedigree.

Figure 4.19 shows the decay in number of switch errors as the number of evaluated individuals increases.

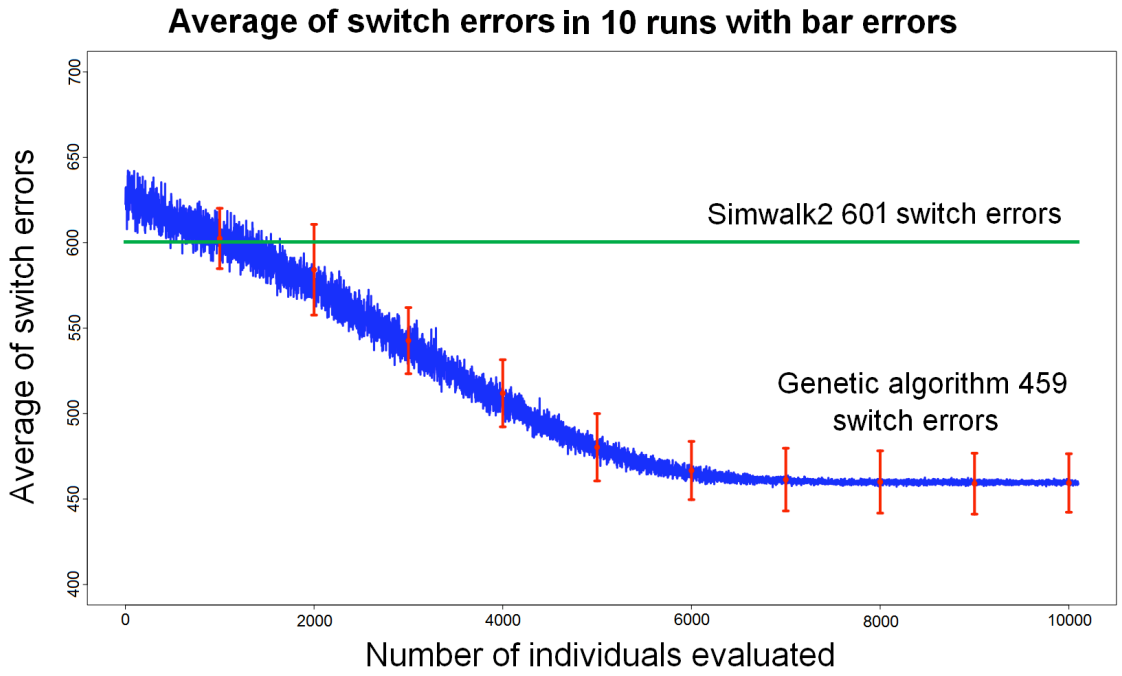


Figure 4.19 Decay of number of switch errors as the number of evaluated individuals increases.

From figure 4.19 we can observe how the number of switch errors in the first individuals takes values between 600 and 650 and how it decreases to 459 after 10000 individuals evaluated. The green line shows the number of switch errors generated by Simwalk2. This plot shows clearly how the GA generated fewer switching errors compared to Simwalk2 in the specific case of Holstein pedigree.

4.9 Summary

In this chapter we presented the design, implementation and partial evaluation of a new method based on genetic algorithms for haplotype inference in pedigrees. The approach includes the construction of trio relational graphs to search for feasible haplotype configurations, improving the complexity of the analysis and the speed for haplotyping groups of individuals, the implementation of a randomized backtracking strategy that permits us to solve the complexity of loops and sample complete pedigree valid haplotype configurations guaranteeing that we search within the space of feasible and consistent sets of haplotypes,

The use of an optimized genetic algorithm and the improvements in the process of sampling complete pedigree valid haplotype configurations permit this new method to perform better than Simwalk2 in runtime, number of recombinations, and number of switching errors, in a test pedigree consisting of 79 individuals, form which 40 are founders.

Additional tests of the new method are needed to test its performance on larger pedigrees and larger sets of markers. However, these preliminary results indicate that the method seems to be a highly suitable approach to infer haplotypes in complex pedigrees.

