Genotype imputation based on discriminant and cluster analysis

medhat ahmed abbas mahmoud

Norwegian University of Life Sciences
Department of Animal and Aquaculture Sciences
Master Thesis 30 credits 2012

Genotype imputation based on discriminant and cluster analysis

M.Sc. Student
Medhat Ahmed Abbas Mahmoud

Main supervisor:
Prof. Theo Meuwissen

*Department of Animal and Aquaculture Sciences
Norwegian University of Life Sciences
Ås, Norway*

Co-supervisor:
Prof. Thore Egeland

*Department for Chemistry, Biotechnology and Food Science
Norwegian University of Life Sciences
Ås, Norway*

Declaration

I certify that the work presented in this thesis is, to the best of my knowledge and belief, original, except as acknowledged in the text, and that the material has not been submitted, either in whole or in part, for a degree at this or any other university.

And to declare that I have read and understood the University's rules, requirements, procedures and policy relating to my higher degree research award and to my thesis. I certify that I have complied with the rules, requirements, procedures and policy of the Norwegian University of Life Sciences.

Print Name:__________________________________.

Signature:____________________________________. Date: / / .

PREFACE

This report is my master thesis for the conclusion of Master program in Bioinformatics and Applied Statistics at the Department for Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, Norway

This thesis is written to provide guidance for using Discrimination and Clustering analysis in SNP imputation. Imputation is a fairly new application for LDA and Clustering that has many facets that remain mysterious to the average person. People who have never used the LDA and Clustering may have major misconceptions about its content, use, and impact on Bioinformatics. People who have used the LDA and Clustering may only be familiar with a small portion of the information available to them. This thesis will seek to help familiarize people with some application of LDA and Clustering Analysis.

Differing from most of the other studies on Genotype imputation, this analysis was performed using a new application of linear discrimination and clustering analysis rather than using the traditional Regression or Maximum likelihood approaches. The feasibility of the methods in predicting the missing SNP is discussed.

Print Name:________________________________.

Signature:__________________________________.   Date: /   /   .

Acknowledgements

Some people experience the research and writing of a M.Sc. as an isolating experience. My experience has been the reverse. During the course of this study I have been surrounded by a network of people who provided support, advice, expertise and friendship. So that, I would like to acknowledge to my main supervisor supervisors: Professor Theo Meuwissen, I really appreciate your willingness to accept me as your thesis student, and for the guidance and comments that lead to the successful completion of this work.

And a warm thanks to my co-supervisor: Professor Thore Egeland. He gave me a lot of trust and flexibility on the project. Without him, I couldn’t have dealt with such a challenging project. He gave me not only a lot of detailed instructions to use “R” on my project but also many useful practical tips on scientific communication and how to implement a research project. All our meeting were fruitful because they are a good discussion partner. Today I finished my report and I will continue to challenge myself in the future with what I learned. This is not the end but just the start.

ABSTRACT

The recent development of high-throughput systems for genotyping SNP in Eukaryote has led to an extraordinary amount of research activity, particularly in areas such as whole-genome selection of livestock and genome-wide association studies for detection of quantitative trait loci (Van Tassell et al., 2008). Recent technological advances allow us to rapidly genotype more than 10 million SNPs in an individual, accounting for 10% of the estimated number of common SNPs (more than 1% minor allele frequency) across the population. As a result of missing SNPs, true associations might be missed if the causal SNP is not genotyped or if the causal variant is an unknown variant. SNP imputation is important in reducing the cost of re-sequencing and when genotyping all considered animals may be too costly and sometimes not feasible because DNA may not be available for all animals. Computational algorithms and statistical methods have been developed to account for some of the unobserved variants. The main idea behind these methods is based on the observation that SNPs in close proximity to one another in the genome tend to be correlated, or in non-random association (linkage disequilibrium). Several powerful methods to impute missing SNP genotypes already exist that, apart from the genotypic information at the locus of interest, "using only pedigree data" (Gengler, 2007, 2008), "only surrounding markers" (FastPHASE; Scheet and Stephens, 2006), or both (Li and Jiang, 2003; Kong et al., 2008; Meuwissen and Goddard, 2010; Mulder et al., 2010b). The mixed model (BLUP) method presented by Gengler et al. (2007) uses BLUP to find the missing gene content conditional on genotypic information of relatives. "Several articles have described comparisons of imputation methods with respect to computational efficiency and the accuracy of results" (Pei YF, 2008; Yu Z, 2007; Nothnagel M, 2009). Overall, MACH, BEAGLE, and IMPUTE have been shown to have a proximate similar accuracy, and all of these programs have been shown to outperform other methods for imputation such as FAST PHASE (Scheet P, 2006) and PLINK (Purcell S, 2007). Consequently, we perceived a substantial need to proposing a new technique for SNP Imputation with applying linear Discrimination and Clustering Analysis Algorithms. To evaluate the factors potentially affecting imputation accuracy rates (ARs), we used simulated data sets to investigate the effects of Linkage disequilibrium (LD), Minor allele frequency (MAF) of un-typed SNPs, marker density (MD), reference sample size (n) and the different numbers of SNPs in every haplotype block, in imputation accuracy rate (AR) and the performance of linear discriminant analysis and clustering Analysis as a SNP imputation method.

Key words: SNP Imputation, Clustering, Linear discrimination.

TABLE OF CONTENTS

GLOSSARY OF TERMS

Glossary of Genetic terms

(prepared by Guardeep Sagoo, University of Sheffield, UK)

Adenine (A):
purine base that forms a pair with thymine in DNA and uracil in RNA.

Allele:
one of the possible forms of a gene at a given locus. Depending on the technology used to type the gene, it may be that not all DNA sequence variants are recognised as distinct alleles.

Allele frequency:
often used to mean the population relative frequency (i.e. proportion) of an allele.

Base:
(abbreviated term for a purine or pyrimidine in the context of nucleic), a cyclic chemical compound containing nitrogen that is linked to either a deoxyribose (DNA) or a ribose (RNA).

Base pair (bp):
a pair of bases that occur opposite each other (one in each strand) in double strand DNA/RNA. In DNA adenine base pairs with thymine and cytosine with guanine. RNA is the same except that Uracil tacks the place of thymine.

Chromosome:
the self-replication threadlike structure found in cell. Chromosomes, which at certain stage of meiosis and mitosis consist of two identical sister chromatids, joined at the centromere, and carry the genetic information encoded in the DNA sequence.

Cytosine (C):
pyrimidine base that forms a pair with guanosine in DNA.

Deoxyribose:
the sugar compound found in DNA.

Diploid:
have two versions of each autosome, one inherited from the father and one from the mother. Compare with haploid.

Dominant allele:

result in the same phenotype irrespective of the other allele at the locus.

Eukaryote:

organism whose cells include a membrane-bound nucleus. Compare with prokaryote.

Exons:

parts of a gene that are transcribed into RNA and remain in the mature RNA product after splicing. An exon may code for a specific part of the final protein.

Gamete:

a sex cell, sperm in males, egg in females. Two haploid gametes fuse to form a diploid zygote.

Gene:

a segment of DNA that specifies a protein.

Genome:

all the genetic material of an organism.

Genotype:

the (unordered) allele pair(s) carried by an individual at one or more loci. A multilocus genotype is equivalent to the individual’s two haplotypes without the phase information.

Guanine (G):

purine base that forms a pair with cytosine in DNA.

Haploid:

has a single version of each chromosome.

Haplotype:

the allele at different loci on a chromosome. An individual's two haplotypes imply the genotype; the converse is not true, but in the presence of strong linkage disequilibrium haplotypes may be inferred from genotype with few errors.

Haplotype block:

A set of single-nucleotide polymorphisms (SNPs) on a single chromosome of a chromosome pair that are statistically associated and transmitted together when they are passed on from parent to child.

Heritability:
the proportion of the phenotypic variation in the population that can be attributed to underlying genetic variation.

Thymine (T):
pyrimidine base that forms the pair with adenine in DNA.

ABBREVIATIONS

AMOVA:	Analysis of Molecular Variance
ANOVA:	Analysis of Variance
CI:	Confidence Interval
DNA:	Deoxyribonucleic acid
DH:	Doubled Haploids
IBS:	Identical by State
IBD:	Identity by Descent
IIS:	Identity in State
LS:	Least-Squares
LDA:	Linear discriminant analysis
LD:	Linkage Disequilibrium
LLR:	Log-Likelihood Ratio
MCAR:	Missing completely at random
MD:	Marker density
MNAR:	Missing Not at Random

PCA:	Principal Component Analysis
REML:	Restricted Maximum likelihood
SNP:	single Nucleotide Polymorphism

1 INTRODUCTION

1.1 Background

Imputation is the substitution of some value for missing data, the practice of ‘filling in’ missing data with plausible values, is an attractive approach to analysing incomplete data. When substituting for a single value, it is known as "unit imputation"; when substituting for a component or a complete variable or item, it is known as "item imputation".

After imputing all missing values, the dataset can then be technically analysed using normal methods for complete data. We should ideally take into our account that there is a greater degree of uncertainty than if the imputed values had actually been observed.

There are many reasons behind why the data is missing, one nature of missing data could be ‘missing completely at random’ (MCAR), and it may be because the equipment malfunctioned, or the data were entered in an uncorrected way. When some data are missing completely at random, it means that the probability that an observation X_i is missing is unrelated to the value of X_i or to the value of any other variables, e.g. Human HapMap would not be considered as MCAR if Whites were more likely to omit reporting genotype than African Americans. MCAR is an important consideration, because in this case the analysis remains unbiased. We may lose power for our design, but the estimated parameters are not biased by the absence of data. If data are not completely missing at random then they are classified as ‘Missing Not at Random’ (MNAR). When the data are MNAR then we have the problem of a biased dataset, and the only way to obtain an unbiased estimate of parameters is to model the missing-ness or to write a model that accounts for the missing data (Dunning and Freedman 2008).The traditional treatments for missing data could be one of the following techniques,

A- **The list-wise deletion:** an entire record is excluded from analysis if any single value is missing.

B- **Random Within-Cell Hot-Deck:** where a missing value was imputed from a randomly selected similar record.

C- **Nearest Neighbour (within cell) Hot-Deck:** the missing value is imputed as the average of the covariate values of the nearest neighbours in the same data.

D- **Nearest Neighbour Cold-Deck:** this uses information from a previous survey or historical information on different dataset.

E- **Cell Mean:** here respondents are divided into classes. Then the cell mean for a particular class is used for all missing values in that class.

F- **Regression Methods:** this method uses a regression model to predict the value to be imputed.

In addition to the obvious advantage of allowing standard statistical methods of complete data analysis, applying Imputation in row data also has the important advantage of allowing the use of row information available from data collector but not available to an external data analyser such as a university social scientist analysing public information. This information may involve details and important knowledge about some characters of a group of people living somewhere sharing some Interests and Hobbies or may be this information coming from HR department from any company regarding some knowledge about interviewing procedures and reasons for nonresponse or any other important information for public-use files, or may be facts, such as street or post addresses of dwelling units, that cannot be placed on public-use files because of confidentiality restrictions. This kind of information, even though inaccessible to the user of a public-use file, can often improve the imputed values.

One more advantage of imputation by the database editor is that the missing data problem is handling (solving) once, rather than many times by the final users (Data-consumers). This implies consistency and stability of the data-bases across users, and a consequent consistency of results from using the same statistical analyses. Usually when we are applying the same statistical analysis method (e.g., multiple linear regression) on apparently the same data-base this will resulting in different answers and conclusions because of the differences in the way that Data-users and consumers handle missing data. This situation leads to unnecessary confusion and wasted resources. Imputation by the database constructor leads to greater consistency, stability and thereby to reduced costs of this type.

