NNT : 2015SACLE013

UNIVERSITÉ PARIS-SACLAY
UNIVERSITÉ D'EVRY VAL D'ESSONNE
STRUCTURE ET DYNAMIQUE DES SYSTÈMES VIVANTS

Doctoral Thesis

presented by

Alia DEHMAN

entitled

Spatial Clustering of Linkage Disequilibrium blocks for Genome-Wide Association Studies

in fulfilment of the requirements
for the degree of Doctor

*in **Life sciences and health***
(Doctoral School n^o577)

Defended on December, 9th 2015 in front of

Reviewers	David Causeur	Agrocampus Ouest
	Franck Picard	UMR CNRS 5558
Examiners	Maria Martinez	INSERM U1043
	Tristan Mary-Huard	AgroParisTech INRA UMR de Génétique Végétale
Supervisors	Christophe Ambroise	Université d'Evry Val d'Essonne
	Pierre Neuvial	UMR CNRS 8071

In memory of my uncle Borg . . .

Remerciements

Mes premiers remerciements vont à mes deux directeurs Christophe Ambroise et Pierre Neuvial pour avoir encadré et soutenu mes travaux de thèse durant ces trois années. Christophe, c'est d'abord à toi que je dois d'être là. Merci d'avoir cru en moi et de m'avoir fait découvrir la biostatistique, un champ des mathématiques appliquées dont j'ignorais l'existence mais qui m'a convaincue de son importance dans la recherche médicale. Pierre, merci d'avoir suivi de si près tout mon travail pendant ces trois années et d'avoir partagé avec moi ta grande culture scientifique et tes connaissances en programmation. À tous les deux, merci pour votre grande disponibilité, votre patience et vos encouragements.

Merci aux Professeurs Franck Picard et David Causeur, pour m'avoir fait l'honneur d'évaluer ma thèse. Je vous remercie tous les deux pour le temps que vous avez consacré à mon travail et pour les commentaires enrichissants et précieux pour la suite. Merci à Maria Martinez et Tristan Mary-Huard d'avoir si aimablement et rapidement accepté de faire partie de mon jury.

Je tiens à remercier tous les membres de l'équipe Statistique et Génome. Quand je suis arrivée au laboratoire pour mon stage de Master 2, j'y ai été extrêmement bien accueillie par vous tous, Michèle, Margot, Cyril, Carène, Yolande, Catherine, Maurice, Franck, Julien, Marie-Luce, Anne-Sophie, Abass, ... Un grand merci à mes collègues et amis docteurs et futurs docteurs Sarah, Quentin, Morgane, Jean-Michel, Virginie et Benjamin. Une pensée particulière au Professeur Émérite Bernard Prum qui a été l'un des premiers à me faire confiance.

Mes derniers remerciements et non les moindres vont à ma famille. Merci d'abord à mes oncles et tantes qui ont toujours été fiers de leur nièce et qui ne s'en sont jamais cachés : merci de croire autant en moi. Une pensée particulière à mon cher oncle Fayçal parti trop tôt. Il a été le premier à me soutenir dans mon projet de faire des études de statistiques. Un grand merci à ma belle famille, tonton Naceur, tata Achraf et Imenouch, qui prennent tant soin de moi. Lallou, ma soeurette, quelle chance précieuse que tu sois là ! Je ne pourrai jamais assez te dire à quel point ton soutien compte pour moi. Merci pour tous nos bons moments à deux et nos conversations interminables au téléphone. Mes plus profonds remerciements sont pour mes chers parents Amani et Apati. Ma reconnaissance pour tout ce que vous m'apportez va bien au-delà de ces quelques lignes et je ne saurais exprimer tout ce que je vous dois. Ammouni, merci pour tout le temps que tu me consacres et merci pour tous les bons petits plats que tu me prépares pour ma semaine lorsque je viens à Lyon pour le week end. Merci de prendre autant soin de moi. Appouti, merci de pouvoir toujours compter sur toi, merci pour tes conseils avisés et surtout merci de m'avoir redonné confiance en moi à chaque moment de doute. Merci pour cette chance inestimable de pouvoir compter sur vous deux ! Enfin, un grand merci à mon merveilleux compagnon et mari. Mehdouch, merci pour ton écoute, ta compréhension, ton

soutien inconditionnel et surtout ta patience pendant ces trois années. Merci pour ton amour qui me donne des ailes !

Contents

List of Figures

List of Tables

Abbreviations

AHC	Agglomerative Hierarchical Clustering
AUC	Area Under the Curve
BALD	Blockwise Approach using Linkage Disequilibrium
DNA	Desoxyribo Nucleic Acid
DSL	Disease Susceptibility Loci
FDR	False Discovery Rate
FN	False Negative
FP	False Positive
FWER	Family-Wise Error Rate
GLM	Generalized Linear Models
GWAS	Genome-Wide Association Studies
HIV	Human Immunodeficiency Virus
HMM	Hidden Markov Models
htSNP	haplotype tag Single Nucleotide Polymorphism
HWE	Hardy-Weinberg Equilibrium
Lasso	Least Absolute Shrinkage and Selection Operator
LD	Linkage Disequilibrium
MAF	Minor Allele Frequency
MATILDE	MCMC Algorithm To Identify blocks of Linkage DisEquilibrium
MDL	Minimum Description Length
OLS	Ordinary Least Squares
PDK	Positive Definite Kernel
SMA	Single-Marker Analysis
SNP	Single Nucleotide Polymorphism
STR	Short Tandem Repeats

SVM	Support Vector Machines
TN	True Negative
TP	True Positive
tSNP	tag Single Nucleotide Polymorphism

Chapter 1

Introduction

1.1 General background

It is now known that many diseases have a genetic component (Rousseau & Laflamme 2003). A Desoxyribo Nucleic Acid (DNA) sequence may be represented as a sequence of letters $\{A$, T, G and $C\}$ called bases. In a given position on a chromosome or locus, we can find different versions of the genetic text or alleles. In the simplest case, the change of one allele alone at a given locus can be responsible for a disease: it is called monogenic disease; in a less trivial situation, the disease called multifactorial or complex disease is the result of multiple genetic and environmental components, which is the case with most cancers, psychiatric and autoimmune diseases (Hunter 2005).

Single Nucleotide Polymorphisms (SNPs) are markers constituting a rich and abundant source of genetic information. Defined as positions in the chromosome where the genetic text varies by a single base from one individual to another, they constitute 90% of all human variation (Crews et al. 2012). The decrease in both cost and time of SNP genotyping recently helped to launch large-scale genetic association studies, also called Genome-Wide Association Studies (GWAS), in order to explore a significant part of genetic polymorphisms that may be involved in the biological mechanisms behind diseases (Risch 2000). One popular GWAS approach is to collect, according to certain criteria, a sample of individuals suffering from the disease called cases and non-affected individuals called controls and to determine the positions in the genome where the genetic text differs significantly between cases and controls. This is called a case-control association study.

Despite having identified hundreds of genetic variants responsible for a variety of simple/complex diseases and other phenotypic traits, more must be done to reach the goal of identifying major disease risks and understanding disease mechanism. Indeed, the genetic variants identified to date have mostly weak effects on the disease risk and explain only a small proportion (reaching 20% to 25% in some well-studied cases) of the *total heritability* (Lander 2011). Geneticists define the proportion of heritability of a trait explained by a set of known genetic variants to be the proportion of the phenotypic variance explained by the additive effects of known variants divided by he proportion of the phenotypic variance attributable to the additive effects of all variants, including those not yet discovered (Zuk et al. 2012). The relatively poor performance of GWAS in explaining heritability can be attributed to fundamental limitations in the GWAS methodology. First, structural changes such as variations in copy number (CNV) or epigenetics (which are variations that are caused by external or environmental factors that control genes) are not handled by the GWA strategy. Indeed, the technologies used in whole genome (including sequencing) can not detect this type of variation. Techniques are being developed to detect large-scale epigenetic variants through dedicated technologies, and Epigenome-Wide Association Studies (EWAS) are being conducted (Rakyan et al. 2011). Second, the observed missing heriability can be due to the fact that genetic markers do not act independently but rather exhibit *epistasis* : a phenomenon in which the effect of a genetic variant can either be masked or increased by one or more other markers. Detecting epistasis from GWAS is fundamentally difficult because of the large number of SNPs and relatively small number of samples. And computationally speaking, researchers have to face the high-dimensionality issue, whereas the search space of the problem grows exponentially with the number of involved markers. Third, rare variants of large effect can also play an important part in common diseases. Studying them requires sequencing genomic regions to identify those in which the aggregate frequency of rare variants is higher in cases than controls (Huang et al. 2015, Consortium et al. 2015). Furthermore, given the background rate of rare variants (around 1% per gene), many thousands of samples will be needed to achieve statistical significance. Although the inferred effect sizes are larger, the overall contribution to the heritability may still be small due to their low frequency (Lander 2011). Finally, a part of the missing heritability may be due to an underutilisation of the large amount of GWAS data, and the incorporation of biological information on genotypic data may help explaining a part of the missing phenotypic variance. Thus, the development of methods in statistical genetics which take into account biological infomation is very important in order to improve these studies and provide a proper interpretation of their findings. Moreover,

the need to furnish software and algorithmic tools that allow to apply such methods and that are adapted to the high-dimensionality of GWAS data, has become stronger.

The work carried out in the framework of this thesis emerged from the will to enhance the power of statistical methods routinely used in GWAS by incorporating linkage disequilibrium information in the marker selection model.

Single-marker tests have been widely used in GWAS for detecting marker associations (Burton et al. 2007, Sham & Purcell 2014). This approach assumes that SNPs are independent of one another, ignoring the important correlation structure due to LD, and remains then unsatisfactory for many reasons. From a biological point of view, it is possible that the observed data only contain SNPs that are in LD with causal ones since the causal marker may not be genotyped. From a statistical perspective, the distinction between SNP-level and LD block-level associations is related to an identifiability issue: assuming that causal SNPs are observed, is their association to the phenotype strong enough so that they can be distinguished from indirect associations between SNPs in strong LD with causal ones?

In this manuscript, we propose an approach of marker selection that explicitly takes into account the block structure induced by LD among the genetic data in order to detect groups of SNPs that are associated with the phenotype. In a second contribution, the proposed method is scaled to high-dimensional data by means of an optimized implementation of one of its steps. This work is also accompanied by an important software development with the provision of the implementation to the scientific community.

1.2 Manuscript overview

This manuscript is organized in four main chapters. Each chapter ends with a summary and discussion of the results presented in that chapter.

The first introductory chapter starts with the presentation of statistical and genetic notions that are necessary for the proper understanding of the remainder of the manuscript. After an initiation to hypothesis testing and multivariate linear models, several genetic precepts around the genome and genotypes are introduced. We then introduce the linkage disequilibrium, its most popular measures and two approaches to estimate it. Finally, within the framework of Genome-wide

association studies, the concepts of epidemiology of complex diseases, high-throughput genotyping and single-marker analyses are presented. This first chapter ends with an introduction to haplotype association analyses.

The second chapter of this manuscript focuses on the issue of detecting linkage disequilibrium blocks. To this end, many block partitioning approaches have been proposed that differ according to wether they use haplotypes or pairwise LD measures for defining the blocks. We first introduce and discuss these methods and then present a novel approach that we developed during this PhD. More specifically, we propose to infer the linkage disequilibrium structure by means of an adjacency-constrained hierarchical clustering according to the Ward's criterion and using LD similarity, followed by the Gap statistic model selection approach in order to estimate the optimal number of blocks.