The method can be extended by incorporating other strategies (besides backtracking) for searching for CPVHCs. In the same way, improvements can be achieved by exploring different parameters in the GA, such increasing the population size, trying different selection methods, and incorporation more genetic parameters in the fitness function (i.e., counting for linkage disequilibrium). Finally, the runtime can be improved by

parallelizing the GA and exploring different subspaces at the same time. (for example [43]).

5 Conclusions and future work

5.1 Conclusions

In this dissertation project we proposed as general objective to identify the most appropriated method to infer haplotypes from the available genotype data from cattle, and to characterize the haplotype block structure based on patterns of linkage disequilibrium within different cattle breeds. After performing a literature review and analyzing the structure of cattle genotype data, we established as specific aims: (1) to evaluate alternative methods for haplotype inference in related and unrelated individuals from cattle data, (2) to apply an adequate method to cattle data, inferring haplotypes and performing a characterization of haplotype block structure based on linkage disequilibrium patterns, and (3) to develop an improved method for haplotype inference, based on a genetic algorithm approach.

In order to achieve aim 1, we performed a comparison in runtime and similarity of inferred haplotypes of three different algorithms applied to unrelated bovine samples, and we did a brief review of the capability of publicly available software for haplotype inference in a typical pedigree. In the case of unrelated individuals, PHASE, fastPHASE, and MERLIN were used to infer haplotypes from two different sets. One set consisted of 157 SNPs from chromosome 5 in 32 Holstein cows, and another set consisted of 2,465

SNPs from chromosome 6 in 27 unrelated cows from the Angus breed. In the case of related individuals, HAPLORE, MERLIN, and Simwalk2 software were applied to a Holstein breed pedigree consisting of 79 individuals, from which 40 were founders.

For unrelated individuals, MERLIN and fastPHASE are fast and comparable while PHASE is very slow. PHASE and fastPHASE produce the most similar haplotypes, with an average of ~80% of similarity. From the agreement graphs we can conclude that, regardless of the order in which resulting haplotypes are taken as paternal or maternal, the most frequently predicted allele is consistent. As a final conclusion for the analysis of unrelated individuals, in the case of cattle data which generally consist of large samples in individuals and SNPs, fastPHASE seems to be the most adequate method to infer haplotypes. Even when MERLIN is faster than fastPHASE, it was designed for analysis of pedigrees and computes gene flow trees, which are not present in unrelated individuals. In addition to being fast, fastPHASE produces very similar results to those from PHASE, which has been reported as the most accurate software so far. Of course, when the sample is small and the number of SNPs is not large, PHASE would be preferred over fastPHASE.

For related individuals, the only publicly available software capable of handling large and complex pedigrees typical in cattle datasets appears to be SIMWALK2. However, it is slow and its accuracy has not been extensively tested

For achieving aim 2, we inferred haplotypes, using fastPHASE, for all individuals in the HapMap data set, and performed a characterization of LD and haplotype block structure across 101 high-density targeted regions. We estimated the extent of LD along with the

estimation of ancestral population size for different generations. Then, haplotype block characterization allowed us to elucidate the breed-specific block structure and its variability compared with all other breeds. And haplotype block density correlation, haplotype block boundary comparison, and haplotype sharing between breeds and subgroups helped us to elucidate high-resolution similarities between breeds, and also permitted us to differentiate breeds by geographic separation versus those related by shared ancestry. Finally, breeds were clustered given computed genetic distances based on haplotype block analysis.

In conclusion, for achieving this goal we performed the first high-resolution analysis of haplotype block structure in worldwide cattle samples. Novel results show that cattle and human share a high similarity in LD and haplotype block structure in the scale of 1-100 kb. Unexpected similarities in haplotype block structure between dairy and beef breeds make them non-differentiable. And, finally, our results suggest that it would be necessary to successfully assay ~30,000 SNPs to construct an LD map for association studies, and ~580,000 SNPs to characterize the haplotype block structure across the entire bovine genome.