Just as there are obvious advantages to imputing one value for each missing value, there are obvious disadvantages of this procedure arising from the fact that the one imputed value cannot itself represent any uncertainty about which value to impute, if one value were really adequate, then that value was never missing. Hence, analyses that treat imputed values just like observed values generally systematically underestimate uncertainty, even assuming the precise reasons for nonresponse are known. Equally serious, imputation cannot represent any additional uncertainty that arises when the reasons for nonresponse are not known. The under-representation of uncertainty with imputation can be a major problem.

1.2 Multiple imputation

In this approach, each missing value is imputed several, say *M*, times and the variation between the *M* imputation is used to estimate the increase in variance due to nonresponse and imputation. Multiple imputation was originated by Rubin (1978, 1987). Unlike single imputation (filling in a single value for each missing one), multiple imputation was meant from the outset as a method to provide not only a solution to the missing data problem by imputation but also to reflect the uncertainty inherent in the imputations. The aim of multiple imputation is to provide multiply imputed data sets that should enable researchers to perform different kinds of analyses and obtain inferences with valid standard errors, confident intervals, and statistical test, in a simple way. Standard analyses are performed on each of the *M* data sets, and the results are combined using relatively simple formulae to obtain valid inferences.

The SAS code for multiple imputation algorithms can be found in Appendix-B (Paul D. Allison, 2005).

1.3 Literature review

1.3.1 Genome-wide imputation

Recent technology in high-throughput genotyping estimated that the human genome contains more than 7 million common single nucleotide polymorphisms (SNPs) with minor allele frequencies (MAF) about 5% (Barrett JC., Cardon L. R., 2006), and only a small fraction of these SNPs can be directly assayed using current high-density microarrays. Due to the linkage disequilibrium (LD) among neighbours markers, many un-typed or missing SNPs are highly correlated with one or more surrounding nearby assayed SNPs. Therefore, testing assayed SNPs for association to traits of interest will have some power to detect or prediction of un-typed causal SNPs. Further, if the assayed SNPs are uniformly distributed across the genome, maximal genetic coverage can be achieved (Hao K., Schadt E. E., 2008). The same in genome-wide association studies, where significant signals suggest association between phenotypes and causal SNPs in the surveyed genome region. To improve this type of association analysis, the genotypes of missing SNPs can be imputed or predicted based on nearby markers (SNP) and then directly tested for association with phenotypes of interest (Servin B., Stephens M., 2007). The main targets behind such studies are

1- Give researchers the possibility to combine experiments carried out on different. microarrays e.g. Affymetrix and Illumina arrays for genome-wide meta-analyses.
2- Allow researchers to easily replicating or comparing previous genes Discovered across array types.
3- Enables the investigation across a large number of SNPs to detecting the fine structure of the association peak, improving interpretation of results and location of the causal SNP.

1.3.2 Discriminant-based Imputation.

Discriminant-based imputation is a model based method for categorical or binary variables. The detailed imputation method is described in the methodology section (Applied multivariate statistical analysis; R. A. Johnson, D. W. Wichern, 2007).

When most of the predictor variables arc numerical an alternative method for logistic or polytomous regression imputation is Discriminant imputation. Categorical predictor variables are replaced by their corresponding dummy variables. The starting point of this method is the rule of Bayes

$$P(y = j|x) = \frac{P(x|y=j)P(y=j)}{\sum_{v=0}^{S-1} P(x|y=j)P(y=j)} \quad \text{....... (the rule of Bayes)}$$

$j = 0, \dots., s - 1$ (where in our case we using 1 and 2 only (as minor or major))

Under the assumption that $x = (x_1, x_1, \dots, x_p)$ given $y=j$ is normally distributed with a mean vector μ_j and covariance matrix Σ, the imputation model is

$$P(y = j|x) = \frac{f(x|\mu_j;\Sigma_j)\pi_j}{\sum_{v=0}^{S-1} f(x|\mu_v;\Sigma_v)\pi_v}$$

π_j is the probability that y = j and $f(.|\mu, \Sigma)$ is the probability density function of a multivariate normal distribution with a mean vector μ and covariance matrix Σ.

Discriminant imputation

Let y be a categorical imputation variable with categories 1 and 2 and *(x1, x2,…, xp)* the set of predictor variables resulting from replacing any categorical predictor variable of y (major and minor allele) by its corresponding dummy variables (1 and 2). Let n_j be the number of values of Y_{obs} in category j, $f(.|\mu, \Sigma)$ the probability density function of the multivariate normal distribution with mean vector μ and variance Σ, respectively. Under the assumption that the conditional probability distribution of $x = (x_1, x_1, \dots, x_p)$ given $y = j$ is a multivariate normal distribution with mean vector μ_j and covariance matrix Σ the underlying statistical model of discriminant imputation is given by

$$P(y = j|x) = \frac{f(x|\mu_j;\Sigma_j)\pi_j}{\sum_{v=0}^{S-1} f(x|\mu_v;\Sigma_v)\pi_v}$$

The previous model follows directly from substitution of $P(x|y = v) = f(x|\mu_v; \Sigma_v)$ and $P(y = v) = \pi_v$ into the formula of Bayes.

1.3.3 Nearest-neighbour (Clustering-based Imputation)

Nearest-neighbour imputation method (NIM) is an alternative form of hot-deck donor imputation. With this imputation, values from one record (the "donor") are used to replace the erroneous and missing values in another record (the "recipient"). The name "hot-deck" indicates that the donor and the recipient come from the same data set. Only records that are error-free may be used as donors.

To apply nearest-neighbour hot-deck imputation, a distance function *D(i,k)* must be defined that the measures the distance between two units *i* and *k*, where *i* is the item non-respondent and *k* is an arbitrary item respondent. The distance function *D(i,k)* can be defined in many different ways. A frequently used general distance function is the so called Minkowski distance:

$$D(i,k) = \left(\sum\nolimits_{j} \left|x_{ij} - x_{kj}\right|^{z}\right)^{\frac{1}{z}}$$

Where the *x* variables are numerical, and the sum is taken over all auxiliary variables; x_{ij} (x_{kj}) denotes the value of variables x_j in record *i (k)*. Let the smallest value of *D(i,k)* be attained for item respondent d *[d=arg min*$_k$ *D(i,k)]*, then respondent *d* is said to be the nearest-neighbour of the item non-respondent *i* and becomes its donor. For *z=2* the Minkowiski distance is the **Euclidean distance** and for *z=1* it is the so-called **city-block** distance. For larger *z,* large difference between x_{ij} and x_{kj} are "punished" more heavily. **In this Study we will use the Euclidean distance.**

Practically, we divided the dataset (including the records with missing values) into (n) clusters. Next, missing values of an instance *i* are patched up with the plausible values generated from *K*'s cluster. The following experiments will test the performance of the proposed method in genotype imputation task.

The Advantages of Nearest-neighbour imputation (NIM)

- The NIM works fast in practice. A limited number of imputation action is generated, using a limited number of donors, and one of these imputation actions is then selected.
- The NIM is able to handle numerical and categorical data simultaneously. At the time NIM was developed, all existing regression-based applications could handle either numerical data or categorical data, but not a combination (cf. Bankier et al. 1994).

1.4 Aim of the study

The general aim of this study is to test the performance of modern multivariate techniques like (linear discriminant and clustering analysis) in SNP imputation. But in genotype data there are many factors that affecting the imputation accuracy, this will be investigated by

1- Testing linear discriminant imputation and clustering imputation in low and high Linkage disequilibrium genome regions (LD).

2- Testing linear discriminant imputation and clustering imputation in different levels of Minor allele frequency genome regions (MAF).

3- Testing linear discriminant imputation and clustering imputation in different levels Marker density regions (HD, LD).

4- Testing linear discriminant imputation and clustering imputation with different Reference sample sizes (n).

5- Testing linear discriminant imputation and clustering imputation with different Haplotype block sizes (K).

   N.B. We measure the Haplotype block size by counting the number of SNPs per haplotype block, not by Centimorgan.

2. MATERIALS AND METHODS

2.1 Material and Data-simulation

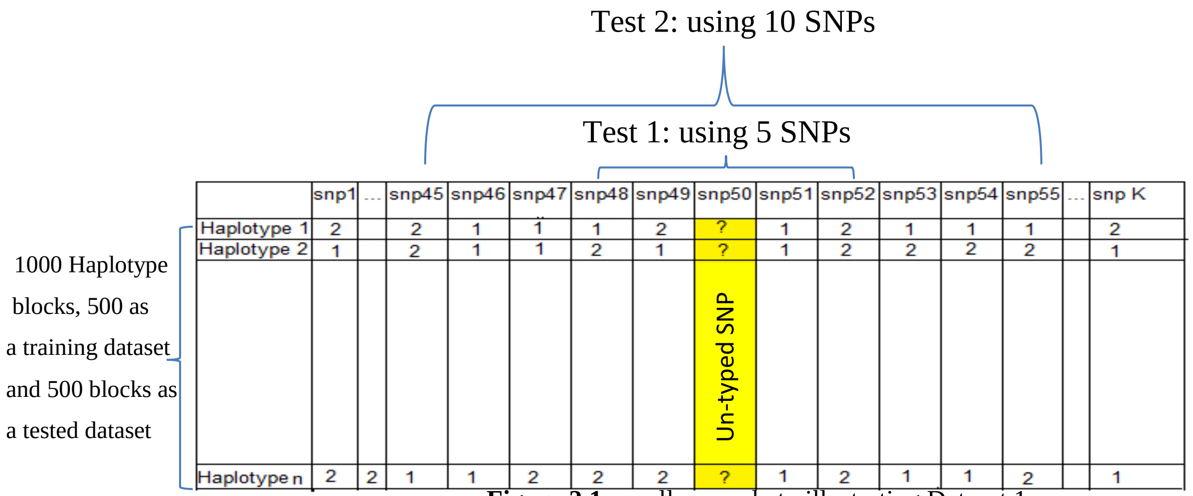
Many datasets have been simulated for this study (See Table1), each Dataset consisted of a number of haplotype blocks (rows of individuals) and a number of SNPs markers (columns of variables), simulated with some constants parameters and only one varied parameter (parameter under investigation), for example: to investigate the effects of Minor allele frequency (MAF) of un-typed SNPs in imputation accuracy rate by a given imputation method, we simulate a different datasets with a constant correlation between SNPs, a constant reference sample size (n) and a constant number of SNPs in every haplotype blocks, but with a different levels of Minor allele frequency (MAF) of un-typed loci in each datasets, then we measured the differences in accuracy rate coming from using different dataset with different MAF. For more details we summarised “how did we simulate each dataset and why each one have their own parameters” in the following descriptions.

- **Dataset 1**:

- Simulated to investigate the effect of the different numbers of SNPs (markers) in each haplotype block in imputation Accuracy rate, in a region of **low linkage disequilibrium**.

-Dataset1, consisted of 1000 haplotype blocks (1000 rows, 500 haplotypes as a training dataset and 500 haplotypes as a test dataset), with low correlation between SNPs = 0.2 (low average correlation between SNPs = low linkage disequilibrium region) and MAF = 0.5 (maximal MAF).