The third chapter is dedicated to an application of the LD block partitioning approach for variable selection in Genome-wide association studies. In order to improve the power of these studies by detecting associated markers which may have been missed by the single-marker analysis, we propose to take advantage of the strong dependency structure between nearby SNPs, induced by LD, and to explicitly look for sets of LD blocks jointly associated to the phenotype of interest. To this end, we present the Blockwise Approach using Linkage Disequilibrium (BALD) which consists in inferring the LD blocks using the two steps described in the previous chapter and then identifying associated groups of SNPs using the Group Lasso regression model. The efficiency of this approach is then investigated by comparing it to state-of-the-art regression methods on simulated, semi-simulated and real data. The R package `BALD` used to apply this approach is available at the website of the laboratory `http://www.math-evry.cnrs.fr/logiciels/bald`.

The fourth chapter of this manuscript focuses on the issue of scaling the BALD approach, and more particularly its adjacency-constrained hierarchical clustering step, to high-dimensional GWAS data. We then propose an efficient implementation of such an algorithm in the general context of any similarity measure, not necessarily the LD similarity. This implementation requires a user-defined parameter h which controls the maximum lag between items for similarity calculations, resulting in a h-band similarity matrix. By means of a simple expression of the Ward's criterion and the usage of a min-heap structure, we reduce the time and space complexities of the adjacency-constrained hierarchical clustering algorithm. The interest of this

novel implementation is illustrated in GWAS applications, where h is several orders of magnitude smaller than the number of SNPs to be clustered.This improved implementation is also integrated to the R package `BALD`.

Finally, the last chapter corresponds to a conclusion of this manuscript. This chapter summarizes all the contributions made in the framework of this research work but also opens on different scientific perspectives related to possible improvements in these contributions.

Chapter 2

Context

2.1 Statistical precepts

2.1.1 Hypothesis testing

2.1.1.1 Simple hypothesis testing

Faced with complex and random events, decision making is difficult and the tools of the hypothesis testing theory are intended to guide the choice between different alternatives where there is not necessarily a "good answer". The decision always includes an error probability and the goal is then to minimize the number of errors. In general, it is to decide whether observed differences between an existing model and observations are real or can be considered to be due to mere chance of sampling. Such a decision may often be translated according to a parameter of the distribution of the observed data: the null hypothesis H_0 consists in assuming that this value is true and the alternative hypothesis H_1 is generally the complementary of H_0.

Formally speaking, a testing procedure follows a defined sequence of steps that consist in:

i. Stating a null hypothesis (H_0) and an alternative hypothesis (H_1);

ii. Calculating a random variable of decision, the test statistic ($\mathcal{S}$), which corresponds to a function of the observations. It measures a distance between what we observe and what we expect under the null hypothesis. The greater this distance, the less likely the null hypothesis H_0;

iii. Computing the statistical confidence measure called p-*value* which corresponds to the probability to obtain an observed statistic ($\mathcal{S}^{obs}$) higher/lower (depending on the form of the null hypothesis) than the obtained value if H_0 were true.

iv. Draw conclusions in function of the value of the p-value. A small p-value indicates that there is a significant difference between the observed statistic $\mathcal{S}^{obs}$ and the expected one under the null hypothesis, which suggests that the null hypothesis does not accurately describe the observed data. The null hypothesis is therefore rejected. Conversely, a high p-value means that the observation is not sufficiently inconsistent with the null hypothesis for it to be rejected.

In practice, making the decision of accepting or rejecting H_0 requires comparing its corresponding p-value to a confidence threshold termed *level of the test*, usually noted α.

Given a null hypothesis H_0, four outcomes are possible depending on wether the null hypothesis is true or false and wether the statistical test rejects or not the null hypothesis. The challenge is therefore to know how to take the right decision by controlling a given risk of being wrong.
More precisely, there are two ways of taking a wrong decision and therefore two types of risks: the statistical test can reject H_0 where H_0 is true. This type of error is called *type-I error* and the associated risk the *type-I error rate* is noted α. If the procedure does not reject H_0 when H_1 is true, then it commits a *type-II error* with an *type-II error rate* noted β. The true state and the possible decisions of the procedure are summarized in Table 2.1.

	H_0 is not rejected	**H_0 is rejected**
H_0 is true	true negative ; $(1-\alpha)$	false positive ; type-I error rate (α)
H_0 is false	false negative ; type-II error rate (β)	true positive ; power $(1-\beta)$

TABLE 2.1: Outcomes of a statistical test and associated risks.

Defined as above, the confidence threshold α corresponds to the false positive rate:

$$\alpha = \mathbb{P}\{ \text{ the test rejects } H_0 \text{ while } H_0 \text{ is true } \}.$$

2.1.1.2 Multiple hypothesis testing

A multiple hypothesis problem occurs when several ($m > 1$) tests are carried out simultaneously. m null hypotheses $H_0^1, H_0^2, \ldots, H_0^m$ are then considered at the same time.

When m statistical tests are performed simultaneously, depending on wether each hypothesis tested is true or false and the statistical test accepts or rejects the null hypothesis, each of the m results will fall in one of the four outcomes defined in Table 2.1. The equivalent Table 2.2, corresponding to the multiple testing procedure, indicates the actual number of false positives (FP) and false negatives (FN) instead of their respectives rates (α and β).

	H_0 is not rejected	**H_0 is rejected**	**Total**
H_0 is true	true negatives (TN)	false positives (FP)	$m_0 = TN + FP$
H_0 is false	false negatives (FN)	true positives (TP)	$m_1 = FN + TP$
Total	$m_U = TN + FN$	$m_R = FP + TP$	m

TABLE 2.2: Outcomes of m statistical tests.

The risk of making a type-I error increases with the number of hypotheses tested. For instance, if we consider a simultaneous hypothesis testing procedure of $m = 10000$ hypotheses, by choosing $\alpha = 0.05$ for each test, we expect 500 hypotheses to be rejected by simple chance. This may not be acceptable when one wishes to highlight a phenomenon involving some hundreds of variables. Consequently, it is necessary to consider alternative confidence measures instead of the false-positive rate. More relevant measures (Shaffer 1995, Dudoit et al. 2003, Hochberg & Tamhane 2009) such as the Family-Wise Error Rate and the False Discovery Rate were developed for this purpose.

Family-Wise Error Rate The first alternative confidence measure proposed to handle the multiple-testing issue is the Family-Wise Error Rate (FWER) criterion. It is defined as the probability of falsely rejecting at least one null hypothesis among the family of hypotheses considered:

$$\text{FWER} = \mathbb{P}(FP \geq 1).$$

Thus, controlling the FWER at a given level corresponds to the control of the probability of having at least one false positive.

Several procedures have been developed in order to control the FWER (Bonferroni 1936, Šidák 1967). The simplest and most widely used approach is to apply the Bonferroni procedure (Bonferroni 1936) which accounts for the number m of tests performed. If we note p_i the p-value of a test $i \in \{1 \ldots, m\}$, then the p-value obtained by the Bonferroni procedure equals to

$p_i^{\text{Bonf}} = mp_i$. Therefore, controlling the FWER at a level of 5% requires to apply a threshold of 5% to the new p_i^{Bonf} p-values corresponding to the product of each p-value with the number of tests m. For example, to ensure that the FWER is not greater than 5% when performing $m = 1000$ tests, each test is considered as significant only if its p-value is less than $0.05/1000 = 5 \times 10^{-5}$.

Dealing with the multiple-testing issue by controlling the FWER is simple and straightforward using the Bonferroni procedure, which is valid for any dependence structure of the hypotheses tested. However, the control of the FWER in itself is not ideal when the number of tests if very large. It leads to too stringent adjusted p-values and therefore many missed findings while preventing against any single false-positive.

False Discovery Rate In order to overcome the FWER limitations, Benjamini & Hochberg (1995) introduced the False Discovery Rate (FDR). The FDR approach focuses on the expected proportion of true null hypotheses that are falsely rejected. More formally, using the notations of Table 2.2, the FDR is defined as:

$$\text{FWER} = \begin{cases} 0 & \text{if } m_R = 0 \\ \mathbb{E}\left[\frac{FP}{FP+TP}\right] = \mathbb{E}\left[\frac{FP}{m_R}\right] & \text{otherwise.} \end{cases}$$

In order to control the FDR at a level α, the Benjamini-Hochberg procedure consists in first listing the p-values in increasing order $p_{(1)} \leq p_{(2)}, \leq \cdots \leq p_{(m)}$. Then, the number $i^\star$ of the test verifying:

$$i^\star = \max_i \left(p_{(i)} \leq \frac{i}{m}\alpha \right)$$

is determined. If $i^\star$ does not exist then no hypothesis is rejected. Otherwise, all the hypotheses H_0^j with $p_j \leq p_{i^\star}$ are rejected (see Figure 2.1). We can show that this procedure is equivalent to applying the threshold α to the p-values $p_i^{\text{BH}} = \frac{m \times p_i}{m_{R_i}}$ with m_{R_i} the number of tests with p-values smaller than p_i.

Finally, the Benjamini-Hochberg procedure controls the FDR only in families with independent or positively dependent test statistics (Benjamini & Hochberg 1995, Benjamini & Yekutieli 2001).

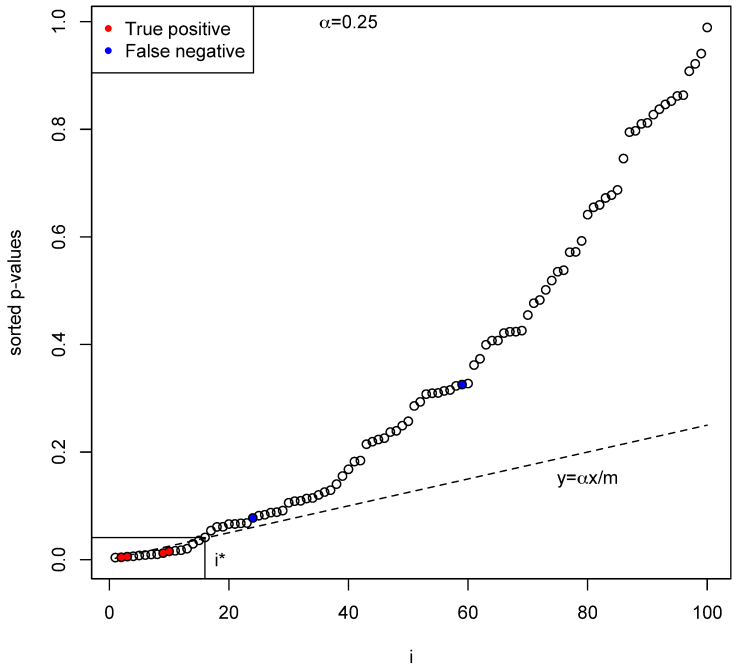


FIGURE 2.1: Application of Benjamini & Hochberg (1995) procedure for FDR control. The ranked p-values were simulated from an example with 100 tests of which 6 were true alternative hypotheses. The significance threshold α was set to 0.25.

2.1.2 Multivariate linear models in high-dimension

Finding out relationships between explanatory variables (or predictors) and observation (or response) is a major issue in many fields. When dealing with increasingly large amounts of data, we are often seeking to determine a small subset of variables that explain the observation and that also allow to predict it. In this manuscript, we will be considering the specific case of a linear combination between the variables and the response using the *multivariate linear model*.