For achieving aim 3, we designed and implemented a new method based on genetic algorithms for haplotype inference in pedigrees. The approach in this work includes the construction of trio relational graphs to search for feasible haplotype configuration in trios, improving the complexity of the analysis and the speed for haplotyping groups of individuals, the implementation of a randomized backtracking strategy that permits to

solve the complexity of loops and sample complete pedigree valid haplotype configurations guaranteeing to always generate consistent sets of haplotypes.

The use of an optimized genetic algorithm and the improvements in the process of sampling complete pedigree valid haplotype configurations made this new method to perform better in runtime, number of recombinations, and number of switching errors, than Simwalk2, the only publicly available method capable of handling large and complex pedigrees, in an test pedigree consisting of 79 individuals, from which 40 are founders. Additional tests of the new method are needed to test it performance on larger pedigrees and larger sets of markers. However, these preliminary results indicate that the method seems to be a highly suitable approach to infer haplotypes in complex pedigrees.

In summary, the primary contributions of this dissertation project include:

1. The first high-resolution characterization of haplotype block structure in the cattle genome, consisting in analysis of LD and block structure, which showed great similarity with humans, analysis of effective population size which shows history of breeds development, analysis of density correlation, block boundary discordances, and haplotype sharing which shows clear differentiation between indicus, African, and composite subgroups, but not between dairy and beef subgroups.

2. A new approach based on genetic algorithms for haplotype inference in large and complex pedigrees. It includes a new representation that constrains search to space of feasible solutions, a new population initialization methods, crossover and

mutation operators, and a new fitness function that minimizes recombinations. The new approach outperforms SimWalk2 in accuracy and runtime, and is scalable for larger datasets

5.2 Future work

Different directions may be taken in order to continue expanding the results found in this dissertation project. First, as a scientific direction, the pipeline developed for LD and haplotype block characterization may be applied to the 50K Illumina chip data set genotyped by USDA (Dr. Curt Van Tassell). It would help to perform a high-resolution characterization of linkage disequilibrium and haplotype block structure in thousands individuals in a fast and optimized way, since the generated scripts were tested an improved when analyzing the HapMap data set. In the case of haplotype inference in cattle pedigrees, USDA Holstein populations consist of larger pedigrees than the test sample used for testing Simwalk2 and the GA-based approach. If the size and complexity of a pedigree increases from moderate to large Simwalk2 would become infeasible in runtime. However, the GA-based method provides the option of parallel processing. By initializing different populations and running a specific process for each population we would be able to explore different regions from the search space at the same time. It would improve the runtime and would make the GA-based method the only suitable method for handling large and complex pedigrees.

As a technical direction for improving results from this dissertation project we propose to test and validate the developed method for haplotype inference with more data, including

small, medium, and large pedigrees. This would help to verify in a more supported way the statement that the developed GA approach performs better than Simwalk2 for haplotype inference. Another improvement would be to translate the complete implementation of the GA-based method to the C language in order to make it faster, and to be able to make it publicly available.

REFERENCES

REFERENCES

1. Meuwissen TH, Hayes BJ, Goddard ME: **Prediction of total genetic value using genome-wide dense marker maps**. *Genetics* 2001, **157**(4):1819-1829.

2. Sonstegard TS, van Tassell CP: **Bovine genomics update: making a cow jump over the moon**. *Genetical research* 2004, **84**(1):3-9.

3. Gibbs R, Weinstock G, Kappes S, Loren S, Womack J: **White paper on Bovine Genomic Sequencing Initiative**. 2004.

4. Powell RL, Norman HD, Sanders AH: **Progeny testing and selection intensity for Holstein bulls in different countries**. *J Dairy Sci* 2003, **86**(10):3386-3393.

5. Schrooten C, Bovenhuis H, van Arendonk JA, Bijma P: **Genetic progress in multistage dairy cattle breeding schemes using genetic markers**. *J Dairy Sci* 2005, **88**(4):1569-1581.

6. The Bovine HapMap Consortium: **Genome-Wide Survey of SNP Variation Uncovers the Genetic Structure of Cattle Breeds**. *Science* 2009, **324**(5926):528-532.