- And to investigate the effect of the different numbers of SNPs (markers) in each haplotype block, we measured the accuracy rate coming from 10 different imputation tests each one done by using different numbers of SNPs (markers) in each haplotype block (e.g. test 1 we used 4 SNPs surrounding the missing one, test 2 we used 9 SNPs surrounding the missing one and so on ..) (See figure 2.1)


Test 2: using 10 SNPs
Test 1: using 5 SNPs
1000 Haplotype blocks, 500 as a training dataset and 500 blocks as a tested dataset
Un-typed SNP

	snp1	...	snp45	snp46	snp47	snp48	snp49	snp50	snp51	snp52	snp53	snp54	snp55	...	snp K
Haplotype 1	2		2	1	1	1	2	?	1	2	1	1	1		2
Haplotype 2	1		2	1	1	2	1	?	1	2	2	2	2		1
Haplotype n	2	2	1	1	2	2	2	?	1	2	1	1	2		1

Figure 2.1: small example to illustrating Dataset 1

- To simulated dataset 1, we used the following “R-code"

```
> library(mvtnorm)    ## Loading (mvtnorm) Package: These functions providing the density function and a random number generator for the multivariate normal distribution with specific mean and a covariance matrix (sigma).

> nsim=1000;nSNP=100;rho=0.2 ##  A dataset with 1000 haplotype block, 100 SNPs markers and a constant correlation between markers = 0.2 (Not realistic, but we assume that haplotype blocks were inherited as one locus with no recombination between SNPs )

> sigma= matrix(rho,ncol=nSNP,nrow=nSNP)
> diag(sigma)=rep(1,nSNP)                    ## A designing matrix for the correlation between SNPs

> x <- rmvnorm(n=nsim, mean=rep(0,nSNP), sigma) ## Running the function with a given number of haplotype blocks (n), mean (mean), and correlation between SNPs(sigma).
> Dataset_1=apply(x,2,function(x) 1*(x<0)+2*(x >= 0))
                                              ## This function (apply) is implemented here to
                                                 convert the continuous variable to a discrete
                                                 variable ('1' and '2' for minor and major allele
                                                 frequencies ).

>cc=paste("snp",1:nSNP,sep="")
> colnames(Dataset_1)=cc                      ##  columns names

>save(Dataset_1,file=" Dataset_1.RData")      ##  Saving dataset
```

(See appendix-A).

- **Dataset 2**:

- Simulated to investigate the effect of the different numbers of SNPs (markers) in each haplotype block in imputation Accuracy rate, in a region of **High linkage disequilibrium**.

- Dataset2 Consisted of 1000 haplotype blocks (1000 rows, 500 haplotypes as a training dataset and 500 haplotypes as a test dataset), with high correlation between SNPs = 0.8 (high linkage disequilibrium region) and MAF = 0.5 (maximal MAF).

- And to simulate dataset 2, we used the previous “R-code" with changing the value of (`rho)` in sigma (correlation matrix) from 0.2 to 0.8. For more detail (See appendix-A).

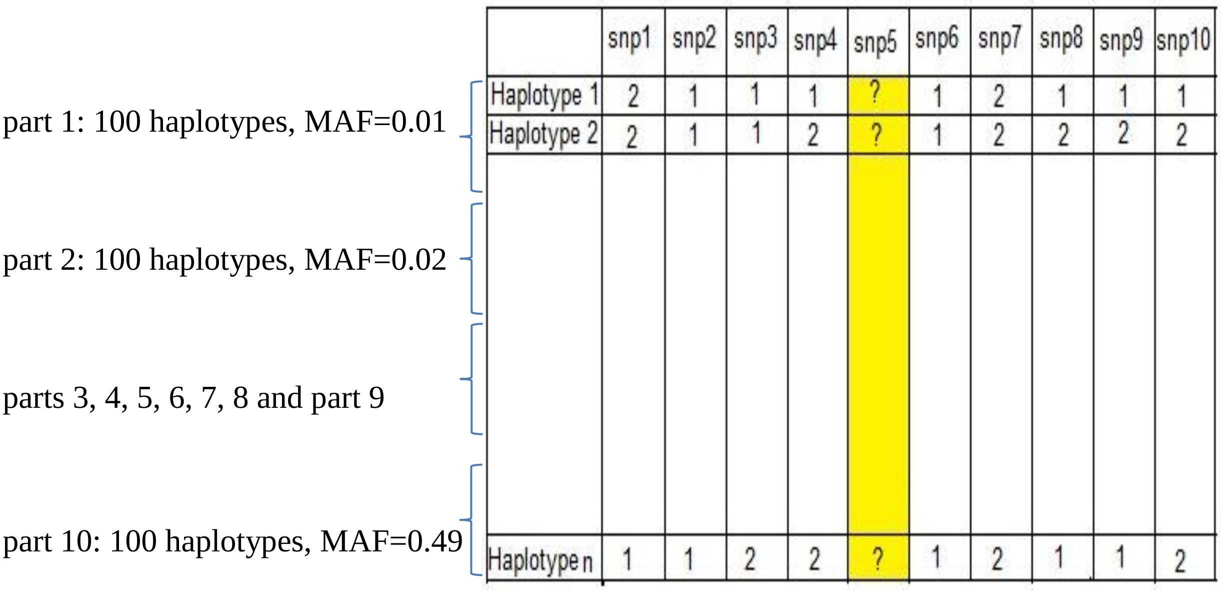
- **Dataset 3:**

- Simulated to investigate the effects of Minor allele frequency (MAF) of un-typed SNPs in imputation accuracy rate.

- We divided our dataset 3 into 10 different parts, each part consisted of 100 haplotype blocks (100 rows, 50 haplotypes as a training dataset and 50 haplotypes as a test dataset), with 10

SNPs in each haplotype block and low correlation between SNPs = 0.2 (to capture the major variation due to MAF, as we found previously from (dataset 1) that the variation in Accuracy rate is huge when using low correlation between only 10 SNPs in each haplotype block), then we varied the MAF of un-typed SNPs in all different 10 parts (e.g. part 1 with MAF=0.01, part 2 with MAF=0.02, part 3 with MAF=0.04, part 4 with MAF=0.06, part 5 with MAF=0.08, part 6 with MAF=0.1, part 7 with MAF=0.2, part 8 with MAF=0.3, part 9 with MAF=0.4, part 10 with MAF=0.49) (See figure 2.2))

N.B. “MAF is an attribute of a SNP, but here we simulated empirical datasets to test the performances of our methods under different conditions of genotype datasets with different parameters”


part 1: 100 haplotypes, MAF=0.01
part 2: 100 haplotypes, MAF=0.02
parts 3, 4, 5, 6, 7, 8 and part 9
part 10: 100 haplotypes, MAF=0.49

	snp1	snp2	snp3	snp4	snp5	snp6	snp7	snp8	snp9	snp10
Haplotype 1	2	1	1	1	?	1	2	1	1	1
Haplotype 2	2	1	1	2	?	1	2	2	2	2
Haplotype n	1	1	2	2	?	1	2	1	1	2

Figure 2.2: small example to illustrating Dataset 2.

- Then we simulated this dataset by “R" program, using the following code, for example: to simulate (part one) in Dataset 3 we used the next loop

```
nsim=100                          ##  number of Haplotype blocks
part1=rep(0,nsim)
for (i in 1:nsim){
  p=pmin(rbeta(1,3,7),0.05)       ## for more details about “beta function”
                                        (see appendix-A)
  part1[i]=sample(1:2,1,prob=c(p,1-p),replace=T)
}
```

The previous loop will give us a vector (of one SNP locus) containing 100 variables (minor and major alleles) with a frequency of 0.05 of minor alleles and 0.95 of major alleles.

For more details (see appendix-A).

- Dataset 4:

- Simulated to investigate the effects of marker density (MD) in imputation accuracy rate.

- In this case we duplicate the Dataset to 10 different datasets, each one varied from the others in their correlation between SNPs, but constants with other parameters.

- Each dataset Consisted of 1000 haplotype blocks (1000 rows, 500 haplotypes as a training dataset and 500 haplotypes as a test dataset), with 10 SNPs in each haplotype block and MAF = 0.50.

- To simulated dataset 4 we using “R-code" from Dataset 1 simulation-code.

- Dataset 5:

- Simulated to investigate the effects of reference sample size (n) in imputation accuracy rate.

- In this case to investigate the effects of reference sample size (n), we divided our dataset into 9 sup-datasets: 9 training datasets and 9 test datasets.

- **Sup-datasets 1:** consisted of 100 haplotypes as training-dataset and the rest 900 haplotypes as test-dataset.
- **Sup-datasets 2:** consisted of 200 haplotypes as training-dataset and the rest 800 haplotypes as test-dataset.
  and so on until the sup-datasets 9
- **Sup-datasets 9:** consisted of 900 haplotypes as training-dataset and the rest 100 haplotypes as test-dataset.

- Each Test consisted of 1000 haplotypes and 10 SNPs, with correlation between SNPs = 0.20 and MAF =0.10.

- And to divide such dataset we used the following “R-code".

```
Training_1 <-(as.data.frame(Dataset_5[1:100,]))

Test_1 <-(as.data.frame(Dataset_5[101:1000,]))
```

Table 1: Presentation of all datasets used in imputation experiment.

Dataset	Test	Correlation	MAF %	No. haplotypes	No. SNP
1	No. of SNPs in **(LLD)** region	0.2	49	1000	Vary
2	No. of SNPs in **(HLD)** region	0.8	49	1000	Vary
3	Minor allele frequency **(MAF)**	0.2	Vary	1000	10
4	Marker density **(MD)**	Vary	49	1000	10
5	Reference sample size **(n)**	0.2	10	1000	10

2.2 Methods and implementation

2.2.1 Linear discriminant analysis

Linear discriminant analysis (LDA): is sometimes known as Fisher's linear discriminant analysis, after its inventor, Ronald A. Fisher, who published it in The Use of Multiple Measures in Taxonomic Problems (1936). It is typically used as a feature extraction step before classification. LDA and the related Fisher's linear discriminant are methods used in statistics, pattern recognition and machine learning to find a linear combination of features which characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. LDA doesn't change the location but only tries to provide more class separability and draw a decision region between the given classes.

Different features of functional signal may have different significance for recognition and may not be independent. Classical linear discriminant analysis provides a way to combine such features in a discriminant function. Such a function, when applied to a pattern, yields an output that is an estimate of the class membership of this pattern. The discriminative technique provides minimization of the error rate of classification (Afifi and Azen, 1979).

Let us assume that any given variables of SNPs (in a given haplotype block) can be described by vector *X* of *p* characteristics *(x_1, x_2,…, x_p)*, that can be measured ($x_{1=1}$ for major allele and $x_{1=2}$ for minor allele) . The linear discriminant analysis procedure finds a linear combination of the measures (called the linear discriminant function or LDF), that provides maximum discrimination between major alleles (class 1 or ' π_1') and minor alleles (class 2 or 'π_2').

$$Z = \sum_{i=1}^{p} a_i x_i \quad …….. \text{ (LDF)}$$

The LDF classifies *X* into class 1 if $Z > c$ and into class 2 if $Z < c$. The vector of coefficients $(\alpha_1, \alpha_2, \dots, \alpha_p)$ and threshold constant c and derived from the training set by maximizing the ratio of between-class variation of z to the within-class variation and are equal to (Afifi and Azen, 1979):

$$\vec{a} = s^{-1}(\vec{m}_1 - \vec{m}_2)$$

And

$$\vec{c} = \vec{a}(\vec{m}_1 + \vec{m}_2)/2,$$

Where $\vec{m}_i$ are the sample mean vectors of characteristics for class 1 and class 2, respectively; s is pooled covariance matrix of characteristics

$$S = \frac{1}{n_1 + n_2 - 2}(s_1 + s_2),$$

In which S_i are estimated covariance matrices and n_i is the sample size of class i. Based on these equations, we can calculate the coefficients of the LDF and threshold constant *c* using the values of characteristics of major and minor alleles from the training datasets and then test the accuracy of the LDF on the test dataset. The significance of a given characteristics or a set of characteristics can be estimated by the generalized distance between two classes (called the Mahalanobis distance), given by

$$D^2 = (\vec{m}_1 - \vec{m}_2)' S^{-1}(\vec{m}_1 - \vec{m}_2),$$

Which is computed based on values of the characteristics in the training dataset of classes 1(major alleles) and 2(minor alleles).