2.1.2.1 The multivariate linear model

The multivariate linear model is defined as:

$$y_i = \beta_0 + \sum_{j=1}^{p} \mathbf{X}_{ij}\beta_j + \varepsilon_i, \ i = 1, \ldots, n.$$

where y_i is the response to be explained by the vector of variables $\mathbf{X}_{i.} = (\mathbf{X}_{i1}, \mathbf{X}_{i2}, \ldots, \mathbf{X}_{ip})$. The scalar β_0, called *intercept*, and the coefficients $(\beta_j)_{j=1,\ldots,p}$ are the parameters of the model

to be estimated. The quantities ε_i are random variables and correspond to the residuals of the model that we assume to be gaussian, that is $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

If we note $\mathbf{y}$ the vector of n observations of the response and add a column of 1 at the left of the matrix $\mathbf{X}$, then the multivariate linear model can be written as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\beta} = {}^t(\beta_0, \beta_1, \ldots, \beta_p)$ is the vector of parameters to be estimated and $\boldsymbol{\varepsilon}$ the vector of residuals that we assume independent and identically distributed.

The classical estimator of a linear model is the *Ordinary Least Squares* (OLS) estimator which is defined as the vector minimizing the Residual Sum of Squares (RSS):

$$\hat{\boldsymbol{\beta}}^{\text{OLS}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min}\, RSS(\boldsymbol{\beta}), \text{ with } RSS(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2.$$

$\|.\|_2$ denotes the Euclidean norm.

When ${}^{\mathbf{t}}\mathbf{X}\mathbf{X}$ is invertible, the solution $\hat{\boldsymbol{\beta}}^{\text{OLS}}$ is unique and is called the *Markov-Gauss estimator*:

$$\hat{\boldsymbol{\beta}}^{\text{OLS}} = ({}^{\mathbf{t}}\mathbf{X}\mathbf{X})^{-1\,\mathbf{t}}\mathbf{X}\mathbf{y}.$$

It is an unbiased estimator of $\boldsymbol{\beta}$ with a covariance matrix verifying:

$$Var(\hat{\boldsymbol{\beta}}^{\text{OLS}}) = \sigma^2({}^{\mathbf{t}}\mathbf{X}\mathbf{X})^{-1}.$$

2.1.2.2 The high-dimensionality issue

The OLS estimator presented above remains unsatisfactory for many reasons such as:

- the accuracy of the estimate: the least squares estimator is unbiased but still has a substantial variance. Thus, the prediction accuracy can be improved by reducing some coefficients and setting to zero others in order to reduce the variance of the predicted values.

- the interpretation of the model: we often need to select a subset of explanatory variables in order to highlight the most important effects on the response. In other words, the aim is an estimator which is *parsimonious*, that is with many coefficients exactly equal to 0 and therefore a subset of non-zero coefficients that is small in comparison with the initial number of variables p.
- this estimator is only defined when the number of variables p is less than the number of observations n. Indeed, otherwise, the matrix ${}^{\mathbf{t}}\mathbf{XX}$ is at most of rank n and is therefore not invertible.

This last argument is particularly relevant as it reflects an issue facing various research fields nowadays: *high-dimensional data*. High-dimensional data are data where the number of variables p is greater or even much greater than the number of observations n. Since high-dimensional problems remain insoluble with the classical analyses such as the OLS estimator, alternative models, such as *penalized regression models*, have been proposed in order to deal with these issues.

2.1.2.3 Penalized regression models for high-dimensional data

Penalization can be seen as integrating a prior knowledge of the solution through a regularization term or penalty. One possible approach of penalization is to estimate the vector of parameters $\boldsymbol{\beta}$ using the criterion:

$$\hat{\boldsymbol{\beta}}^{\text{pen}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|, \tag{2.1}$$

where $\|\boldsymbol{\beta}\|$ denotes a norm of $\boldsymbol{\beta}$ and λ a regularization (or penalization) parameter. The optimal value of λ is that which minimizes the prediction error of the estimator $\hat{\boldsymbol{\beta}}^{\text{pen}}$ and can be estimated using algorithms such as *cross-validation* (Arlot et al. 2010).

Several penalized regression models, differing by the norm considered, have been proposed. Below are the most commonly used regularized linear models as well as their advantages and drawbacks.

Ridge. When the explanatory variables are highly correlated, the ${}^{\mathbf{t}}\mathbf{XX}$ matrix involved in the calculation of the least squares estimator has one or more of its eigenvalues close to 0. As

a result, the $\hat{\boldsymbol{\beta}}^{\text{OLS}}$ coefficients are likely to take disproportionately high values. In order to overcome this limitation, the Ridge regression model has been proposed:

$$\hat{\boldsymbol{\beta}}^{\text{Ridge}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2. \tag{2.2}$$

The analytical solution of this equation leads to the expression:

$$\hat{\boldsymbol{\beta}}^{\text{Ridge}} = ({}^{\mathbf{t}}\mathbf{X}\mathbf{X} + \lambda\mathbf{I}_p)^{-1\,\mathbf{t}}\mathbf{X}\mathbf{y},$$

where $\mathbf{I}_p$ is the $p \times p$ identity matrix. The set of eigenvalues of ${}^{\mathbf{t}}\mathbf{X}\mathbf{X}$, including the smaller ones that reflect the correlations, are therefore offset by λ. Imposing such a constraint to the eigenvalues of ${}^{\mathbf{t}}\mathbf{X}\mathbf{X}$ allows to control the magnitude of the Ridge's coefficients, and then to reduce the variance of the estimator, which can improve its prediction performance.

Lasso. As λ increases, the Ridge's coefficients approach but do not equal zero. Hence, no variable is ever excluded from the model. In contrast, the use of an ℓ_1-penalty does reduce terms to zero. This yields the Lasso model. Introduced by Tibshirani (1996), the Least Absolute Shrinkage and Selection Operator (Lasso) estimator is written as follows:

$$\hat{\boldsymbol{\beta}}^{\text{Lasso}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2.3}$$

$$\text{with} \qquad \|\boldsymbol{\beta}\|_1 = \sum_{j=0}^{p} |\beta_j|.$$

The Lasso therefore uses a penalty on the ℓ_1-norm of the estimator's coefficients. The λ parameter controls the sparsity of the model, so that if $\lambda \to \infty$, no predictor is selected. And for a relatively small value of λ, all the variables are included in the model, that is they all have a non-zero coefficient.

To better understand the role of the penalization and the functioning of the Lasso approach, let us consider the case $n > p$. And let us look more closely at the form of the Lasso estimator in the simple case where the variables are independent, that is if the matrix $\mathbf{X}$ is orthogonal. In that case, the OLS estimator is well defined and for $j \in [1, p]$, the Lasso estimator is of the form:

$$\hat{\beta}_j^{\text{Lasso}} = \text{sign}(\beta_j^{\text{OLS}})(|\beta_j^{\text{OLS}}| - \lambda)_+.$$

This equation shows that the coefficients β_j^{OLS} of the least squares estimator are thresholded. They are shrunk by a factor λ if they are greater than λ and zeroed otherwise. Again, this example illustrates the role of the parameter λ which determines somehow the sparsity of the model: the greater the value of λ, the more zero coefficients in the Lasso estimator.

Adaptive Lasso. Introduced by Zou (2006), the Adaptive Lasso is a weighted version of the Lasso estimator defined as:

$$\hat{\boldsymbol{\beta}}^{\text{AdapL}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda \sum_{j=1}^{p} \hat{w}_j |\beta_j|,$$

where

$$w_j = \frac{1}{|\widetilde{\beta}_j|^\gamma},$$

$\gamma > 0$ and $\widetilde{\beta}_j$ is an initial non-zero $\sqrt{n}$-consistent estimate for β_j, that is it converges in probability to the true vector of parameters with a convergence rate of $\sqrt{n}$. If the initial estimator $\widetilde{\boldsymbol{\beta}}$ is zero-consistent in the sense that, as the sample size increases, estimators of zero coefficients converge to zero in probability and estimators of non-zero coefficients do not converge to zero, then the adaptive weights for the zero coefficients converge to infinity, while the adaptive weights for the non-zero coefficients are bounded. The adaptive Lasso allows then to obtain unbiased (in asymptotic sense) estimates for significant coefficients and, at the same time, reduce to zero the estimates of nuisance variables. In addition, the adaptive Lasso still allows continuous subset selection property of the Lasso.

2.1.2.4 Penalized regression models for structured data (with unknown groups)

The Lasso and Adaptive Lasso models do not incorporate any information on correlation structure between the explanatory variables. More specifically, the Lasso model tends to select at random only one variable in each group of correlated variables. In order to overcome this limitation, other methods have been proposed.

Elastic-Net. Zou & Hastie (2005) propose the Elastic-Net estimator:

$$\hat{\boldsymbol{\beta}}^{\text{EN}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2^2 \tag{2.4}$$

where λ_1 and λ_2 are two regularization parameters. Like the Lasso, the Elastic-Net simultaneously performs automatic variable selection and continuous shrinkage. Unlike the Lasso, the Elastic-Net includes a ridge (ℓ_2) penalty which tends to select groups of correlated variables. It also allows selecting more than n explanatory variables, while Lasso does not.

Fused Lasso. In some data sets, explanatory variables can have a natural order and the use of such information can help to better interpret regression results. To this end, the Fused Lasso model has been introduced by Tibshirani et al. (2005). It is written as follows:

$$\hat{\boldsymbol{\beta}}^{\text{FusedL}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\mathbf{D}\boldsymbol{\beta}\|_1, \tag{2.5}$$

where

$$\mathbf{D}\colon \mathbb{R}^p \to \mathbb{R}^{p-1}$$
$$^t(\beta_1, \beta_2, \ldots, \beta_p) \mapsto {}^t([\beta_2 - \beta_1], \ldots, [\beta_p - \beta_{p-1}])$$

and λ_1 and λ_2 are two regularization parameters. Like the Elastic-Net estimator, the Fused Lasso model differs from the Lasso regression model by its second penalty term which allows to select the relevant variables even when they are highly correlated. Indeed, this fusion term (Land & Friedman 1996) is designed to make successive coefficients as close as possible to each other. Unlike the Elastic-Net estimator, the Fused Lasso model is more efficient in situations where the explanatory variables have a natural ordered correlation structure (grouping structure) between them. Indeed, the first penalty term encourages sparsity in the estimated coefficients; while the second constraint term encourages sparsity in their differences leading to a flatness of the coefficients profiles as a function of j. The theoretical results of (Rinaldo et al. 2009) relative to this estimator confirm the importance of the "ordered" correlation structure of the relevant variables. One difficulty in using the Fused Lasso is computational speed. As shown in Tibshirani et al. (2005), speed becomes a practical limitation for problems with dimensions $p > 2000$ and $N > 200$.

Smooth-Lasso. Introduced by Hebiri (2008), the Smooth-Lasso estimator is inspired by the Fused Lasso model :

$$\hat{\boldsymbol{\beta}}^{\text{SmoothL}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\sum_{j=2}^{p}(\beta_j - \beta_{j-1})^2, \tag{2.6}$$

where λ_1 and λ_2 are two regularization parameters. Indeed, both Smooth-Lasso and Fused Lasso combine a ℓ_1 penalty with a fusion term. The main difference between the two models is that the Smooth-Lasso uses the ℓ_2 distance between successives coefficients whereas the Fused Lasso uses the ℓ_1 distance between them. The strict convexity of the ℓ_2 distance allows the Smooth-Lasso model an easier estimation of the regression coefficients leading to a large reduction of computational cost. At the same time, the sparsity of the resulting model is still ensured by the ℓ_1 penalty. In practice, Hebiri (2008) shows that when the explanatory variables have a natural ordered correlation structure (grouping structure) between them, the Smooth-Lasso then offers better performance selection than the Fused Lasso and Elastic-Net models. This is particularly true for variables located within groups, in terms of indices. However, the performance of the Smooth-Lasso is slightly degraded when selecting variables located at the borders of blocks.