7. Bonnen PE, Wang PJ, Kimmel M, Chakraborty R, Nelson DL: **Haplotype and linkage disequilibrium architecture for human cancer-associated genes**. *Genome Res* 2002, **12**(12):1846-1853.

8. Fallin D, Cohen A, Essioux L, Chumakov I, Blumenfeld M, Cohen D, Schork NJ: **Genetic analysis of case/control data using estimated haplotype frequencies: application to APOE locus variation and Alzheimer's disease**. *Genome Res* 2001, **11**(1):143-151.

9. Hugot JP, Chamaillard M, Zouali H, Lesage S, Cezard JP, Belaiche J, Almer S, Tysk C, O'Morain CA, Gassull M *et al*: **Association of NOD2 leucine-rich repeat variants with susceptibility to Crohn's disease**. *Nature* 2001, **411**(6837):599-603.

10. Mas A, Blanco E, Monux G, Urcelay E, Serrano FJ, de la Concha EG, Martinez A: **DRB1-TNF-alpha-TNF-beta haplotype is strongly associated with severe aortoiliac occlusive disease, a clinical form of atherosclerosis**. *Human immunology* 2005, **66**(10):1062-1067.

11. Reif A, Herterich S, Strobel A, Ehlis AC, Saur D, Jacob CP, Wienker T, Topner T, Fritzen S, Walter U *et al*: **A neuronal nitric oxide synthase (NOS-I) haplotype associated with schizophrenia modifies prefrontal cortex function**. *Mol Psychiatry* 2006, **11**(3):286-300.

12. VanRaden PM: **Efficient Methods to Compute Genomic Predictions**. *Journal of Dairy Science* 2008, **91**(11):11.

13. Cole JB, VanRaden PM, O'Connell JR, Van Tassell CP, Sonstegard TS, Schnabel RD, Taylor JF, Wiggans GR: **Distribution and Location of Genetic Effects for Dairy Traits**. In: *Interbull annual meeting.* Niagara Falls, NY; 2008.

14. Tassell CV: **An overview of the Bovine Hapmap project**. 2006.

15. Lee PH: **Computational Haplotype Analysis: An overview of computational methods in genetic variation study.** Ontario, Canada: Queen's University; 2006.

16. Matukumalli LK, Grefenstette JJ, Hyten DL, Choi IY, Cregan PB, Van Tassell CP: **SNP-PHAGE--High throughput SNP discovery pipeline**. *BMC Bioinformatics* 2006, **7**:468.

17. Wang X, Korstanje R, Higgins D, Paigen B: **Haplotype analysis in multiple crosses to identify a QTL gene**. *Genome Res* 2004, **14**(9):1767-1772.

18. Zeggini E, Weedon MN, Lindgren CM, Frayling TM, Elliott KS, Lango H, Timpson NJ, Perry JR, Rayner NW, Freathy RM *et al*: **Replication of genome-wide association signals in UK samples reveals risk loci for type 2 diabetes**. *Science* 2007, **316**(5829):1336-1341.

19. Wray NR, Goddard ME, Visscher PM: **Prediction of individual genetic risk to disease from genome-wide association studies**. *Genome Res* 2007, **17**(10):1520-1528.

20. Hunter DJ, Kraft P, Jacobs KB, Cox DG, Yeager M, Hankinson SE, Wacholder S, Wang Z, Welch R, Hutchinson A *et al*: **A genome-wide association study identifies alleles in FGFR2 associated with risk of sporadic postmenopausal breast cancer**. *Nat Genet* 2007, **39**(7):870-874.

21. McPherson R, Pertsemlidis A, Kavaslar N, Stewart A, Roberts R, Cox DR, Hinds DA, Pennacchio LA, Tybjaerg-Hansen A, Folsom AR *et al*: **A common allele on chromosome 9 associated with coronary heart disease**. *Science* 2007, **316**(5830):1488-1491.

22. Gusfield D, S.H. O: **Haplotype inference**. In: *CRC Handbook in Bioinformatics.* Edited by Press C; 2005: 1-25.

23. Clark AG: **Inference of haplotypes from PCR-amplified samples of diploid populations**. *Mol Biol Evol* 1990, **7**(2):111-122.