Implementation

Example 3.1 (a set of training and test SNPs-genotypes datasets from our simulated dataset) Consider two groups of alleles in SNP3: π_1, minor allele (with number 1), and π_2, those major allele (with number 2). The training-dataset consisted of 5 different SNP genotyped (5 variables). We want to use this training dataset to construct a classification rule (equation) which can be used to classify SNP3 in a new dataset (test dataset) where SNP3 is missing. (See table 3.2.2 and 3.2.3)

Table 3.2.2: Training dataset

Hap.	SNP1	SNP2	SNP3	SNP4	SNP5
1	2	1	2	1	1
2	1	1	1	2	2
3	2	2	1	1	2
4	1	1	2	1	1
5	1	1	2	2	2

Table 3.2.3: Test dataset

Hap.	SNP1	SNP2	SNP3	SNP4	SNP5
6	1	1	?	1	2
7	2	2	?	1	2

R commands

```
lda(SNP3 ~ SNP1 + SNP2 + SNP4 + SNP5, data = Training)
Coefficients of linear discriminants: LD1
SNP1  1.939638e-16
SNP2 -1.718108e+00
SNP4 -1.145405e-01
SNP5 -1.489027e+00
```

So the LDA model should be

```
SNP3 ≈μ+SNP1 (1.939638e-16) +SNP2 (-1.718108e+00) +SNP4 (-
1.145405e-01) + SNP5 (-1.489027e+00) + e

e: error
```

Now, in order to identify the missing SNP number 3 in the Test dataset, e.g. haplotype number 6

```
predict(DAModel.5, data.frame('SNP1'=1, 'SNP2'=1, 'SNP4'=1,
'SNP5'=2))
$class
[1] 2
```

So the SNP3 in haplotype 6 (record no. 6) expected to = 2 (major allele class).

2.2.2 Clustering analysis

Grouping, or clustering, is distinct from the classification or discrimination discussed in the previous method. Classification pertains to a known number of groups, and the operational objective technique in that no assumptions are made concerning the number of groups or the group structure. Grouping is done on the basis of similarities or distances (dissimilarities).The inputs required are similarity measures or data from which similarities can be computed.

To summarize, the basic objective in cluster analysis is to discover natural grouping of the items (or variables). In turn, we must first develop a quantitative scale on which we measure the association (similarity) between objects.

Similarity measures

Most efforts to produce a rather simple group structure from a complex data set require a measure of “closeness,” or “similarity.” There is often a great deal of subjectivity involved in the choice of a similarity measure. Important considerations include the nature of the variables (discrete, continues and binary), scales measurements (nominal, ordinal, interval and ratio), and subject matter knowledge.

When items (unit or cases) are clustered, proximity is usually indicated by some sort of distance. By contrast, variables are usually grouped on the basis of correlation coefficients or similar measures of association.

Distance and similarity coefficients for pair’s items

Distance: straight line or Euclidean.

If we consider the point $P = (\chi_1, \chi_2)$ in the plan, the straight -line distance,$d(O, P)$, from P to the origin $O = (0,0)$is, according to Pythagorean theorem, $d\ (O, P) = \sqrt{{x_1}^2 + {x_2}^2}$

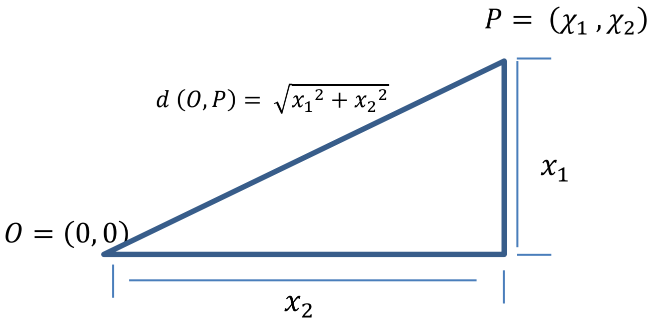


Figure 3.1 Distance given by the Pythagorean Theorem

The situation is illustrated in Figure 3.1 In general, if the point *P* has *p* coordinates so the $P = (x_1, x_2, \dots, x_p)$, the strait line distance from *P* to the origin $O = (0, 0)$ is

$d\ (O, P) = \sqrt{{x_1}^2 + {x_2}^2 + \dots + {x_p}^2}$ ---------- (3.1)

All points$(x_1, x_2, \dots, x_p)$ that lie a constant squared distance, such as c^2, from the origin satisfy the equation

$d^2\ (O, P) = {x_1}^2 + {x_2}^2 + \dots + {x_p}^2 = c^2$ ------ (3.2)

Because this is the equation of a hypersphere (a circle if p =2), points equidistant from the origin lie on a hypersphere.

The strait-line distance between two arbitrary points *P* and *Q* with coordinates

$P = (x_1, x_2, \dots, x_p)$, and $Q = (y_1, y_2, \dots, y_p)$ is given by

$d\ (P, Q) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_p - y_p)^2}$ -------- (3.3)

Using Matrix and vector notations

$$x' = [x_1, x_2, \dots, x_p], \text{ and } y' = [y_1, y_2, \dots, y_p] \text{ is given by}$$

$$d\ (x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_p - y_p)^2}$$

$$= \sqrt{(x - y)'(x - y)} \quad \text{-----------} \quad (3.4)$$

The statistical distance between the same two observations is of the form

$d\ (x, y) = \sqrt{(x - y)'\ A\ (x - y)}$ ----------- (3.5)

Ordinary A =S$^{-1,}$ where S contain the sample variances and co-variances. However, without prior knowledge of the distinct groups, these sample quantities cannot be computed. For this reason, Euclidean distance is often preferred for clustering.

Another distance measure is the previously mentioned Minkowski metric.

$d\ (x, y) = \left[\sum_{i=1}^{p} |x_i - y_i|^m\right]^{1/m}$ -------- (3.6)

For *m = 1, d(x,y)* measures the “city-block” distance between two point in *p* dimension. For *m = 2, d(x,y)* becomes the Euclidean distance. In general, varying *m* changes the weight given to large and smaller differences.

Two additional popular measures of “distance” or dissimilarities are given by the Canberra metric and the Czekanowski coefficient. Both of these measures are defined for nonnegative variables only. We have

Canberra metric: $$d\ (x, y)\ =\ \sum_{i=1}^{p} \frac{|x_i - y_i|}{(x_i + y_i)} \text{ ---------- } (3.7)$$

Czekanowski coefficient: $$d\ (x, y)\ = 1 - \frac{2 \sum_{i-1}^{p} \min(x_i, y_i)}{\sum_{i=1}^{p} (x_i + y_i)} \text{ --------- } (3.8)$$

Whenever possible, it is advisable to use “true” distance that is, distances satisfying the distance properties of *d(P,Q) ≤ d(P,R)+d(R,Q) “triangle inequality”* where *R* is any other intermediate point, for clustering objects. On the other hand, most clustering algorithms will accept subjectively assigned distance number that may not satisfy, for example, the triangle inequality.

Hierarchical clustering methods

In my point of view, I found that the Hierarchical clustering methods (HLM) is the most suitable way to measure the distance and grouping the haplotype blocks into a clusters according to the SNPs markers variation.

We can rarely examine all grouping possibilities, even with the largest and fastest computers. Because of this problem, a wide variety of clustering algorithms have emerged that find “reasonable” clusters without having to look at all configurations.

Hierarchical clustering techniques proceed by either a series of successive mergers or a series of successive divisions. *Agglomerative hierarchical methods* start with the individual object. Thus, there are initially as many clusters as objects. The most similar objects are first grouped, and these initial groups are merged according to their similarities. Eventually, as the similarities decreased, all subgroups are fused into single cluster.

Division hierarchical methods work on the opposite direction. An initial single group of objects is divided into two subgroup such that the objects in one subgroup are “far from” the objects in the other. These subgroups are then further divided into dissimilar subgroup; he process continues until there are many groups as objects that is, until each object forms a group.

The result of both agglomerative and division methods may be displayed in the form of a two-dimensional diagram known as *dendrogram*. As we shall see, the dendrogram illustrate the mergers or division that have been made at successive level.

In this Thesis and in our application in SNP imputation, we shall concentrate on agglomerative hierarchical procedures and, in particular, *linkage methods*.

Linkage methods are suitable for clustering items, as well as variables. This is not true for all hierarchical agglomerative procedures. We shall discuss, in turn, *single linkage* (minimum distance or nearest neighbours), *complete linkage* (Maximum distance or farthest neighbour) and *average linkage* (average distance). The merging of clusters under the three linkage criteria is illustrated schematically in figure 3.2

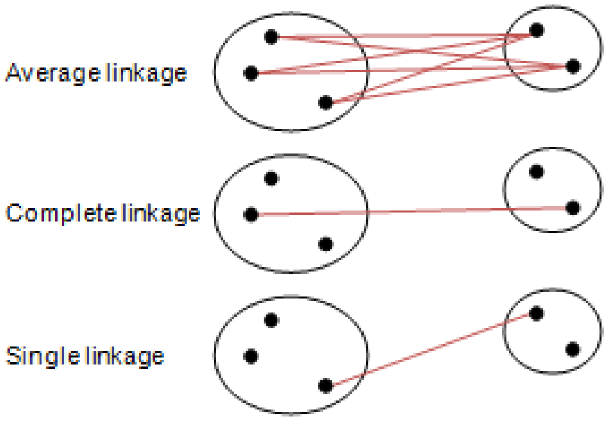


Figure 3.2 the three linkage methods.

From the figure, we see that the single linkage results when groups are fused according to the distance between their nearest members. Complete linkage occurs when groups are fused according to the distance between the farthest members. For average linkage, groups are fused according to the average distance between pairs of members in the respective test.

In our study we select the complete linkage method because our data have only two type of variable (minor allele=1 and major allele =2) and we was searching for a method that can give us the maximum differences between clusters in our genotypic data.

Complete linkage

In complete linkage clustering: the distance (similarity) between clusters is determined by the distance (similarity) between the two elements, one from each cluster, that are most *distant.* Thus, complete distance ensures that all items in a cluster are within maximum distance (or minimum similarity) of each other.

Implementation

Example 3.2 (Clustering using complete linkage and the Euclidean distance)

To illustrate the application of Clustering in SNP imputation, we consider the hypothetical correlation distances between pairs of 5 Haplotypes, each haplotype have (n) numbers of SNPs as follows:

e.g correlation distance between haplotype 1 and haplotype 3:

$d\ (h_1, h_3)\ = [1 - \ cor(h_1, h_3)] * 10$ …… to simplifying the calculation

$$D = \{d_{ik}\} = \begin{array}{c|ccccc} hap_1 & 0 & & & & \\ hap_2 & 9 & 0 & & & \\ hap_3 & 3 & 7 & 0 & & \\ hap_4 & 6 & 5 & 9 & 0 & \\ hap_5 & 10 & 10 & (2) & 8 & 0 \end{array}$$

At the 1st stage, objects 3 and 5 are merged, since they are most similar. This gives the cluster (35).

At stage 2, we compute

$d_{(35)1}$ = max {d_{31},d_{51}} = max {3,10} = 10

$d_{(35)2}$ = max {d_{32},d_{52}} = 10

$d_{(35)4}$ = max {d_{34},d_{54}} = 9

And the modified matrix becomes

$$\begin{array}{r} hap_{(35)} \\ hap_1 \\ hap_2 \\ hap_4 \end{array}\begin{bmatrix} 0 & & & \\ 10 & 0 & & \\ 10 & 9 & 0 & \\ 9 & 6 & (5) & 0 \end{bmatrix}$$

The next merger occurs between the most similar groups, 2 and 4, to give the cluster (24).