2.1.2.5 Penalized regression models for structured data (with known groups)

Some kinds of data have a known correlation structure among the explanatory variables and in some cases, it is appropriate to select or drop a group of correlated variables simultaneously. In the remainder of this section, we will suppose the p predictors are divided into G groups of sizes $p_1, \ldots, p_G$.

Group Lasso. Yuan & Lin (2005) propose the Group Lasso model to handle groups of predictors. The Group Lasso estimator may then be written as:

$$\hat{\boldsymbol{\beta}}^{\text{GL}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\sum_{g=1}^{G}\sqrt{p_g}\|\boldsymbol{\beta}^g\|_2, \tag{2.7}$$

where $\boldsymbol{\beta}^g$ denotes the p_g-dimensional vector of regression coefficients corresponding to the g^{th} group, so that ${}^t\boldsymbol{\beta} = ({}^t\boldsymbol{\beta}^1, \ldots, {}^t\boldsymbol{\beta}^G)$.

The properties of the ℓ_1-norm are used to identify relevant groups and discarding others, while the ℓ_2-norm involves all the variables of a relevant group uniformly. Hence, by construction, the Group Lasso coefficients within a group tend to be either all zero or all nonzero.

Smoothed Group Lasso. A generalized version of the Group Lasso regression model has been introduced by Liu et al. (2013). The smoothed Group Lasso estimator is defined as :

$$\hat{\boldsymbol{\beta}}^{\text{SmoothedGL}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda_1\sum_{g=1}^{G}\sqrt{p_g}\|\boldsymbol{\beta}^g\|_2+\frac{\lambda_2}{2}\max_{g=1,\ldots,G}p_g\sum_{g=1}^{G-1}\zeta_g\left(\frac{\boldsymbol{\beta}^g}{\sqrt{p_g}}-\frac{\boldsymbol{\beta}^{g+1}}{\sqrt{p_{g+1}}}\right)^2, \tag{2.8}$$

where λ_1 and λ_2 are two regularization parameters and ζ_g is the canonical correlation (Johnson & Wichern 2002) between two jth and $(j+1)$th groups.

Similarly to the Group Lasso model, the first part of the Smoothed Group Lasso penalty allows automatic group selection while ensuring sparsity thanks to the ℓ_1 penalty. The second penalty term allows to take into account possible correlations between adjacent groups. Indeed, when $\zeta_g = 0$ for all $g \in [1,\ldots,G]$, then this penalty reduces to zero and the Smoothed Group Lasso estimator is equivalent to the Group Lasso model. Conversely, when there is $\zeta_g > 0$, then the two adjacent groups g and $g+1$ are correlated and the smoothed Group Lasso tends to shrink the corresponding regression coefficients similarly.

Sparse Group Lasso. In some cases, sparsity of groups and within each group are required. Toward this end, the Sparse Group Lasso model has been introduced (Simon et al. 2013). It is defined as:

$$\hat{\boldsymbol{\beta}}^{\text{SparseGL}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\sum_{g=1}^{G}\sqrt{p_g}\|\boldsymbol{\beta}^g\|_2, \tag{2.9}$$

The Sparse Group Lasso model combines then the two ℓ_1 and ℓ_2 penalties in order to enable bi-level selection. Indeed, variable selection is carried out at the group level and at the level of individual covariates, resulting in the selection of relevant groups as well as variables within these groups.

To sum up, it has been demonstrated that extending the simple sparsity model to considering more sophisticated *structured sparsity models*, which describe the interdependency between the explanatory variables, increases the interpretability of the results and leads to better prediction

and estimation performance when the prior knowledge matches data (Stojnic et al. 2009, Jenatton et al. 2009, Huang et al. 2011). Moreover, the structured sparsity regression models presented above have been extended to more general settings to encode prior knowledge on various sparsity patterns, where the key idea is to allow the groups to have an overlap. For instance, the hierarchical selection method of Zhao et al. (2006) assumed that the input variables form a tree structure, and designed groups so that the child nodes enter the set of relevant inputs only if its parent node does. Situations with arbitrary overlapping groups (Jenatton et al. 2011) and prior graph structure (Jacob et al. 2009) have also been studied.

Such sparse regularization models, capable of encoding sophisticated prior knowledge are used in a wide variety of applications in scientific fields such as neuroimaging (Gramfort & Kowalski 2009), face recognition (Bach et al. 2012) or bioinformatics (Rapaport et al. 2008).

2.2 Genetic precepts

2.2.1 Genome and genetic diversity

The cell is the structural, functional and reproductive unit of all living beings (except for viruses). Inside eukaryotic cells, the nucleus is the seat of a genetic heritage, the *genome*. The genome influences the development of the living beings and their characteristics, also called *traits* or *phenotypes*. For instance, external phenotypes include the size of an individual or the color of his eyes and a physiological trait can be his blood pressure.

The genome consists of one or more elements called *chromosomes*. The number of chromosomes differs from one species to another. For example, bacteria have only one chromosome while the human genome accounts 46 of them: 22 pairs of *autosomes*, and 2 sexual chromosomes. More specifically, each human being inherits one autosome from his father and the other autosome from his mother. A chromosome is an oriented sequence of 4 different molecules called *nucleotides* that are Adenine (A), Thymine (T), Cytosine (C) and Guanine (G). Such sequence of nucleotides constitutes the Desoxyribo Nucleic Acid or *DNA* (see Figure 2.2). According to a now fully deciphered code, some DNA sequences of letters allow the cell producing molecules participating in all mechanisms of life (breathing, eating...) : *proteines*. Such a DNA sequence that codes for a protein is called a *gene*.

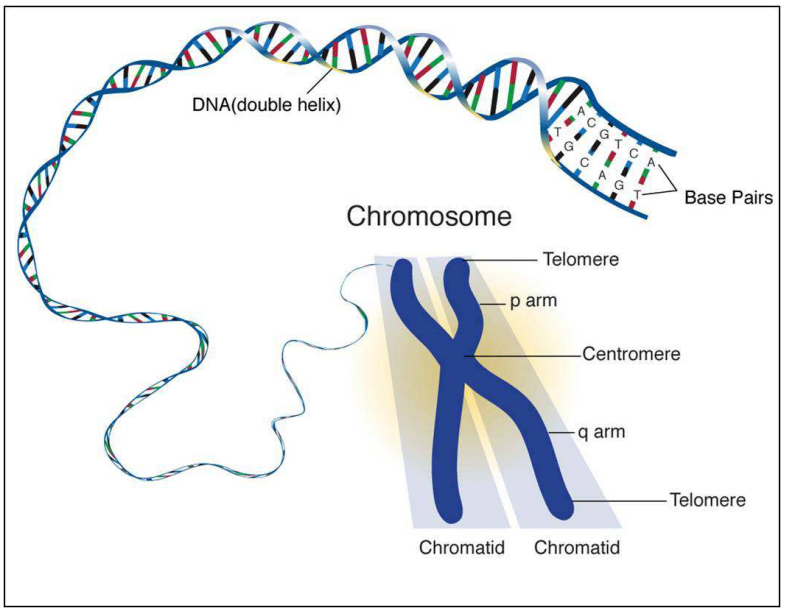


FIGURE 2.2: **From the chromosome to the DNA**. This figure is the property of Diaphragmatic Hernia Research and Exploration; Advancing Molecular Science

The size of the human genome is about 3.2 billion base pairs. As part of the *Human Genome Project*, its complete sequencing revealed a number of genes between 20000 and 25000, which represents about 5% of the genome. The remaining 95% contains a large amount of DNA whose function is not fully understood yet.

Genetic diversity within a population is mainly due to two types of events: *mutation* and *recombination*.

Mutation

A mutation is a random and spontaneous change in the sequence of the DNA which may affect one or more nucleotides. It can for instance correspond to a deletion of a base or the insertion of a novel base in a sequence (left panel of Figure 2.3). Depending on the affected base, this mutation can be "silent", that is with no effect on the resulting protein, or conversely have a positive or negative effect on the corresponding protein and therefore on the individual.

Moreover, the mutations allow the genetic analysis: the variability introduced by those mutations allows geneticists to identify and locate genetic factors involved in genetic diseases. Thus, mutations such as those occurring at a single nucleotide (also termed SNP for Single Nucleotide Polymorphism), can serve as "genetic markers". This aspect will be detailed in a following section.

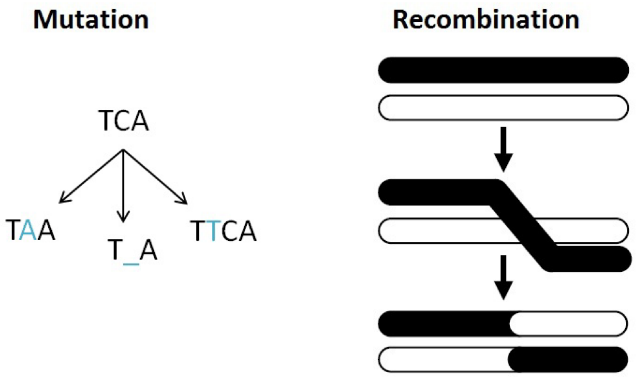


FIGURE 2.3: Schematics of mutation and recombination phenomena.

Recombination

Another source of genetic diversity is the phenomenon of recombination. Recombination takes place during the formation of gametes[1]: the *meiosis*. The right panel of Figure 2.3 illustrates the recombination between two homologous chromosomes. Suppose that one member of the chromosome pair is painted black and the other member is painted white. Instead of inheriting an all-black or an all-white parental chromosome, the offspring inherits a chromosome that alternates between black and white. The point of exchange is called *crossover*.

2.2.2 Genotype and haplotype

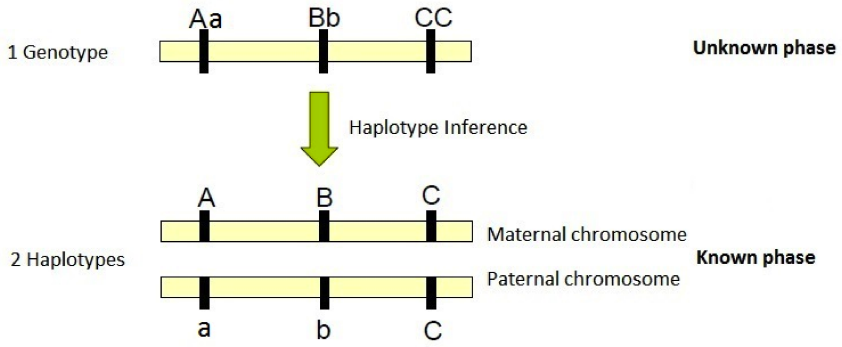


FIGURE 2.4: Haplotype phasing from genotype.

In general, a string of consecutive alleles lying on the same chromosome constitutes a *haplotype*. The alleles appearing in the haplotype are said to be *in phase*. In humans, for a given locus, one

[1] cells involved in reproduction

allele is inherited from the mother and the other allele is passed down from the father. The combination of two such alleles constitute a *genotype*. In order to illustrate the difference between haplotype and genotype we will consider the following example presented in Figure 2.4. This schematics shows 3 loci distributed in a chromosome. The observation of the genotypes Aa, Bb and CC does not provide knowledge on the phase of these genotypes. In other words, a step of *haplotype inference* is necessary in order to distinguish between the two possible combinations of haplotype pairs: (ABC, abC) or (AbC, aBC).