24. Gusfield D: **Inference of haplotypes from samples of diploid populations: complexity and algorithms**. *J Comput Biol* 2001, **8**(3):305-323.

25. Brown DG, Harrover IM: **A new formulation for haplotype inference by pure parsimony**. 2005.

26. Huang YT, Chao KM, Chen T: **An approximation algorithm for haplotype inference by maximum parsimony**. *J Comput Biol* 2005, **12**(10):1261-1274.

27. Lancia G, Pinotti MC, Rizzi R: **Haplotyping population by pure parsimony**. *Journal on comounting* 2004, **16**(4):348-359.

28. Bonzooni P, Vedova GD, Dondi R, Li J: **The haplotyping problem: an overview of computational models and solutions** *Computer Science and Technology* 2003, **18**(6):14.

29. Gusfield D: **Algorithms on strings, Trees and Sequences: Computer Science and Computational biology**: Cambridge University Press; 1997.

30. Gusfield D: **Haplotyping as Perfect Phylogeny: Conceptual Framework and Efficient Solutions** In: *RECOMB 2002.* Washington DC; 2002.

31. Bafna V, Gusfield D, Lancia G, Yooseph S: **Haplotyping as perfect phylogeny: a direct approach**. *J Comput Biol* 2003, **10**(3-4):323-340.

32. Abecasis GR, Cherny SS, Cookson WO, Cardon LR: **Merlin--rapid analysis of dense genetic maps using sparse gene flow trees**. *Nat Genet* 2002, **30**(1):97-101.

33. Halperin E, Eskin E: **Haplotype reconstruction from genotype data using Imperfect Phylogeny**. *Bioinformatics* 2004, **20**(12):1842-1849.

34. Excoffier L, Slatkin M: **Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population**. *Mol Biol Evol* 1995, **12**(5):921-927.

35. Scheet P, Stephens M: **A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase**. *Am J Hum Genet* 2006, **78**(4):629-644.

36. Qin ZS, Niu T, Liu JS: **Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms**. *Am J Hum Genet* 2002, **71**(5):1242-1247.

37. Stephens M, Donnelly P: **A comparison of bayesian methods for haplotype reconstruction from population genotype data**. *Am J Hum Genet* 2003, **73**(5):1162-1169.

38. Stephens M, Smith NJ, Donnelly P: **A new statistical method for haplotype reconstruction from population data**. *Am J Hum Genet* 2001, **68**(4):978-989.

39. Niu T, Qin ZS, Xu X, Liu JS: **Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms**. *Am J Hum Genet* 2002, **70**(1):157-169.

40. Mitchell M: **An introduction to genetic algorithms**: MIT Press; 1999.

41. Michalewicz Z, Fogel DB: **How to solve it: Modern Heuristics**: Springer Press; 2002.

42. Tapadar P, Ghosh S, Majumder PP: **Haplotyping in pedigrees via a genetic algorithm**. *Hum Hered* 2000, **50**(1):43-56.

43. Lee SH, Van der Werf JH, Kinghorn PB: **Using an evolutionary algorithm and parallel computing for haplotyping in a general complex pedigree with multiple marker loci**. *BMC Bioinformatics* 2008, **9**:10.

44. Sobel E, Lange K: **Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics**. *Am J Hum Genet* 1996, **58**(6):1323-1337.

45. Zhang BX, Wang RS, Wu LY, Chen L: **Models and Algorithms for Haplotpying Problem**. *Current Bioinformatics* 2006, **1**(1):105-114.

46. Zhang K, Sun F, Zhao H: **HAPLORE: a program for haplotype reconstruction in general pedigrees without recombination**. *Bioinformatics* 2005, **21**(1):90-103.

47. Nyholt DR: **GENEHUNTER: Your 'One-Stop Shop' for Statistical Genetic Analysis?** *Human Heredity* 2001, **53**:2-7.

48. Li J, Jiang T: **An Exact Solution for Finding Minimum Recombinant Haplotype Configurations on Pedigrees with Missing Data by Integer Linear Programming**. In: *RECOMB 2004.* San Diego CA: ACM; 2004.