At stage 3, we have

$d_{(24)(35)}$ = max $\{d_{2(35)}, d_{4(35)}\}$ = max {10,9} = 10

$d_{(24)1}$ = max $\{d_{21}, d_{41}\}$ = max {9,6} = 9

And the distance matrix

$$\begin{array}{r} (35) \\ (24) \\ 1 \end{array}\begin{bmatrix} 0 & & \\ 10 & 0 & \\ 10 & 9 & 0 \end{bmatrix}$$

The next merger produces the cluster (124). At the final stage, the group (35) and (124) are merged as the single cluster (12345) at level

$d_{(124)(35)}$ = max $\{d_{1(35)}, d_{(24)(35)}\}$ = max {10,10} = 10

The dendrogram is given in Figure 3.3

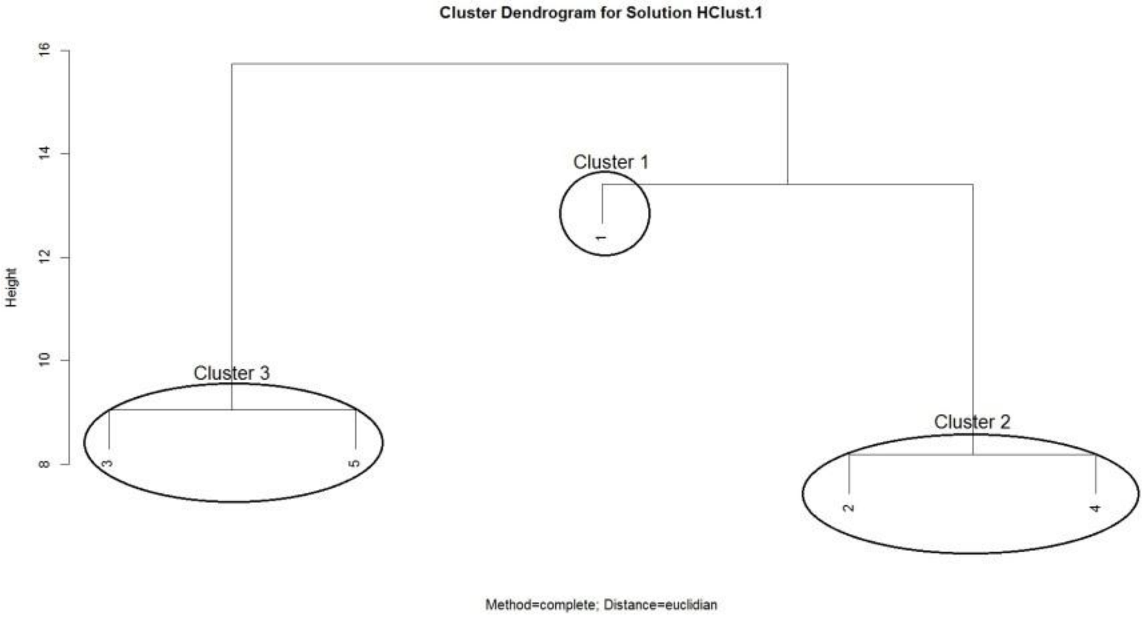


Figure 3.3 complete linkage dendrogram for distances between five haplotypes

R commands

```
➢ hclust(dist(model.matrix(~-1 +
  hap1+hap2+hap3+hap4+hap5,Dataset)), method= "complete")
➢ plot(HClust.1, main= "Cluster Dendrogram for Solution
  HClust.1", sub="Method=complete;Distance=euclidian")
```

- `Dataset$hclus.label <- assignCluster(model.matrix(~-1 + hap1 + hap2 + hap3 + hap4 + hap5, Dataset), Dataset, cutree(HClust.1, k = 3))`

The last line to add the cluster number, or to assign every haplotype to its cluster as follows

Hap.	**SNP1**	**SNP2**	**….**	**SNPn**	**Cluster**
1	2	1	….	1	1
2	1	1	….	2	2
3	2	2	….	2	3
4	1	1	….	1	2
5	1	1	….	2	3

In this case we can use this result in imputation the missing SNPs as follows:

Any missing SNP in a given haplotype can be substituted with a known SNP in another haplotype sharing the same cluster, e.g. if a haplotype #3 have a missing SNP we can substitute it with the same SNP in the Haplotype #5 as they are in the same cluster.

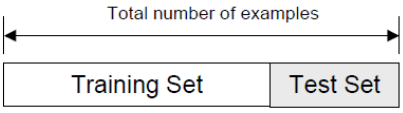
2.3 Validation

The holdout method

The holdout method is the simplest method of cross validation. Each data set is split into two parts or sets, called the training-dataset (reference data-set) and the test-dataset. In LDA the prediction model is fit using the training data-set only. But in Clustering the prediction model is fit using the training data-set and the Test data-set. Usually we using 50% training data-set in this study, except in the last experiment where we measuring the effect of the size of the training dataset (where we varying the size of the training dataset (n)).

Then the same model is used to predict the outcome values for the data in the test data-set (only in LDA, were it has never seen these output values before).

The errors it makes (when we using the model to predict the outcome) are accumulated to give the mean absolute test set error, which is used to evaluate the model, in other word, the accuracy of this model counted by measuring the correlation between the true and the predicted value of the imputed SNP vector.



The advantage of this validation method is that It gives us the possibility to measure how much the size of the training-dataset (reference data-set) can affect the imputation accuracy, because in real life usually the data set contain some completed data (which can considered as training-dataset) and the rest have some missing values (considered as test-dataset)

However, the evaluation usually depends heavily on which data points end up in the training-dataset and which end up in the test-dataset. And **estimating the error rate** will be misleading if we happen to get an “unfortunate” split.

4.0 RESULTS

4. 1 Comparison between the performance of LDA and Clustering analysis in SNP imputation.

1- **Figure 1:** shows the effects of size of haplotype block (number of SNPs per haplotype), on imputation accuracy rate (AR) using low and high linkage disequilibrium dataset (LLD, HLLD). When LDA is used for imputation with constant MAF =49% and low linkage disequilibrium data the accuracy rate ranging from 60% (using 5 SNPs) to 70% (using 100 SNPs), while with High linkage disequilibrium data the accuracy rate ranging from 88% (using 5 SNPs) to 93% (using 100 SNPs). This is a high LD dataset AR is generally substantially higher and there is less improvement by increasing the number of SNPs. (See appendix-C, Table 4.1).

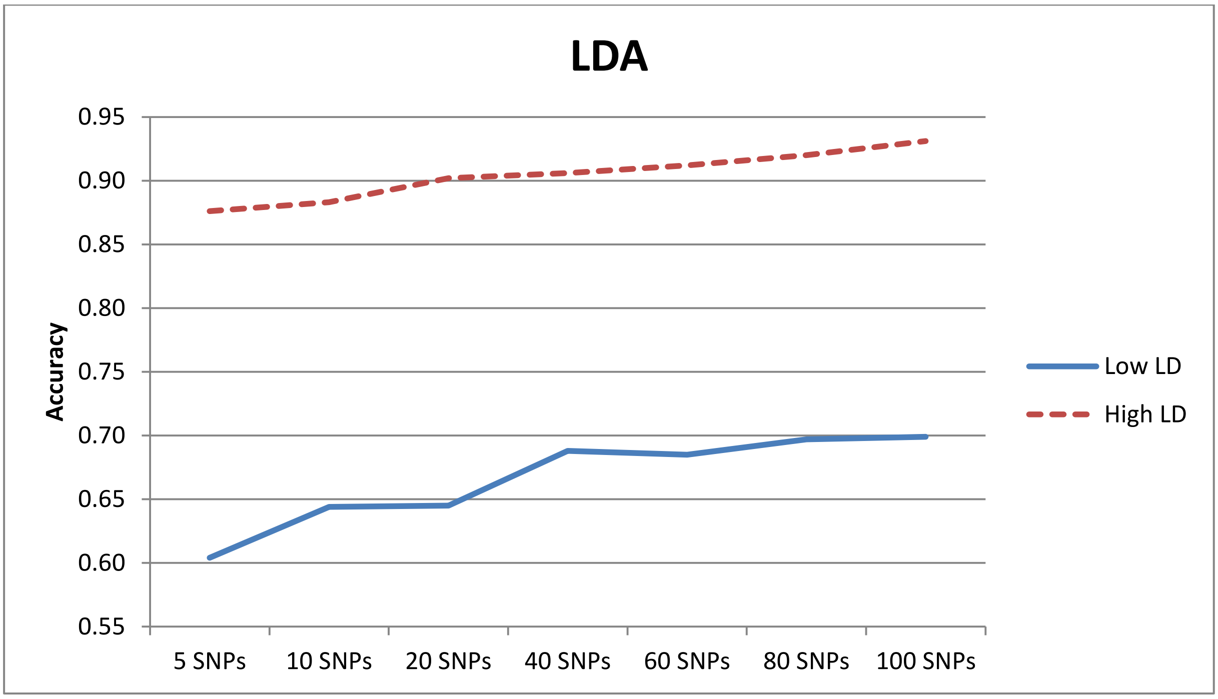


Figure 1: The effects of using Low and High linkage disequilibrium dataset on Accuracy rate of LDA in imputation.

2- **Figure 2:** Shows the effects of number of SNPs surrounding the missing one, in imputation accuracy rate (AR) using low and high linkage disequilibrium dataset (LLD, HLD). When clustering is used for imputation with constant MAF =49% and low linkage disequilibrium data the accuracy rate ranging from 55% (using 5 SNPs) to 71% (using 100 SNPs), while with High linkage disequilibrium data the accuracy rate ranging from 75% (using 5 SNPs) to 91% (using 100 SNPs). Generally clustering is less accurate than LDA and need more SNPs to reach high accuracy. (See appendix-C, Table 4.2).

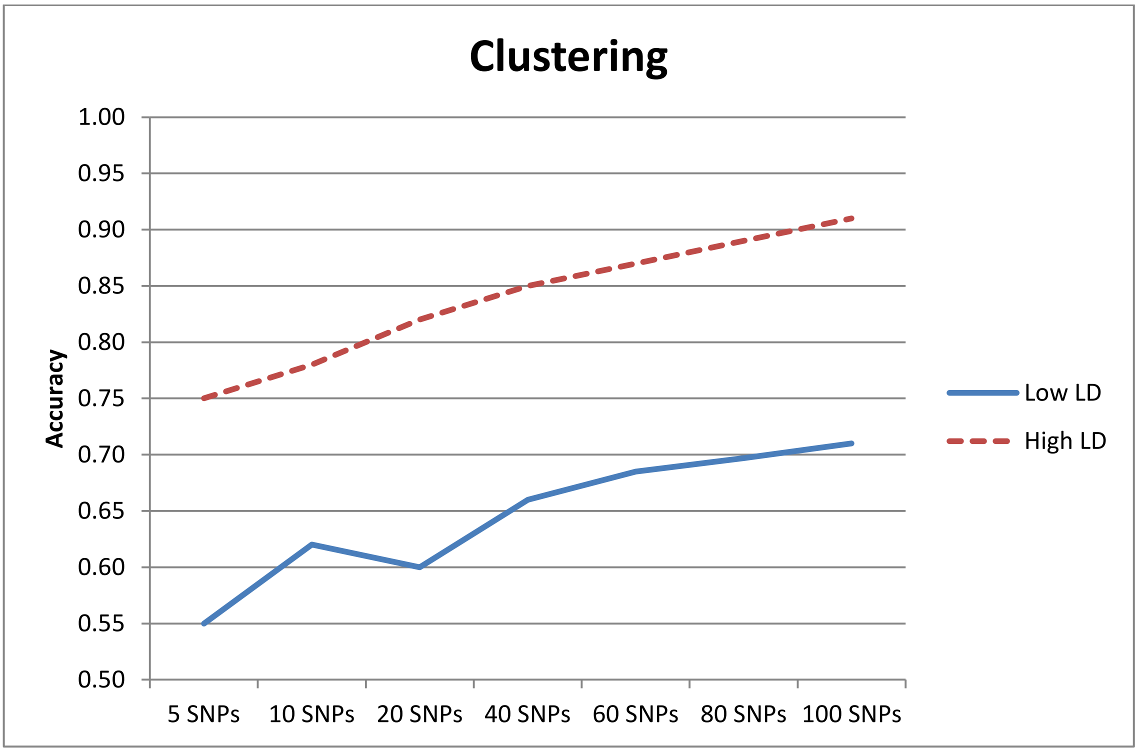


Figure 2: The effects of no.SNPs using Low and High linkage disequilibrium dataset on accuracy rate of Clustering in imputation.