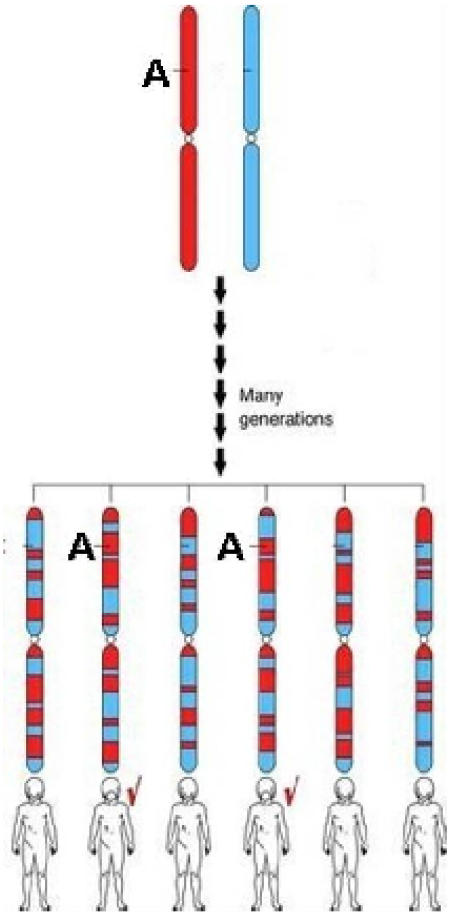


FIGURE 2.5: Two ancestral chromosomes being reshuffled through recombination over many generations to yield different descendant chromosomes.
Copyright http://www.hapmap.org/

Figure 2.5 shows how, over generations, new haplotypes appear from ancestral haplotypes through mutations in DNA sequences and recombination between them. If a mutation A is present on the ancestral haplotype 1 (red) and absent from the ancestral haplotype 2 (blue), then some of the offsprings who have inherited that part of the ancestral chromosome will carry the mutation. Furthermore, individuals with this mutation A have a high chance that the neighboring region of A is identical to that corresponding to the ancestral chromosome 1. This is explained by the fact that the probability that recombination events have occurred between the

mutation and the neighborhood is very low. This observation illustrates the phenomenon of *linkage disequilibrium* between close loci that are jointly transmitted over generations. This aspect is described in more detail in Section 2.3.

2.2.3 Hardy-Weinberg equilibrium

The Hardy-Weinberg Equilibrium (HWE) law (Hardy 1908, Weinberg 1908) concerns the relationship between allele frequencies and genotype frequencies in a population under certain assumptions. Let us consider a locus with the two alleles A and a with frequencies p_A and p_a respectively. The HWE law states that the frequencies of the genotypes AA, Aa and aa equal to p_A^2, $2p_Ap_a$ and p_a^2 respectively and the allele and genotype frequencies are constant over generations. The necessary assumptions to reach this equilibrium are (i) infinite population size, (ii) random mating process and finally (iii) no natural selection, no mutations and no population migration.

In real life, one or more of these assumptions can be violated, and a deviation from the Hardy-Weinberg proportions can then be observed. Several statistical tests of the HWE consisting in comparing the observed and expected genotype proportions have been proposed (Weir & Cockerham 1996, Guo & Thompson 1992).

2.3 Linkage Disequilibrium

2.3.1 Definition

Linkage disequilibrium (LD) describes a dependence relationship between two alleles at two different loci. We say that two loci are in linkage disequilibrium if the probability of observing this particular combination of alleles (or haplotype) does not equal the product of the probabilities of observing each allele individually. As cited in Ardlie et al. (2002), many factors can increase or decrease the LD. The most important factor is the genetic recombination which separates two loci on the same chromosome and therefore breaks the statistical dependance between them. Other factors can influence the LD such as the population structure or the natural selection.

2.3.2 Pairwise measures of LD

A number of measures for the strength of LD have been proposed. They can be broadly differentiated by their ability to consider exactly two loci or more than two loci at a time. There is a vast amount of literature on the topic of LD measures (Lewontin 1988, Devlin & Risch 1995, Jorde 2000, Agapow & Burt 2001, Rinaldo et al. 2005). However, the most commonly used measures are limited to LD between two loci.

To formally introduce pairwise LD measures, let consider the distribution of alleles for n individuals across two bi-allelic loci 1 and 2 with the possible alleles a/A and b/B respectively. Let us first assume that 1 and 2 are independent of one another. That is, the presence of an allele at locus 1 does not influence the presence of a particular allele at locus 2. Furthermore, let p_a, p_A, p_b and p_B denote the population frequencies for a, A, b and B alleles respectively. Since each individual carries two homologous chromosomes, there are in total $N = 2n$ observations across the n individuals of our sample. The expected distribution of alleles under independence between loci 1 and 2 are given in Table 2.3.

		Locus 2		
		B	**b**	
Locus 1	**A**	$n_{\text{AB}} = Np_Ap_B$	$n_{\text{Ab}} = Np_Ap_b$	$n_{\text{A.}} = Np_A$
	a	$n_{\text{aB}} = Np_ap_B$	$n_{\text{ab}} = Np_ap_b$	$n_{\text{a.}} = Np_a$
		$n_{\text{.B}} = Np_B$	$n_{\text{.b}} = Np_b$	$N = 2n$

TABLE 2.3: Expected allele distribution under independence of the loci 1 and 2.

If now the loci 1 and 2 are in fact dependent, then the observed counts will deviate from the numbers in Table 2.3. The amount of such deviation is represented by the scalar $\mathcal{D}$ in Table 2.4.

		Locus 2		
		B	**b**	
Locus 1	**A**	$n_{\text{AB}} = N(p_Ap_B + \mathcal{D})$	$n_{\text{Ab}} = N(p_Ap_b - \mathcal{D})$	$n_{\text{A.}}$
	a	$n_{\text{aB}} = N(p_ap_B - \mathcal{D})$	$n_{\text{ab}} = N(p_ap_b + \mathcal{D})$	$n_{\text{a.}}$
		$n_{\text{.B}}$	$n_{\text{.b}}$	$N = 2n$

TABLE 2.4: Allele distribution under LD.

$\mathcal{D}$ is one of the earliest measures of linkage disequilibrium to have been proposed. It quantifies the difference between the observed frequencies of a two-loci haplotype and the expected frequencies if the alleles were sampled at random. According to Table 2.4, $\mathcal{D}$ can also be written as follows:

$$\mathcal{D} = p_{AB} - p_Ap_B = p_{ab} - p_ap_b.$$

Hence, the greater the value of $\mathcal{D}$, the more the loci are in linkage disequilibrium. Nevertheless, the range of $\mathcal{D}$ si highly dependent on the specific values of the allele frequencies, which makes difficult the comparison of LD among many pairs of markers with diverse frequencies. For this reason, several other measures have been advised and the two most common measures are $\mathcal{D}'$ and r^2.

Introduced by Lewontin (1964), $\mathcal{D}'$ is a normalized form of $\mathcal{D}$:

$$\mathcal{D}' = \frac{\mathcal{D}}{\mathcal{D}_{\text{max}}}$$

with

$$\mathcal{D}_{\text{max}} = \begin{cases} \min(p_A p_b; p_a p_B) \text{ if } \mathcal{D} > 0 \\ \min(p_a p_b; p_A p_B) \text{ if } \mathcal{D} < 0. \end{cases}$$

$\mathcal{D}'$ takes its values between -1 and 1 and its principal usage is in characterizing historical patterns of recombination. A value of $\mathcal{D}' = 1$ indicates a *complete LD*, that is an absence of recombination event. In this case, at most three out of the four possible two-loci haplotypes are observed in the sample.

The r^2 coefficient (Hill & Robertson 1968) is determined by dividing the square of $\mathcal{D}$ by the product of the four allele frequencies:

$$r^2 = \frac{\mathcal{D}^2}{p_a p_A p_b p_B}.$$

Whereas many historical mutations in a recombination-free region have $\mathcal{D}' = 1$, both mutation history and recombination drive $r^2 = 1$.

To better understand the interest of the r^2 measure, let us introduce the two random variables X and Y relative to the loci 1 and 2. X will be defined as the indicator variable of the A-carrying event and Y the indicator variable of the B-carrying event. Consequently, both X and Y follow a Bernoulli distribution of parameters respectively p_A and p_B. Knowing that the expected value of a Bernoulli random variable of parameter p is p and its variance equals to $p(1-p)$, the squared correlation between X and Y can be written as follows:

$$\begin{aligned}
\text{Corr}^2(X,Y) &= \frac{\text{Cov}^2(X,Y)}{\text{Var}(X)\text{Var}(Y)} \\
&= \frac{(\mathbb{E}[XY]-\mathbb{E}[X]\mathbb{E}[Y])^2}{\text{Var}(X)\text{Var}(Y)} \\
&= \frac{(p_{AB}-p_A p_B)^2}{p_A(1-p_A)p_B(1-p_B)} \\
&= \frac{\mathcal{D}^2}{p_a p_A p_b p_B} \\
&= r^2.
\end{aligned}$$

Therefore, the LD measure r^2 between two loci corresponds to the correlation between the indicators of the presence of the major allele (or minor allele) at these two loci. Its values are then between 0 and 1. A r^2 of 1, called *perfect LD*, indicates that the two loci have not been separated by recombination and have the same allele frequency. In this case, exactly two out of the four possible haplotypes are observed in the sample and the two loci provide the same information. A r^2 of 0 indicates a perfect equilibrium between the two loci.

$\mathcal{D}'$ is mainly used in linkage studies which focus on the transmission issue. Conversely, in association studies, r^2 is favored as there is a direct relationship between power to detect a causal variant and the r^2 measures between the causal and genotyped variants (Pritchard & Przeworski 2001, Kruglyak 1999). The notions of linkage studies and association studies will be presented in Section 2.4.

2.3.3 Estimating LD

In practice, in population-based investigations, the counts in the contingency Table 2.4 are not observed. Only genotype counts as in Table 2.5 are known. Two main approaches have then been developed to infer the haplotype frequencies p_{aa}, p_{Aa}, p_{bb} and p_{Bb} from unphased genotype data to be able to assess the amount of LD between the two loci.

		Locus 2		
		BB	**Bb**	**bb**
Locus 1	**AA**	n_1	n_2	n_3
	Aa	n_4	n_5	n_6
	aa	n_7	n_8	n_9

TABLE 2.5: Genotype counts for 2 bi-allelic loci.

An estimation of the amount of LD between two loci has been proposed by Clayton & Leung (2007). The approach consists in inferring the haplotype counts n_{AA}, n_{Aa}, n_{BB} and n_{Bb} from the genotype contingency Table 2.5. In particular, by assuming Hardy-Weinberg equilibrium in the population, we notice that the pair of haplotypes of each genotype configuration in Table 2.5 can be inferred except for the double heterozygote: Aa in locus 1 and Bb in locus 2. For example, the only possible pair of haplotypes for the n_1 individuals with genotypes aa and bb is (ab, ab). Conversely, for the n_5 individuals carrying the genotypes aA and bB, two pairs of haplotypes are plausible: (ab, AB) or (aB, Ab). Therefore, if we note p the proportion of the doubly heterozygous subjects to carry the haplotypes (ab, AB) and thus $(1-p)$ to carry (aB, Ab), four relationships between the genotype counts and the haplotype counts can be deduced:

$$\begin{cases} n_{\text{AB}} = 2n_1 + n_2 + n_4 + pn_5 \\ n_{\text{Ab}} = 2n_3 + n_2 + n_6 + (1-p)n_5 \\ n_{\text{aB}} = 2n_7 + n_8 + n_4 + (1-p)n_5 \\ n_{\text{ab}} = 2n_9 + n_8 + n_6 + pn_5. \end{cases} \tag{2.10}$$

By adding the two equalities

$$\begin{cases} pn_{\text{aB}}n_{\text{Ab}} = (1-p)n_{\text{AB}}n_{\text{ab}} \\ n_{\text{AB}} + n_{\text{Ab}} + n_{\text{aB}} + n_{\text{ab}} = 2n, \end{cases} \tag{2.11}$$

this leads to a cubic equation in p:

$$\begin{aligned} &p(n_2 + 2n_3 + n_5 - pn_5 + n_6)(n_4 + n_5 - pn_5 + 2n_7 + n_8) \\ &-(1-p)(2n_1 + n_2 + n_4 + pn_5)(2n_9 + n_8 + n_6 + pn_5) = 0. \end{aligned} \tag{2.12}$$

Equation 2.12 has either one or three real solutions between 0 and 1 because it changes of sign between $p = 0$ and $p = 1$. Nevertheless, as discussed in Gaunt et al. (2007), simulations suggest that obtaining more than one biologically possible solution to the cubic equation occurs when small sample size, sampling errors or non-random mating result in a distortion of sample HWE. Otherwise, under perfect sample HWE, the cubic equation assumes a single real solution. Once

this unique value of p assessed, the genotype frequencies and then the haplotype proportions involved in the expression of the LD can be calculated using the system 2.10.