49. Fishelson M, Dovgolevsky N, Geiger D: **Maximum likelihood haplotyping for general pedigrees**. *Hum Hered* 2005, **59**(1):41-60.

50. Qian D, Beckmann L: **Minimum-recombinant haplotyping in pedigrees**. *Am J Hum Genet* 2002, **70**(6):1434-1445.

51. Baruch E, Weller JI, Cohen-Zinder M, Ron M, Seroussi E: **Efficient inference of haplotypes from genotypes on a large animal pedigree**. *Genetics* 2006, **172**(3):1757-1765.

52. Khatkar MS, Zenger KR, Hobbs M, Hawken RJ, Cavanagh JA, Barris W, McClintock AE, McClintock S, Thomson PC, Tier B *et al*: **A primary assembly of a bovine haplotype block map based on a 15,036-single-nucleotide polymorphism panel genotyped in holstein-friesian cattle**. *Genetics* 2007, **176**(2):763-772.

53. Gautier M, Faraut T, Moazami-Goudarzi K, Navratil V, Foglio M, Grohs C, Boland A, Garnier JG, Boichard D, Lathrop GM *et al*: **Genetic and haplotypic structure in 14 European and African cattle breeds**. *Genetics* 2007, **177**(2):1059-1070.

54. Khatkar MS, Collins A, Cavanagh JA, Hawken RJ, Hobbs M, Zenger KR, Barris W, McClintock AE, Thomson PC, Nicholas FW *et al*: **A first-generation metric linkage disequilibrium map of bovine chromosome 6**. *Genetics* 2006, **174**(1):79-85.

55. Tenesa A, Knott SA, Ward D, Smith D, Williams JL, Visscher PM: **Estimation of linkage disequilibrium in a sample of the United Kingdom dairy cattle population using unphased genotypes**. *Journal of Animal Science* 2003, **81**:617-623.

56. Boettcher PJ, Pagnacco G, Stella A: **A Monte Carlo approach for estimation of haplotype probabilities in half-sib families**. *J Dairy Sci* 2004, **87**(12):4303-4310.

57. Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES: **High-resolution haplotype structure in the human genome**. *Nat Genet* 2001, **29**(2):229-232.

58. Goldstein DB, Cavalleri GL: **Genomics: understanding human diversity**. *Nature* 2005, **437**(7063):1241-1242.

59. Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M *et al*: **The structure of haplotype blocks in the human genome**. *Science* 2002, **296**(5576):2225-2229.

60. Guryev V, Smits BM, van de Belt J, Verheul M, Hubner N, Cuppen E: **Haplotype block structure is conserved across mammals**. *PLoS Genet* 2006, **2**(7):e121.

61. Gu S, Pakstis AJ, Kidd KK: **HAPLOT: a graphical comparison of haplotype blocks, tagSNP sets and SNP variation for multiple populations**. *Bioinformatics* 2005, **21**(20):3938-3939.

62. Gibson G, S.V. M: **A primer of genome science. Second Edition**: Sinaeur Press; 2004.

63. Stephens M, Scheet P: **Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation**. *Am J Hum Genet* 2005, **76**(3):449-462.

64. Lander ES, Green P: **Construction of multilocus genetic linkage maps in humans**. *Proc Natl Acad Sci U S A* 1987, **84**(8):2363-2367.

65. Zhang K, Zhao H: **A comparison of several methods for haplotype frequency estimation and haplotype reconstruction for tightly linked markers from general pedigrees**. *Genet Epidemiol* 2006, **30**(5):423-437.

66. Sobel E, Sengul H, Weeks DE: **Multipoint estimation of identity-by-descent probabilities at arbitrary positions among marker loci on general pedigrees**. *Hum Hered* 2001, **52**(3):121-131.

67. Sobel E, Papp JC, Lange K: **Detection and integration of genotyping errors in statistical genetics**. *Am J Hum Genet* 2002, **70**(2):496-508.