3 - **The effects of Minor allele frequency (MAF): Figure 3.**

Using LDA with constant correlation between SNPs = 0.10 and using the surrounding 10 SNPs the accuracy rate ranging from 0.99 (using MAF=0.10) to 0.75(using MAF=0.49), (**See Figure 3**). While using Clustering with constant correlation between SNPs = 0.10 and using the surrounding 10 SNPs the accuracy rate ranging from 92% (using MAF=0.10) to 69% (using MAF=0.49). It seems that AR is much more accurate when MAF is low compared to when it is high. A lower MAF usually corresponds to a stronger LD with nearby markers and the recombination plays a primary role in LD decay (Yu-Fang Pei., 2008). (See appendix-C, Table 4.3).

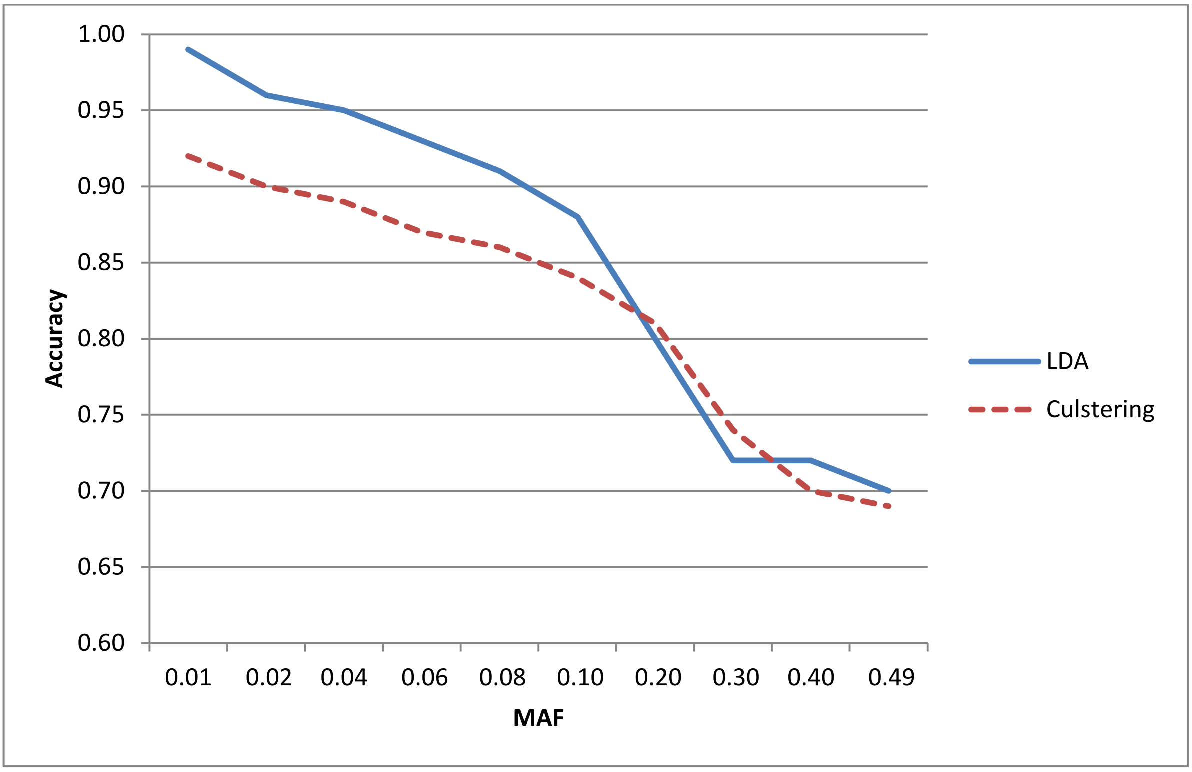


Figure 3: The effects of Minor allele frequency on accuracy rate using LDA and Clustering.

4 - **The effects of marker density (MD):** Figure 4.

Using LDA With constant MAF =50% and using the surrounding 10 SNPs the accuracy rate ranging from 61% (using Corr. =0.10) to 97% (using Corr. =0.90). (**See Figure 4**) While using Clustering with constant MAF =49% and using the surrounding 10 SNPs the accuracy rate ranging from 55% (using Corr. =0.10) to 94% (using Corr. =0.90). (See appendix-C, Table 4.4).

Here, we measure the effect of Marker density by varying the correlation between markers (SNPs).

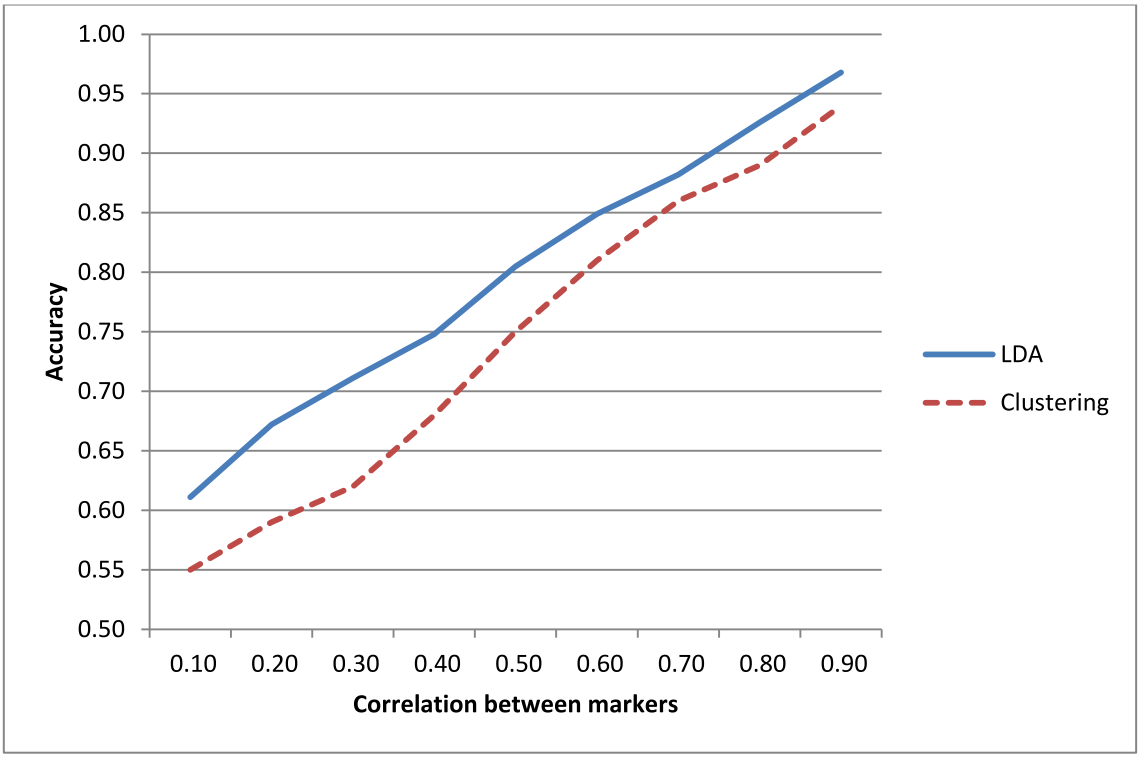


Figure 4: The effects of marker density on accuracy rate using LDA and Clustering.

5 - **The effects of reference sample size (n):** Figure 5.

Using LDA, with constant MAF =10%, constant correlation between SNPs = 0.10 and using the surrounding 10 SNPs the accuracy rate ranged from 72% (using n =0.10) to 83% (using n =0.90), while using Clustering, with constant MAF =10%, constant correlation between SNPs = 0.10 and using the surrounding 10 SNPs the accuracy rate ranged from 33% (using n =0.10) to 79% (using n =0.90). This shows that clustering needs higher (n) to reach high accuracy. (See appendix-C, Table 4.5).

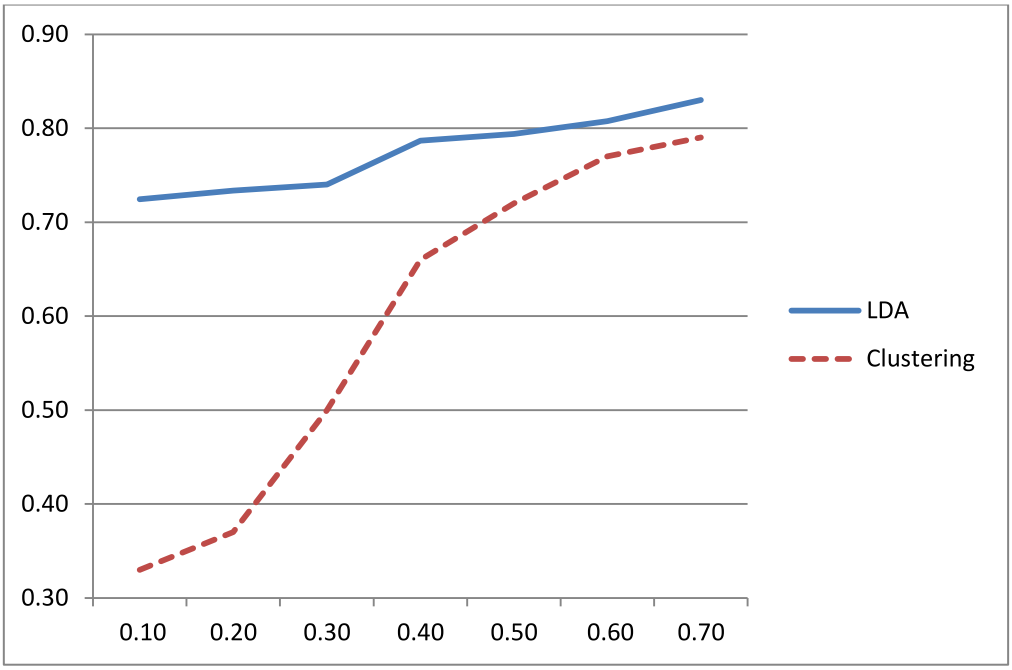


Figure 5: The effects of reference sample size on accuracy rate using LDA and Clustering.

5. DISCUSSION AND CONCLUSION

This study compared two different approaches (Discriminant-based SNP imputation and Nearest-neighbour or Clustering-based SNP imputation) using haplotype blocks instead of individual markers or all available markers. The average number of SNPs per haplotype blocks varying from 5 SNPs (in low LD region) to 100 (in High LD region). To investigate the performance of these two methods we simulated a group datasets each one simulated to test the effects of Linkage disequilibrium (LD), Minor allele frequency (MAF) of un-typed SNPs, marker density (MD), reference sample size (n) and the different numbers of SNPs in every haplotype block, in imputation accuracy rate (AR) and the performance of The Linear discriminant analysis and Clustering Analysis as a SNP imputation method. The dataset was also split in a training dataset and test dataset. The methods were validated using the holdout method then measuring the correlation between the true and imputed SNP in test dataset.

The performance of the elementary imputation methods, clustering and discrimination is generally good. However, to compare the performance of each algorithm with the currently used methods like in MACH, BEAGLE, and IMPUTE, more test experiments are needed to be conducted. Furthermore, to be sure that the algorithms are reliable, the same data sets should be used to run the experiments. Like any simulation study, this one has its limitations and advantages in some cases like:

1- In low LD region, the clustering-based method can use the correlation between records instead of the correlation between markers in the imputation process.