In the same spirit of this approach, a maximum likelihood method can be adopted for estimating the LD between a pair of loci. More particularly, let $p_1, \ldots, p_9$ be the genotype frequencies corresponding to the $n_1, \ldots n_9$ genotype counts. Therefore, the vector $(n_1, \ldots, n_9)$ follows a multinomial distribution of parameters $(p_1, \ldots, p_9)$

$$f(n_1, \ldots, n_9 | p_1, \ldots, p_9) = \frac{n!}{n_1! \ldots n_9!} p_1^{n_1} \ldots, p_9^{n_9},$$

and the multinomial likelihood would be written:

$$L((p_{AB}, p_{Ab}, p_{aB}, p_{ab}) | n_1, \ldots, n_9) = \prod_{j=1}^{9} p_j^{n_j}.$$

		Locus 2		
		BB	**Bb**	**bb**
Locus 1	**AA**	$p_1 = p_{AB}^2$	$p_2 = 2p_{AB}p_{Ab}$	$p_3 = p_{Ab}^2$
	Aa	$p_4 = 2p_{AB}p_{aB}$	$p_5 = 2p_{AB}p_{ab} + 2p_{Ab}p_{aB}$	$p_6 = 2p_{Ab}p_{ab}$
	aa	$p_7 = p_{aB}^2$	$p_8 = 2p_{aB}p_{ab}$	$p_9 = p_{ab}^2$

TABLE 2.6: Relationships between genotype and haplotype frequencies.

Therefore, by exploiting the relationships between the genotype and haplotype frequencies in Table 2.6, the log-likelihood would then give:

$$\begin{aligned} \log(L) \quad = \quad & 2n_1 \log(p_{AB}) + n_2 \log(2p_{AB}p_{Ab}) + 2n_3 \log(p_{Ab}) + n_4 \log(2p_{AB}p_{aB}) \\ & + n_5 \log(2p_{AB}p_{ab} + 2p_{Ab}p_{aB}) \\ & + n_6 \log(2p_{Ab}p_{ab}) + 2n_7 \log(p_{aB}) + n_8 \log(2p_{aB}p_{ab}) + 2n_9 \log(p_{ab}). \end{aligned}$$

Given that the solution must verify $p_{AB} + p_{Ab} + p_{aB} + p_{ab} = 1$, we seek to maximize under constraint the following quantity:

$$\log(L) - \lambda \left(\sum_{k=1}^{4} p_k - 1 \right). \tag{2.13}$$

Setting to zero the four partial derivatives of the expression 2.13 leads to the following equations:

$$\begin{cases} \dfrac{2n_1+n_2+n_4}{p_{AB}} + \dfrac{n_5 p_{ab}}{p_{AB}p_{ab}+p_{Ab}p_{aB}} = \lambda, & (2.14a) \\ \dfrac{2n_3+n_2+n_6}{p_{Ab}} + \dfrac{n_5 p_{aB}}{p_{AB}p_{ab}+p_{Ab}p_{aB}} = \lambda, & (2.14b) \\ \dfrac{2n_7+n_4+n_8}{p_{aB}} + \dfrac{n_5 p_{Ab}}{p_{AB}p_{ab}+p_{Ab}p_{aB}} = \lambda, & (2.14c) \\ \dfrac{2n_9+n_6+n_8}{p_{ab}} + \dfrac{n_5 p_{AB}}{p_{AB}p_{ab}+p_{Ab}p_{aB}} = \lambda, & (2.14d) \\ p_{AB}+p_{Ab}+p_{aB}+p_{ab} = 1. & (2.14e) \end{cases}$$

On one hand, we can easily deduce the value of λ by summing the first four equations:

$$\lambda = 2\sum_{i=1}^{9} n_i = 2n$$

On the other hand, we have:

$$\begin{cases} p_{AB}+p_{Ab}+p_{aB}+p_{ab} = 1, \\ p_{AB}+p_{Ab} = \frac{2(n_1+n_2+n_3)+n_4+n_5+n_6}{2n}, (p_{AB}2.14a + p_{Ab}2.14b) \\ p_{AB}+p_{aB} = \frac{2(n_1+n_4+n_7)+n_2+n_5+n_8}{2n}, (p_{AB}2.14a + p_{aB}2.14c) \\ p_{aB}+p_{ab} = \frac{2(n_7+n_8+n_9)+n_4+n_5+n_6}{2n}, (p_{aB}2.14c + p_{ab}2.14d) \\ p_{Ab}+p_{ab} = \frac{2(n_3+n_6+n_9)+n_2+n_5+n_8}{2n}, (p_{Ab}2.14b + p_{ab}2.14d). \end{cases} \quad (2.15)$$

Also, we can write the quantities p_{Ab}, p_{aB} and p_{ab} as functions of p_{AB}, $p_{A.}$ and $p_{B.}$:

$$\begin{aligned} p_{Ab} &= p_{A.} - p_{AB} & (2.16) \\ p_{aB} &= p_{B.} - p_{AB} & (2.17) \\ p_{ab} &= 1 - (p_{AB}+p_{Ab}+p_{aB}) & (2.18) \\ &= 1 + p_{AB} - p_{A.} - p_{B.}. \end{aligned}$$

Finally, by writing that $p_{A.} = \frac{2(n_1+n_2+n_3)+n_3+n_4+n_5}{2n}$ and $p_{B.} = \frac{2(n_1+n_4+n_7)+n_2+n_5+n_8}{2n}$, then the only missing estimation is p_{AB} to assess the LD. Thus, by using Equation 2.14a, we get:

$$\begin{aligned} \alpha_1(p_{AB}p_{ab} + p_{Ab}p_{aB}) + n_5 p_{AB}p_{ab} - \lambda p_{AB}(p_{AB}p_{ab} + p_{Ab}p_{aB}) &= 0 \\ (\alpha_1 - \lambda p_{AB})(p_{AB}p_{ab} + p_{Ab}p_{aB}) + n_5 p_{AB}p_{ab} &= 0, \end{aligned}$$

with $\alpha_1 = 2n_1 + n_2 + n_4$.

We have then a cubic equation in p_{AB}:

$$\begin{aligned} -2\lambda p_{AB}^3 + (2\alpha_1 + n_5 - \lambda + 2\lambda(p_{A.} + p_{B.}))\, p_{AB}^2 \\ + (-\lambda p_{A.}p_{B.} + \alpha_1 - (p_{A.} + p_{B.})(2\alpha_1 + n_5))\, p_{AB} + \alpha_1 p_{A.}p_{B.} = 0 \end{aligned} \tag{2.19}$$

The relationships 2.16, 2.17 and 2.18 allow to assess the value of the LD.

In practice, the use of these two methods to assess the pairwise LD among a string of consecutive loci can be fastidious as the number of haplotype frequencies to estimate becomes important. Therefore, several *haplotype reconstruction* methods based on the expectation-maximization algorithm (Dempster et al. 1977, Excoffier & Slatkin 1995, Lou et al. 2003) or on Hidden Markov Models (HMM) (Stephens & Donnelly 2003, Delaneau et al. 2008) have been developed in order to estimate these haplotype frequencies and therefore assess the LD measures introduced above.

Ultimately, through characterizing regions of high LD, it was observed that the recombination points in the human genome did not appear random, but rather seemed to cluster in specific regions, subsequently creating a block-like structure on the genome (Gabriel et al. 2002). Several methods have been proposed to determine the boundaries of such *LD blocks* or *haplotype blocks*. Such methods will be reviewed in detail in the next chapter.

2.4 Genome-wide association studies

2.4.1 Epidemiology of complex diseases

Last et al. (2001) defines Epidemiology as "the study of the distribution and determinants of

health-related states and events in populations". This science aims at understanding and controlling diseases, identifying therapeutic targets and defining public health policies.

Genetic Epidemiology combines Epidemiology and Genetics. It corresponds to the study of the role of genetic factors in determining health and disease in families and in populations, and the interaction of such genetic factors with environmental factors. A formal definition of Genetic Epidemiology was proposed by Morton (1982): "a science which deals with the etiology, distribution and control of disease in groups of relatives and with inherited causes of disease in populations."

Different designs of genetic studies exist and can be classified according to the question they aim to answer :

- **Familial aggregation studies**: is there a genetic component to the disease? What is the relative contribution of the environment compared to genes?
- **Segregation studies**: seek to more precisely identify the factors responsible for familial aggregation. Is the aggregation due to environmental, cultural or genetic factors?
- **Linkage studies**: on which part of the genome is the disease gene located?
- **Association studies**: which allele(s) of which gene(s) is associated with the disease? We refer to such alleles as *susceptibility alleles*, *disease susceptibility loci (DSL)* or *causal loci*.

Genetic diseases can have two types of etiologies (or causes): *monogenic* (also called single-gene) or *multifactorial* (also called complex). Monogenic diseases result from modifications in a single gene in the organism while complex diseases are caused by a combination of genetic, environmental, and lifestyle factors. The vast majority of diseases fall into this category and their study is a lot more complex due to the nature and the interplay between the factors concerned. Examples of such diseases are Alzheimer's disease, Type-1-Diabetes, Asthma, cancers, heart diseases or autoimmune diseases.

Supported by the accumulation of a large amount of data, made possible by recent technological development in the field of genotyping, Genetic Epidemiology met all the means necessary for the elucidation of the genetic mechanisms of major multifactorial diseases.

2.4.2 High-throughput genotyping

Advances in molecular biology over the last decade, and the development of novel technologies to manipulate DNA have allowed the availability of whole genome sequences. And the advent of rapid DNA sequencing methods, such as the Next-Generation Sequencing (NGS) technology (Check Hayden 2009), has greatly accelerated biological and medical research. Indeed, information obtained using sequencing allow researchers to identify changes in genes, associations with diseases and phenotypes, and identify potential drug targets.

Within the framework of this thesis, we have rather worked on genotype data which hold information about several types of genetic markers based on the variation of the DNA sequence. Examples of such markers are the Short Tandem Repeats (STR or microsatellites), Indels or Single Nucleotide Polymorphisms (SNPs). Furthermore,

A Single Nucleotide Polymorphism corresponds to the variation of a single base pair in the DNA sequence within a population. Most SNPs are bi-allelic, that is they involve two possible alleles. Their frequency throughout the genome and simplicity to characterize experimentally make the SNPs the markers of choice for investigators to establish a dense and precise mapping of the genome. To date, more than 160 million of SNPs have been identified (according to [2]dbSNP database) and they represent more than 90% of the known human genetic diversity (Kruglyak & Nickerson 2001). A SNP is characterized by its chromosomal location, its alleles and its Minor Allele Frequency (MAF). The minor allele refers to the less common allele in a population (with frequency less than 0.5).