68. Kirkpatrick S, Gelatt CD, Jr., Vecchi MP: **Optimization by Simulated Annealing**. *Science* 1983, **220**(4598):671-680.

69. Lange K, Matthysse S: **Simulation of pedigree genotypes by random walks**. *Am J Hum Genet* 1989, **45**(6):959-970.

70. Leal SM, Yan K, Muller-Myhsok B: **SimPed: a simulation program to generate haplotype and genotype data for pedigree structures**. *Hum Hered* 2005, **60**(2):119-122.

71. Craig DW, Stephan DA: **Applications of whole-genome high-density SNP genotyping**. *Expert review of molecular diagnostics* 2005, **5**(2):159-170.

72. Hyten DL, Song Q, Choi IY, Yoon MS, Specht JE, Matukumalli LK, Nelson RL, Shoemaker RC, Young ND, Cregan PB: **High-throughput genotyping with the GoldenGate assay in the complex genome of soybean**. *TAG Theoretical and applied genetics* 2008.

73. Matukumalli LK, Grefenstette JJ, Hyten DL, Choi IY, Cregan PB, Van Tassell CP: **Application of machine learning in SNP discovery**. *BMC Bioinformatics* 2006, **7**:4.

74. Van Tassell CP, Smith TP, Matukumalli LK, Taylor JF, Schnabel RD, Lawley CT, Haudenschild CD, Moore SS, Warren WC, Sonstegard TS: **SNP discovery and allele frequency estimation by deep sequencing of reduced representation libraries**. *Nat Methods* 2008, **5**(3):247-252.

75. Jakobsson M, Scholz SW, Scheet P, Gibbs JR, VanLiere JM, Fung HC, Szpiech ZA, Degnan JH, Wang K, Guerreiro R *et al*: **Genotype, haplotype and copy-number variation in worldwide human populations**. *Nature* 2008, **451**(7181):998-1003.

76. McKay SD, Schnabel RD, Murdoch BM, Matukumalli LK, Aerts J, Coppieters W, Crews D, Dias Neto E, Gill CA, Gao C *et al*: **Whole genome linkage disequilibrium maps in cattle**. *BMC Genet* 2007, **8**:74.

77. Frazer KA, Ballinger DG, Cox DR, Hinds DA, Stuve LL, Gibbs RA, Belmont JW, Boudreau A, Hardenbol P, Leal SM *et al*: **A second generation human haplotype map of over 3.1 million SNPs**. *Nature* 2007, **449**(7164):851-861.

78. Sabeti PC, Varilly P, Fry B, Lohmueller J, Hostetter E, Cotsapas C, Xie X, Byrne EH, McCarroll SA, Gaudet R *et al*: **Genome-wide detection and characterization of positive selection in human populations**. *Nature* 2007, **449**(7164):913-918.

79. Nilsen H, Hayes B, Berg PR, Roseth A, Sundsaasen KK, Nilsen K, Lien S: **Construction of a dense SNP map for bovine chromosome 6 to assist the assembly of the bovine genome sequence**. *Anim Genet* 2008.

80. Tenesa A, Navarro P, Hayes BJ, Duffy DL, Clarke GM, Goddard ME, Visscher PM: **Recent human effective population size estimated from linkage disequilibrium**. *Genome Res* 2007, **17**(4):520-526.

81. Hayes BJ, Visscher PM, McPartlan HC, Goddard ME: **Novel multilocus measure of linkage disequilibrium to estimate past effective population size**. *Genome Res* 2003, **13**(4):635-643.

82. Hayes BJ, Lien S, Nilsen H, Olsen HG, Berg P, Maceachern S, Potter S, Meuwissen TH: **The origin of selection signatures on bovine chromosome 6**. *Anim Genet* 2008.

83. Sved JA: **Linkage disequilibrium and homozygosity of chromosome segments in finite populations**. *Theoretical population biology* 1971, **2**(2):125-141.