2- The Discriminant-based method also can handle numerical and categorical data simultaneously without rounding-up the results (which can affect the accuracy of imputation).

But in optimal state of genotype data (in High LD, low MAF, and high density haplotype blokes) both methods (Clustering and discrimination) were working efficiently, and the accuracy can reached 89 %.

Further studies and experiments are necessary before one can conclude whether the establishment of Discriminant-based and Clustering-based SNP imputation is feasible or not. The Clustering-based SNP imputation models show a lot of promise for SNP imputation (and in Microarray analysis in general) based on the associations between records instead of using the association between markers.

Results obtained had many similarities with those obtained both from Discriminant-based imputation and Clustering-based SNP imputation approaches in similar datasets.

Linear discrimination can be considered as a complement algorithm for Clustering especially when applied to noisy data in what we can call "Cluster-based pattern discrimination CPD", which differs from standard clustering by being simultaneous subspace selection via linear discriminant analysis (LDA).
LDA is the most widely used in the two dimensions or categorised data. However, both statistical methods suffer from some deficiencies. Clustering analysis has the problem of selecting different values of K (i.e. number of nearest neighbouring haplotype records). Using different K-values results in different performance of the algorithms which in turn affects the final evaluation for the method accuracy. So that we propose to test the optimal K-value each time the algorithm is used. Finally, searching for a new technique and a new application or a new demonstration of Discriminant and Clustering analysis was the main interest of this thesis because nowadays the application of the modern statistical techniques such LDA, Clustering, PCA, PLS …and etc., are so important considerations in the field of Bioinformatics and Applied statistic.

REFERENCES

Abramowitz, M., and I. A. stegun. *Handbook of Mathematical function*. U.S. Department of Commerce, national Bureau of Standard Mathematical Series. 55, (1964).

Adriaans, p., AND d. Zantinge. *Data Mining.* Harlow, England: Addison-Wesely, (1996).

Alkes L Price, Nick J Patterson, Robert M Plenge, and David Reich. *Principal components analysis corrects for stratification in genome-wide association studies.* Nature Publishing Group, (2006).

Anderberg, M. R. *Cluster Analysis for Applications.* New York: Academic Press, (1973).

Berry, M. I. A., and G. Linoff. *Data Mining technology: For Marketing, Sales and Customer relationship Management* (2nd ed). New York: John Wiley, (2004).

Berthold. M., and D. J Hand. *Intelligent Data analysis* (2nd ed.), Berlin, Germany, Springer-Verlag, (2003).

Bryk, A.S. and Raudenbush, S.W. *Hierarchical Linear Models*. Sage, Newbury Park, (1992).

Celeux, G., and G. Govaert. “Gaussian Parsimonious Clustering Models” *Pattern recognition*, (1995) 28, 781-793.

Cormack, R. M. “A Review of Classification (with discussio).” *Journal of the Royal Statistical Society (A)*, 134, no. 3 (1971), 321-367.

Everitt, B. S., S. Landau and M. Leese. *Cluster Anlysis (4th ed).*London: Hodder Arnold, (2001).

Fraley, C., and A. E. Raftery. “Model-Based Clustering, Discriminant Anlysis and Density Estimation.” *Journal of the American statistical Association,* 97 (2002), 611-631.

Gilks, W.R., Richardson, S. & Spiegelhalter, D.J. (Eds.) *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London, (1996).

Grower, J. C. “Some Distance Properties of latent Root and Vector Methods Used in Multivariate Analysis.” *Biometrika,* 53 (1966), 325-338.

Gower, J.C. “Multivariate Analysis and Multidimensional.” *The Statistician,* 17 (1967), 13-25.

Gower, J. C., and D. J. Hand. *Biplots.* London: Chapman and Hall, (1996).

Greenacer, M. J. “Correspondence Analysis of Square Asymmetric Matrices,” *Applied Statistics,* 49, (2000) 297-310.

Greenacre, M. J. Theory and applications of Correspondence Analysis. London: Academic Press, (1984).

Hand, D., H. Mannila, and P. Smyth. *Principles of Data Mining*. Cambridge, MA: Mit Press, (2001).

Hartigan, J. A. *Clustering Algorithms*. New York: John Wiley, (1975).

Hastie, T. R., R. Tibshirani and J. Friedman. *The Elements of Statistical Learning: Data Mining, inference and Prediction.* Berlin, Germany: Spring-Verlag, (2001).

Kennedy, R. L., L. Lee, B. Van Roy, C.D. Reed, and R. P. Lippmann. *Solving Data Mining Problems through Pattern Recognition*. Upper Saddle River, NJ: Prentice Hall, (1997).

Kruskal, J. B. “Multidimensional Scaling by Optimization Goodness of Fit to a Nonmetric Hypothesis.” *Psychometrika*, 29, no. 1 (1964), 1-27.

Kruskal, J.B. “Non-metric Multidimensional Scaling. “A Numerical Method.” *Psychometrika*, 29, no. 1 (1964), 115-129.

Kruskal, J. B., and M. Wish, “Multidimensional Scaling.” *Sage University Paper Series on Quantitative Applications in the Social Sciences*, 07-011. Beverly Hills and London: Sage Publications. (1978).

La Pointe, F-J, and P. Legendre. “A Classification of Pure Malt Scotch Whiskies” *Applied Statistic*, 43, no. 1 (1994), 237-257.

Le Roux, N. J., and S. Gardner. “Analysing Your Multivariate Data as Pictorial: A Case for Applying Biplot Methodology.” International Statistical Review, 73 (2005), 365-387.

Little, R.J.A. and Rubin, D.B. *Statistical Analysis with Missing Data*. J. Wiley & Sons, New York (1987).

Ludwig, J. A., and J. F. Reynolds. *Statistical Ecology- a Primer on Methods and Computing.* New York: Wiley-inter science, (1988).

MacQueen, J. B. “Some methods for Classification and Analysis of Multivariate Observations.” *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1, Berkeley, CA: University of California Press (1967), 281-297.

Mardia, K. V., J. T. Kent, and J. M. Bibby. *Multivariate Analysis* (Paperback). London: Academic Press, (2003).

Meng, X.L. Multiple-imputation inferences with uncongenial sources of input (with discussion). *Statistical Science*, 10 (1995), 538-573.

Morgan, B. J. T., and A. P. G. Ray. “Non-uniqueness and inversions in Cluster Analysis.” *Applied Statistics*, 44, no. 1 (1995), 117-134.

Trygve R Solberg, Anna K Sonesson, John A. Woolliams and Theo HE Meuwissen, *Reducing dimensionality for prediction of genome-wide breeding values:* Norwegian University of Life Sciences (2009).

Pyle, D. *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann, (1999).

Shepard, R. N. “Multidimensional Scaling, Tree-Fitting, and Clustering.” Science, 210 no. 4468 (1980), 390-398.

Sibson, R. “Studies in the Robustness of Multidimensional Scaling” *Journal of the Royal Statistical Society (B),* 40 (1978), 234-238.

Richard A. Johnson and Dean w. Wichern. *Applied Multivariate Statistical Analysis* (Sixth Edition). Person Education, Inc. (2007).

Rubin, D.B. Inference and missing data. *Biometrika*, 63 (1976), 581-592.

Rubin, D.B. *Multiple Imputation for Nonresponse in Surveys*. J. Wiley & Sons, New York (1987).

Rubin, D.B. Multiple imputation (with discussion). *Journal of the American Statistical Association*, 91 (1996), 473-489.

Schafer, J.L. *Analysis of Incomplete Multivariate Data*. Chapman & Hall, London (1997).

Schafer, J.L. Multiple imputation: a primer. *Statistical Methods in Medical Research*, in press (1999).

Takane, Y., F. W. Young, and J. De Leeuw. “Non-metric Individual Differences Multidimensional Scaling: Alternating Least Squares with optimal Scaling Features.” *Psycometrika*, 42 (1977), 7-67.

Ward, Jr., J. H. “Hierarchical Grouping to Optimize an Objective Function.” *Journal of the American Statistical Association*, 58 (1963), 236-244.

Westphal, C., and T. Blaxton. *Data Mining Solutions: Methods and Tools for Solving Real World Problems* (Paperback). New York: John Wiley, (1998).

Whitten, I. H., and E. Frank. *Data Mining: Practical Machine Learning Tool and* Techniques (2nd ed.) (Paperback). San Francisco: Morgan Kaufmann (2005).

Young, F. W., and R. M. Hamer. *Multidimensional Scaling: History, Theory, and Applications*. Hillsdale, NJ: Lawrence Erlbaum Associates, Published (1987).

Yu-Fang Pei, Lei Zhang, Jian Li, Hong-Wen Deng. *Analyses and Comparison of Imputation-Based Association Methods*. Xi’an Jiaotong University, Xi’an, People’s Republic of China, Published (2010).

Appendix-A

Software:

"R" is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

http://www.r-project.org/

Simulation methods:

1- rmvnorm() :

Random generation for the multivariate normal (also called Gaussian) distribution, these functions provide information about the multivariate normal distribution with mean equal to mean and covariance matrix sigma.

"rmvnorm": generates random deviates. (Friedrich Leisch).

```
rmvnorm (n, mean, sigma)
```

-`n`: Number of observations

-`mean`: Mean vector

-`sigma`: Covariance matrix

2- The Beta Distribution: rbeta()

Random generation for the Beta distribution with parameters shape1 and shape2 (and optional non-centrality parameter ncp),

```
rbeta(n, shape1, shape2, ncp = 0)
```

`n`: number of observations.

`shape1, shape2`: positive parameters of the Beta distribution.

`ncp`: non-centrality parameter.

The Beta distribution with parameters shape1 = a and shape2 = b has density

`Γ(a+b)/(Γ(a)Γ(b))x^(a-1)(1-x)^(b-1)`

`for a > 0, b > 0 and 0 ≤ x ≤ 1` where the boundary values at x=0 or x=1 are defined as by continuity (as limits).

The mean is a/(a+b)

and the variance is ab/((a+b)^2 (a+b+1)).

3- The sample function: sample()

Sample takes a sample of the specified size from the elements of x using either with or without replacement.

```
sample(x, size, replace = FALSE, prob = NULL)
```

n: a positive number, the number of items to choose from.

*size:*a non-negative integer giving the number of items to choose.

replace: Should sampling be with replacement?

prob: A vector of probability weights for obtaining the elements of the vector being sampled.