When we seek to discover new biological mechanisms, it is important to start without genetic hypothesis a priori on the studied traits. Advances in biochemistry allowed to cross this barrier and genotype (that is characterize the genotype of a subject) the entire genome through genotyping chips. There are mainly two technologies currently providing genotyping microarrays based on different approaches in selecting SNPs: *Illumina* and *Affymetrix*. These two technologies combine two different approaches of marker selection: (i) random selection and (ii) tagSNP selection.

The random selection is usually used when no prior information is available about the disease and the potential regions of the genome that could intervene in its mechanism. Such selection

[2]See Section A.1

approach is nowadays possible on a large amount of markers with the development of high-throughput genotyping technologies.

Conversely, a tagSNP selection is based on a certain type of SNPs. We have seen in Section 2.3 that the human genome is divided into LD blocks. A *tagSNP* is a SNP among a LD block that captures the entirety of the genetic variability of this block by itself. Using the terminology in Ding & Kullo (2007), there are two ways for choosing the optimal minimum subset of representative SNPs from a set of SNPs: the *tagging SNPs* method (tSNPs) and the *haplotype tagging SNPs* approach (htSNPs). htSNPs are selected based on the haplotype-block model of LD pattern in a region of interest and represent the common haplotypes inferred from the original set of SNPs (Johnson et al. 2001, Patil et al. 2001, Zhang et al. 2002, 2003). For example, Patil et al. (2001) proposed a computational framework with the aim to minimize the number of htSNPs required to distinguish all common haplotypes within each block. On the other hand, tSNPs are selected based on pairwise LD measures, such that a tSNP predicts partially or completely the state of other SNPs (Weale et al. 2003, Halldorsson et al. 2004, Carlson et al. 2004). For instance, Carlson et al. (2004) evaluates the r^2 measures between all the markers and forms "bins" of SNPs, each of which has an r^2 greater than a user-defined threshold ($r^2 > 0.8$ is proposed by the authors) with one or more SNPs within that bin. Within each bin, tSNPs are SNPs which are in linkage disequilibrium with all the other SNPs with more than the specified threshold of r^2.

In practice, with both the random and tagSNP selection strategies, the causal locus can be missed (not genotyped) and the genotyped SNPs that are in linkage disequilibrium with the disease susceptibility locus are then in an *indirect association* with the disease (see Figure 2.6).

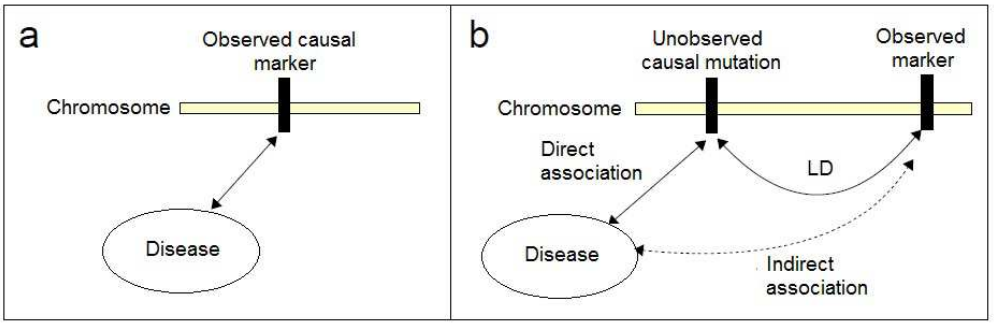


FIGURE 2.6: a) Direct association disease-observed marker. b) Indirect association disease-observed marker.

The work carried out in this thesis falls within the framework of association studies and more particularly those on mapping the complete genome, that is the genome-wide methods. Genome-wide Association Studies (GWAS) aim to identify genetic markers that are associated with a

(qualitative or quantitative) phenotype of interest in a given population, by comparing the DNA of unrelated individuals within that population. GWAS are usually conducted on SNP markers.

In the case of a qualitative phenotype as a disease status, a sample of unrelated normal individuals (called *controls*) and unrelated individuals suffering from the disease (called *cases*) are genotyped. GWAS assume that the discovery of disease susceptibility loci can be achieved by comparing allele frequencies of the SNPs between cases and controls. When the allele of a SNP is significantly more frequent in cases than in controls, then it can be deduced that this allele is (directly or indirectly) associated with the disease. Unlike gene-candidate studies that target a set of potential genes (Tabor et al. 2002), the GWAS approach is a non-directed and exploratory strategy that investigates much of the hole genome without any a priori on the location of the causal loci.

Selecting individuals participating to a GWA study that form a homogeneous cohort is very important to avoid biased findings. More particularly, in a case-control design, the two groups of individuals should be comparable. To this end, samples are included in a study according to certain characteristics such as gender, age or ethnicity. Equivalently, certain features of the markers have to be investigated in order to determine which SNPs can reasonably be included in the GWA study without leading to incoherent results. These verifications include, for example, the amount of missing data or the minor allele frequency for each marker. In this manuscript, we will focus on common variants (with MAF$> 5\%$). This choice will be discussed in the conclusion section.

To date, thousands of GWA studies have been conducted in over 80 diseases and traits. A continuously updated online catalog of the published studies is available at `http://www.genome.gov/gwastudies/` (Welter et al. 2014).

The standard analytical model and most widely used in GWAS is the single-marker analysis.

2.4.3 Single-marker analyses

In association studies, the *single-locus*, also called *single-marker approach* (SMA) consists in analyzing individually the association of each marker to the phenotypic trait studied.

Given a genotype (or design) matrix $\mathbf{X}$, when the phenotype studied is quantitative, for each marker $\mathbf{X}_{.j}$, we fit a single-predictor equation $\mathbf{y} = \boldsymbol{\beta_0} + \beta_j \mathbf{X}_{.j} + \boldsymbol{\varepsilon}$. The significance of the

estimated parameter β_j is assessed by assuming that the errors are normally distributed, and then deriving a p-value from a t-test against an intercept-only model:

$$H_0 = \{\beta_j = 0\}.$$

As a matter of fact, applying such a linear regression model requires making assumptions about the *genetic model*, that is the mode of inheritance of the trait. The encoding of the genotype matrix $\mathbf{X}$ will depend on the genetic model assumed.

The classical genetic models are the following:

- **additive model:** the relative risk of carrying two copies of the risk allele is the square of the risk of carrying one copy, since "additive" refers to the log-scale.
- **dominant model:** the risk of carrying two copies of the risk allele is the same as carrying one copy.
- **recessive model:** there is no increased risk associated with carrying one copy of the risk allele, but there is an increased risk associated with carrying two copies.

When dealing with SNP genotypes $AA/Aa/aa$, these shall then be coded in $0/1/2$, $0/1/1$ or $0/0/1$ according to wether the genetic model is assumed to be additive, dominant or recessive respectively.

The mode of inheritance of the trait studied is generally unknown. The additive model is the most commonly used to code the genotype matrix in GWAS.

Alternatively, when dealing with case-control studies, that is a disease status phenotype, the association analysis of a SNP of interest can be carried out using a Pearson χ^2 test of the status-by-genotype (2×3 table) frequencies. As with any χ^2 test based on a contingency table, all cells ought to have an expected value > 5. However, for rare or highly polymorphic loci (e.g. microsatellites), the probability of empty or poorly filled cells increases with the dimensions of the table. In practice, rows or columns of the contingency table can be merged in order to meet the requirements for the χ^2 test. Alternatively, computationally intensive methods may be considered such as parametric bootstrapping, or permutation testing such as Fisher's exact test.

An alternative to the use of the χ^2 test is to assess the association of a marker of interest with a qualitative trait using the *logistic regression model*. By analogy to ordinary linear regression, in

linear logistic regression the probability $p_i = \mathbb{P}(y_i = 1)$ of case i given the predictor vector $\mathbf{X}_{\cdot j}$ is written as:

$$p_i = \frac{e^{\beta_0 + \beta_j \mathbf{X}_{\cdot j}}}{1 + e^{\beta_0 + \beta_j \mathbf{X}_{\cdot j}}},$$

or equivalently:

$$\text{logit}(p_i) = \log(\frac{p_i}{1 - p_i}) = \beta_0 + \beta_j \mathbf{X}_{\cdot j}.$$

The β_0 and β_j parameters can be estimated using a maximum likelihood method. The Wald statistic:

$$Z = \frac{\hat{\beta}}{\sqrt{Var(\hat{\beta})}}$$

can then be used to assess the significance of the regression coefficient estimated by the logistic model. Z follows a standard normal distribution $\mathcal{N}(0, 1)$ under the null hypothesis $H_0 = \{\beta_j = 0\}$.

	AA	**aA/Aa**	**aa**	**Total**
Case	D_0	D_1	D_2	n_D
Control	H_0	H_1	H_2	n_H
Total	n_{AA}	$n_{aA} + n_{Aa}$	n_{aa}	n

TABLE 2.7: Genotypic table representing the number of individuals for each genotype configuration and each disease status.

One particular case of the logistic regression model is the Armitage trend test (Cochran 1954, Armitage 1955) which assumes the genotype variable $\mathbf{X}_{\cdot j}$ coded in 0, 1 or 2. This test aims to find a linear trend between the probability of having the disease and the genotypes. The order of genotypes assumes that there is a quantitative effect depending on the number of reference allele in the genotype. Using the notations in Table 2.7, the Cochran-Armitage null hypothesis is:

$$H_0 : \{\frac{p_{D_0}}{p_0} = \frac{p_{D_1}}{p_1} = \frac{p_{D_2}}{p_2}\},$$

where p_i is the proportion of individuals with genotype i. According to Sasieni (1997), the test statistic can be expressed as:

$$S_T = \frac{n[n(D_1 + 2D_2) - n_D(n_{aA} + n_{Aa} + 2n_{AA})]^2}{N_D N_H[n(n_{aA} + n_{Aa} + 4n_{aa}) - (n_{aA} + n_{Aa} + 2n_{aa})^2]}$$

This statistic follows a χ^2 distribution with 1 degree of freedom under the null hypothesis.

2.4.4 Haplotype association analyses

Overall, single-locus tests of association are fast, efficient and reliable. They are easy to implement whether the trait is categorical or continuous. Nevertheless, from a biological viewpoint, there is evidence that specific combinations of mutations may have a greater effect on phenotypic variation than any one of the individual causal variants alone. For this reason, multivariate linear models (see Section 2.1.2) have been applied to GWAS by considering SNPs as explanatory variables and the disease of interest as the response. Moreover, the block-like structure of common human genetic variation, due to linkage disequilibrium, is one of the key factors motivating the use of haplotypes to identify sets of loci that are associated with complex traits.