84. Kershaw I: **The Great Famine and Agrarian Crisis in England 1315-1322**. *Past and Present* 1973, **59**(1):3-50.

85. Rosner B: **Fundamentals of Biostatistics, Sixth edition**: Thomson Brooks/Cole; 2006.

86. International HapMap Consortium: **A haplotype map of the human genome**. *Nature* 2005, **437**(7063):1299-1320.

87. Jolliffe IT: **Principal Component Analysis**, 2nd edn: Springer; 2002.

88. Vitezica ZG, Mongeau M, Manfredi E, Elsen JM: **Selecting loop breakers in general pedigrees**. *Hum Hered* 2004, **57**(1):1-9.

89. Henshall JM, Tier B, Kerr RJ: **Estimating genotypes with independently sampled descent graphs**. *Genetical research* 2001, **78**(3):281-288.

90. O'Connell JR, Weeks DE: **An optimal algorithm for automatic genotype elimination**. *Am J Hum Genet* 1999, **65**(6):1733-1740.

91. Lange K, Goradia TM: **An algorithm for automatic genotype elimination**. *Am J Hum Genet* 1987, **40**(3):250-256.

92. Lange K, Boehnke M: **Extensions to pedigree analysis. V. Optimal calculation of Mendelian likelihoods**. *Hum Hered* 1983, **33**(5):291-301.

93. Edward T: **Fuondations of constraint satisfaction** *Academic press* 1993.

94. Bitner JR, Reingold ME: **Backtracking Programming Techniques**. *Communications of ACM* 1975, **18**(11):651-656.

95. **Backtracking** [http://en.wikipedia.org/wiki/Backtracking]

96. **Backtracking** [http://en.wikipedia.org/wiki/Backtracking]

97. O'Connell JR: **Zero-recombinant haplotyping: applications to fine mapping using SNPs**. *Genet Epidemiol* 2000, **19 Suppl 1**:S64-70.

98. Grefenstette JJ: **Incorporating problem specific knowledge into genetic algorithms**. In: *Genetic Algorithms and Simulated Annealing* Edited by Davis LD. London: Pitman; 1987.

99. Grefenstette JJ: **Genetic Algorithms and their Applications**. *Encyclopedia of Computer Science and Technology* 1990, **21**(6).

100. Grefenstette JJ: **Genetic Algorithms for Machine Learning**. *Kluwer Academic Publisher* 1994.

101. Grefenstette JJ: **Optimization of Control Parameters for Genetic Algorithms**. *IEEE Transactions on Systems, Man, and Cybernetics* 1986, **16**(1):7.

102. Burke DS, De Jong KA, Grefenstette JJ, Wu AS, Ramsey CL: **Putting More Genetics Into Genetic Algorithms** *Evolutionary Computation* 1998, **6**(4):23.

103. Mathe E, Grefenstette JJ: **Polyoptimizing genetic algorithms for feature subset selection**. In: *Interface 2004: Classification and Clustering 36th Symposium on the Interface.* Baltimore, MD; 2004.

CURRICULUM VITAE

Rafael Villa Angulo was born on May 03, 1970, in Guamuchil, Sinaloa, Mexico. He received his bachelors in Engineering from the University of Baja California, Mexicali, Mexico (1993). He was awarded Master in Computer Science from the Center of Scientific Research and Higher Education of Ensenada, Ensenada, Mexico (2001). He has been a research scientist in the Engineering Institute of the University of Baja California since 2002. He is currently a doctoral candidate at George Mason University.

Selected publications:

High-Resolution Haplotype Block Structure in the Cattle Genome. Villa-Angulo R., Matukumalli LK., Gill CA., Choi J., Van Tassell CP., Grefenstette JJ. BMC Genetics 2009, 10:19.

Genome-Wide Survey of SNP Variation Uncovers the Genetic Structure of Cattle Breeds. The HapMap Consortium. *Science* 2009, 324(5926):528-532.

A wearable Neural Interface for Real Time Translation of Spanish Deaf Sign Language to Voice and Writing. Villa-Angulo R., Hidalgo SH., 2005. Journal of Applied Research and Technology. **3**, 169-186.