Appendix-B

The SAS code for multiple imputation algorithms can be found in Appendix-B (Paul D. Allison, 2005).

```
/*Proportions, single auxiliary covariate, MCAR*/
%let cut=.50;
data dumsim;
cut=&cut;
do sample = 1 to 500;
do i=1 to 500;
d=ranuni(0)<cut;
x=-1+1*d+1*rannor(0);
miss=ranuni(0)<.5;
if miss=1 then dmiss=.; else dmiss=d;
output;
end;
end;
run;
proc corr data=dumsim; var d x miss;run;
proc mi data=dumsim out=outdum noprint;
var x dmiss;
monotone regression(dmiss=x);
by sample;
run;
/*The following macros do the analysis. Macros may be found below */
%complete
%analyze
%round
%discrim
%logistic
/*Proportions, single auxiliary covariate, MAR*/
%let cut=.01;
data dumsim;
misslope=2;
missint=2;
cut=&cut;
do sample = 1 to 500;
do i=1 to 500;
d=ranuni(0)<cut;
x=-1+1*d+.3*rannor(0);
p=1/(1+exp(-missint-misslope*x));
miss=ranuni(0)<p;if miss=1 then dmiss=.; else dmiss=d;
output;
end;
end;
run;
proc corr data=dumsim; var d x miss;run;
proc mi data=dumsim out=outdum noprint;
var x dmiss;
monotone regression(dmiss=x);
by sample;
run;
%complete
%noround
%round
%logistic
%discrim
```

```
%macro complete;
proc means data=dumsim nway noprint;
var dmiss;
class sample;
output out=a mean=mean lclm=lclm uclm=uclm stderr=se;
run;
data b;
set a;
coverage=lclm<&cut<uclm;
run;
proc means data=b;
var mean se coverage;
run;
%mend complete;
%macro noround;
%do i=1 %to 500;
proc reg data=outdum outest=a covout noprint;
where sample=&i;
model dmiss=;
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var intercept;
ods output ParameterEstimates=parms&i;
run;
%end;
data parms;
set %do j=1 %to 500; parms&j %end; ;
coverage=lclmean<&cut<uclmean;
run;
ods listing;
proc means data=parms;
var Estimate stderr coverage;
run;
%mend noround;
%macro round;
data outround;
set outdum;
if dmiss>.5 then dmiss=1; else dmiss=0;
run;
%do i=1 %to 500;
proc reg data=outround outest=a covout noprint;
where sample=&i;
model dmiss=;
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var intercept;
ods output ParameterEstimates=parms&i;
run;
%end;
data parms;
set %do j=1 %to 500; parms&j %end; ;
coverage=lclmean<&cut<uclmean;
run;
ods listing;
proc means data=parms;
var Estimate stderr coverage;
run;
```

```
%mend round;
%macro logistic;
%do i=1 %to 500;
proc mi data=dumsim out=outlog noprint ;
where sample=&i;
class dmiss;
var x dmiss;
monotone logistic(dmiss=x) ;
run;
proc reg data=outlog outest=a covout noprint;
model dmiss=;
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var intercept;
ods output ParameterEstimates=parms&i;
run;
%end;
data parms;
set %do j=1 %to 500; parms&j %end; ;
coverage=lclmean<&cut<uclmean;
run;
ods listing;
proc means data=parms;
var Estimate stderr coverage;
run;
%mend logistic;
%macro discrim;
%do i=1 %to 500;
proc mi data=dumsim out=outlog noprint ;
where sample=&i;
class dmiss;
var x dmiss;
monotone discrim(dmiss=x) ;
run;
proc reg data=outlog outest=a covout noprint;
model dmiss=;
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var intercept;
ods output ParameterEstimates=parms&i;
run;
%end;
data parms;
set %do j=1 %to 500; parms&j %end; ;
coverage=lclmean<&cut<uclmean;
run;
ods listing;
proc means data=parms;
var Estimate stderr coverage;
run;
%mend discrim;
/* Regression with dummy predictor, MAR*/
%let cut=.5;
%let b=1;
%let c=1;
data dumreg;
b=&b;
```

```
c=&c;
missint=1;
misslope=1;
sig=3;
cut=&cut;
cutmiss=.5;
do sample = 1 to 500;
do i=1 to 500;
d=ranuni(0)<cut;
x=-1+1*d+1*rannor(0);
y=b*d+c*x+sig*rannor(0);
p=1/(1+exp(-missint-misslope*x));
miss=ranuni(0)<p;if miss=1 then dmiss=.; else dmiss=d;
output;
end;
end;
run;
proc reg; model y= x d; run;
ODS LISTING;
proc corr data=dumreg;var d x y miss; run;
proc mi data=dumreg out=outreg noprint;
var x y dmiss;
monotone regression(dmiss=x y);
by sample;
run;
/*The following macros do the analysis. Macros may be found below*/
%completereg
%noroundreg
%roundreg
%discrimreg
%logreg
%macro completereg;
ods listing close;
proc reg data=dumreg;
model y=dmiss x / clb;
ods output ParameterEstimates=parms;
by sample;
run;
ods listing;
data dmiss;
set parms;
where variable='dmiss';
coverage=lowercl<&b<uppercl;
run;
proc means data=dmiss;
var Estimate StdErr coverage;
run;
data x;
set parms;
where variable='x';
coverage=lowercl<&c<uppercl;
run;
proc means data=x;
var Estimate StdErr coverage;
run;
%mend completereg;
%macro noroundreg;
%do i=1 %to 500;
proc reg data=outreg outest=a covout noprint;
where sample=&i;
model y=dmiss x;
```

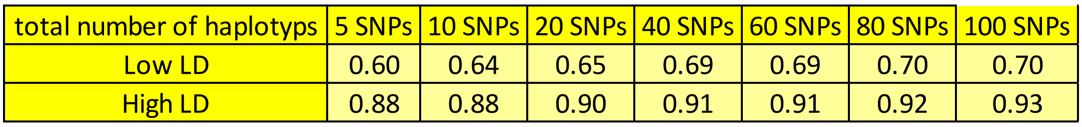
```
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var dmiss x;
ods output ParameterEstimates=parms&i;
run;
%end;
data parmsb;
set %do j=1 %to 500; parms&j %end; ;
where parm='dmiss';
coverage=lclmean<&b<uclmean;
run;
ods listing;
proc means data=parmsb;
var Estimate stderr coverage;
run;
data parmsc;
set %do j=1 %to 500; parms&j %end; ;
where parm='x';
coverage=lclmean<&c<uclmean;
run;
ods listing;
proc means data=parmsc;
var Estimate stderr coverage;
run;
%mend noroundreg;
%macro roundreg;
data outround;
set outreg;
if dmiss>.5 then dmiss=1; else dmiss=0;
run;
%do i=1 %to 500;
proc reg data=outround outest=a covout noprint;
where sample=&i;
model y=dmiss x;
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var dmiss x;
ods output ParameterEstimates=parms&i;
run;
%end;
data parmsb;
set %do j=1 %to 500; parms&j %end; ;
where parm='dmiss';
coverage=lclmean<&b<uclmean;
run;
ods listing;
proc means data=parmsb;
var Estimate stderr coverage;
run;
data parmsc;
set %do j=1 %to 500; parms&j %end; ;
where parm='x';
coverage=lclmean<&c<uclmean;
run;
ods listing;
proc means data=parmsc;
var Estimate stderr coverage;
```

```
run;
%mend roundreg;
%macro logreg;
%do i=1 %to 500;
proc mi data=dumreg out=outlog noprint ;
where sample=&i;
class dmiss;
var x y dmiss;
monotone logistic (dmiss=x y) ;
run;
proc reg data=outlog outest=a covout noprint;
model y=dmiss x;
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var dmiss x;
ods output ParameterEstimates=parms&i;
run;
%end;
data parmsb;
set %do j=1 %to 500; parms&j %end; ;
where parm='dmiss';
coverage=lclmean<&b<uclmean;
run;
ods listing;
proc means data=parmsb;
var Estimate stderr coverage;
run;
data parmsc;
set %do j=1 %to 500; parms&j %end; ;
where parm='x';
coverage=lclmean<&c<uclmean;
run;
ods listing;
proc means data=parmsc;
var Estimate stderr coverage;
run;
%mend logreg;
%macro discrimreg;
%do i=1 %to 500;
proc mi data=dumreg out=outlog noprint ;
where sample=&i;
class dmiss;
var x y dmiss;
monotone discrim (dmiss=x y) ;
run;
proc reg data=outlog outest=a covout noprint;
model y=dmiss x;
by _imputation_;
run;
ods listing close;
proc mianalyze data=a;
var dmiss x;
ods output ParameterEstimates=parms&i;
run;
%end;
data parmsb;
set %do j=1 %to 500; parms&j %end; ;
where parm='dmiss';
coverage=lclmean<&b<uclmean;
```

```
run;
ods listing;
proc means data=parmsb;
var Estimate stderr coverage;
run;
data parmsc;
set %do j=1 %to 500; parms&j %end; ;
where parm='x';
coverage=lclmean<&c<uclmean;
run;
ods listing;
proc means data=parmsc;
var Estimate stderr coverage;
run;
%mend discrimreg;
```

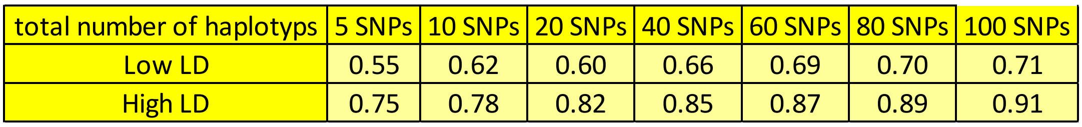
Appendix-C

Table 4.1: The effects of using Low and High linkage disequilibrium dataset on Accuracy rate of LDA in imputation.



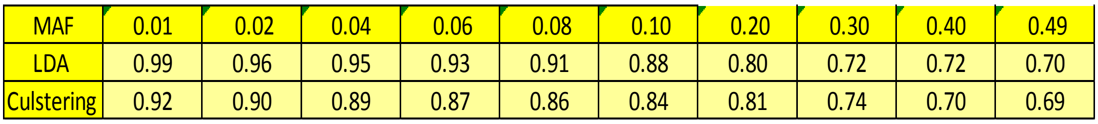
total number of haplotyps	5 SNPs	10 SNPs	20 SNPs	40 SNPs	60 SNPs	80 SNPs	100 SNPs
Low LD	0.60	0.64	0.65	0.69	0.69	0.70	0.70
High LD	0.88	0.88	0.90	0.91	0.91	0.92	0.93

Table 4.2: The effects of no.SNPs using Low and High linkage disequilibrium dataset on accuracy rate of Clustering in imputation.



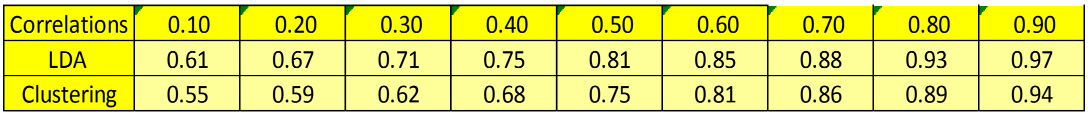
total number of haplotyps	5 SNPs	10 SNPs	20 SNPs	40 SNPs	60 SNPs	80 SNPs	100 SNPs
Low LD	0.55	0.62	0.60	0.66	0.69	0.70	0.71
High LD	0.75	0.78	0.82	0.85	0.87	0.89	0.91

Table 4.3: The effects of Minor allele frequency on accuracy rate using LDA and Clustering.



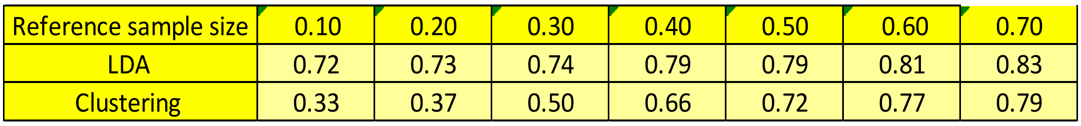
MAF	0.01	0.02	0.04	0.06	0.08	0.10	0.20	0.30	0.40	0.49
LDA	0.99	0.96	0.95	0.93	0.91	0.88	0.80	0.72	0.72	0.70
Culstering	0.92	0.90	0.89	0.87	0.86	0.84	0.81	0.74	0.70	0.69

Table 4.4: The effects of marker density on accuracy rate using LDA and Clustering.



Correlations	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
LDA	0.61	0.67	0.71	0.75	0.81	0.85	0.88	0.93	0.97
Clustering	0.55	0.59	0.62	0.68	0.75	0.81	0.86	0.89	0.94

Table 4.5: The effects of reference sample size on accuracy rate using LDA and Clustering.



Reference sample size	0.10	0.20	0.30	0.40	0.50	0.60	0.70
LDA	0.72	0.73	0.74	0.79	0.79	0.81	0.83
Clustering	0.33	0.37	0.50	0.66	0.72	0.77	0.79