Haplotypes play also an important role in the design and implementation of genetic studies of complex diseases. Indeed, haplotypes can be used as variables of interest for detecting associations between a chromosomal region and a complex trait. When phased haplotype data are available, the goal of the haplotype analysis is to evaluate the relationship between a haplotype H and a vector of phenotypes $\mathbf{Y}$, adjusting for potential covariate effect E, using a generalized-linear-model (GLM) regression framework (McCullagh et al. 1989). More precisely, the GLM framework relates the mean $\boldsymbol{\mu} = E[\mathbf{Y}|H, E]$ to H and E as follows:

$$g(\boldsymbol{\mu}) = \boldsymbol{\alpha} + \mathbf{X}_H\boldsymbol{\beta} + \mathbf{X}_E\gamma.$$

Here, $\mathbf{X}_H$ denotes a design vector that models the effects of a subject's haplotype pair H on $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$ is the related vector of regression coefficients. Likewise, $\mathbf{X}_E$ denotes a design vector for modeling the subject's environmental effects with respective coefficient vector $\boldsymbol{\gamma}$. Finally, $\boldsymbol{\alpha}$ is a scalar intercept parameter. The form of the link function g depends on the distribution of the phenotype $\mathbf{Y}$. For a normally distributed outcome, the identity link $g(\boldsymbol{\mu}) = \boldsymbol{\mu}$ is typically

used, leading to a multiple linear regression. For a binary phenotype, we use the logistic link $g(\boldsymbol{\mu}) = \log[\boldsymbol{\mu}/(1-\boldsymbol{\mu})]$, corresponding to a logistic regression analysis.

The form of the haplotype design vector $\mathbf{X}_H$ must be specified prior to the haplotype analysis. As an example, suppose we were interested in assessing the effects of the haplotype $h^\star$ and define $I(A)$ as the indicator function that takes the value 1 or 0 depending on whether the event A is true or false, respectively. Then, for a subject with $H = (h_k, h_{k'})$, one can model a recessive effect for $h^\star$ using $\mathbf{X}_H = I(h_k = h_{k'} = h^\star)$, a dominant effect for $h^\star$ using $\mathbf{X}_H = I(h_k = h^\star) + I(h_{k'} = h^\star) - I(h_k = h_{k'} = h^\star)$, or an additive effect for $h^\star$ using $\mathbf{X}_H = I(h_k = h^\star) + I(h_{k'} = h^\star)$. The regression coefficient β related to a specific $h^\star$ is then estimated and haplotype-phenotype association can be conducted using tests of the form $H_0 : \beta = 0$ vs. $H_1 : \beta \neq 0$.

GLM-based score statistics for testing global and individual haplotype effects on the phenotype of interest (using either asymptotic or permutation-based p-values) and adjusting for the effects of covariates have also been developed (Schaid et al. 2002).

In presence of rare haplotypes which demonstrate large variability and then lead to invalid test statistics, several methods of haplotype clustering have been suggested (Durrant et al. 2004, Molitor et al. 2003, Seltman et al. 2003, Tzeng 2005). And Tzeng et al. (2006) proposed a haplotype association method based on the GLM framework and using haplotype clusters.

Finally, the field of large-scale genetic studies and more specifically the GWAS has recently boomed thanks to the development of computer tools increasingly sophisticated that allowed cataloguing (through databases) and statistical analysis (through software) of the large amount of data. Some of these bioinformatic resources are presented in Appendix A.

2.5 Conclusion

The aim of the present chapter was to review various statistical and genetic background necessary to conduct genome-wide association studies and haplotype analysis in the context of linkage disequilibrium. LD appears clearly as a central parameter in these large-scale GWA studies for many reasons. First, we have seen that the choice of the markers to genotype for a study depends on the amount of LD in the population. Second, from a statistical point of view, if a simple association test between each marker and the phenotype of interest is performed, then the power of the test, that is its ability to detect a true association, will mainly depend on the amount of LD between the tested marker and the causal loci. Finally, given the growing flow of genotype data, incorporating the linkage disequilibrium information in GWAS should ensure an easier analysis and interpretation of the results, for example by reducing the high-dimensionality of the data.

Chapter 3

Linkage disequilibrium block partitioning

3.1 Existing definitions of linkage disequilibrium blocks

By studying the distribution of linkage disequilibrium across the genome, several authors observed that LD is related to the distance between markers (Kruglyak 1999, Dunning et al. 2000, Abecasis et al. 2001, Pritchard & Przeworski 2001, Reich et al. 2001). More specifically, Figure 3.1 illustrates the negative correlation between the intermarker physical (base-pair) distance and the pairwise LD measures $\mathcal{D}'$ and r^2 for a set of 200 SNPs within chromosome 6 in a study on 605 HIV-infected patients (Dalmasso et al. 2008).

However, the rate of this decrease does not follow a regular pattern and is related to the particular location of the markers in the human genome (Taillon-Miller et al. 2000). These observations reflect the fact that the genome could be clustered into sets of high LD regions or blocks separated by short discrete segments of very low LD called *recombination hotspots* (Daly et al. 2001, Gabriel et al. 2002, Patil et al. 2001, Jeffreys et al. 2001). As described in Section 2.3.3, such regions exhibit limited haplotype diversity, so that a small number of distinct haplotypes account for most of the chromosomes in the population, and they are now termed *haplotype blocks*.

A range of methods have been proposed for defining haplotype blocks. Jeffreys et al. (2001) has defined blocks through direct measurement and localization of recombination hotspots. Apart from this approach, haplotype block definition methods can be classified into two main groups:

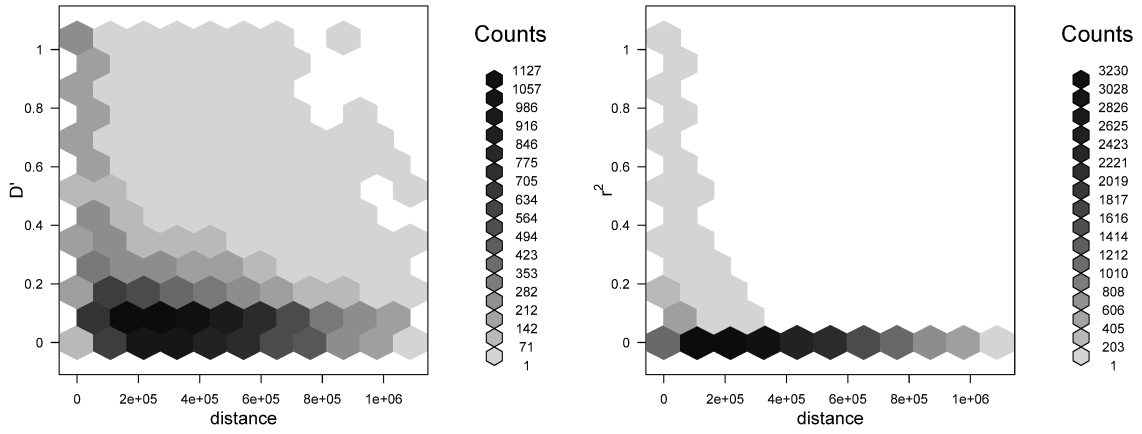


FIGURE 3.1: Decay of pairwise linkage disequilibrium measures $\mathcal{D}'$ (left panel) and r^2 (right panel) over physical distance for 200 SNPs of chromosome 6 in a study on 605 HIV-infected patients (Dalmasso et al. 2008).

those that define blocks as regions with limited haplotype diversity and those that make use of pairwise linkage disequilibrium measures to distinguish high LD regions from recombination hotspots (Cardon & Abecasis 2003, Wall & Pritchard 2003). Some other approaches combine these two strategies in modeling haplotype blocks. Existing LD block partitioning algorithms are summarized in Table 3.1.

When pairwise LD measures are used, a block is defined whenever all pairwise coefficients (adjacent and non-adjacent) within a region exceed some pre-defined threshold. Gabriel et al. (2002) refined this basic definition by using confidence limits on the pairwise LD measures $|D'|$. More specifically, values of $|D'|$ are divided into three categories: strong LD ($|D'|$ close to 1), weak LD ($|D'|$ significantly < 1) and intermediate/unknown LD. The haplotype blocks are then defined as the sets of consecutive loci over which a small proportion ($< 5\%$) show strong evidence of historical recombination. Similarly, Wang et al. (2002) use the $|D'|$ LD measures to define haplotype blocks. More particularly, the four gamete rule define these blocks as regions where all pairs of loci are in complete LD ($|D'| = 1$) or where at least one of the four possible haplotypes has a frequency below 0.01. These two LD block partitioning strategies are implemented in the software Haploview in addition to the Solid Spine method that was internally developed in the software. This third approach searches for a "spine" that is a region of strong LD where first and last loci of the block are in strong LD with all intermediate loci but the intermediate loci are not necessarily in LD with each other. The threshold beyond which the LD is considered as strong is fixed by the user. More recently, model-based approaches

Algorithm	Criterion			htSNP
	Pairwise LD	Haplotypes	Recombination hotspots	
Jeffreys et al. (2001)			X	
Daly et al. (2001)	X	X		
Patil et al. (2001)		X		X
Gabriel et al. (2002)	X			
Wang et al. (2002)	X			
Dawson et al. (2002)	X	X		
Zhang et al. (2002, 2003)		X		X
Zhu et al. (2003)	X			
Phillips et al. (2003)	X			
Twells et al. (2003)	X			
Shifman et al. (2003)	X			
Anderson & Novembre (2003)	X	X		
Mannila et al. (2003)		X		
Koivisto et al. (2003)		X		
Greenspan & Geiger (2004)		X		
Pattaro et al. (2008)	X			
Tomita et al. (2008)	X			
Mourad et al. (2011)	X			

TABLE 3.1: Three criteria are used in existing block partitioning algorithms. Jeffreys et al. (2001) has defined blocks through recombination hotspots. The remaining haplotype block definition methods can be classified into two main groups: those that use the pairwise LD measures and those that define the blocks as regions with limited haplotype diversity. The Patil et al. (2001) and Zhang et al. (2002, 2003) approaches allow to identify haplotype tagging SNPs.

using pairwise LD measures have been developed such as the MCMC Algorithm To Identify blocks of Linkage DisEquilibrium (MATILDE) (Pattaro et al. 2008) and methods using forest of hierarchical latent class models (Mourad et al. 2011) or the Echelon analysis (Tomita et al. 2008) to model LD.

Alternatively, when haplotypes are known, a haplotype block is usually defined when a small number of haplotypes (for example 3 to 5) account for a high proportion of the observations (75%–90%). For instance, Patil et al. (2001) required that in haplotype blocks, at least 80% of the observed haplotypes should be observed two or more times. Moreover, they locate the blocks boundaries so that the most common haplotypes within blocks can be identified using the smallest number of SNPs, called following Johnson et al. (2001), *haplotype tagging SNPs* (htSNPs). Zhang et al. (2002, 2003) formalized this approach by a dynamic programming algorithm.

Finally, Anderson & Novembre (2003) combine haplotype diversity within blocks and LD decay between blocks to find the optimal partition, using the Minimum Description Length principle

(MDL). This same principle is also used in Mannila et al. (2003), Koivisto et al. (2003) and Greenspan & Geiger (2004).

When the only available data are diploid genotype data in which haplotype phase is unknown, several haplotype inference approaches have been proposed to resolve haplotypes from unphased SNP data (Niu et al. 2002, Qin et al. 2002, Stephens & Donnelly 2003). Nevertheless, most of these methods remain computationally intensive when dealing with a large number of loci. Moreover, specific information about haplotypes are not necessarily needed when it comes to define the boundaries of LD blocks. Thus, pairwise methods appear to be easier to apply to genotype data. Nevertheless, in the most popular pairwise approaches, the thresholds of LD and the confidence limits used to define blocks remain subjective and arbitrary. One of the contributions of this thesis are, therefore, to propose an automated block partitioning approach using pairwise LD measures between SNPs. This method consists in a Ward's hierarchical clustering of SNPs with an adjacency constraint and using LD as a similarity measure. Then the Gap statistic model selection approach is applied to the obtained hierarchy in order to define the LD blocks. The remainder of this chapter will be dedicated to a review of cluster analysis and a detailed description of the proposed LD block partitioning approach.

3.2 Background on cluster analysis

3.2.1 Typology of cluster analysis methods

Unsupervised classification or *clustering* (Gordon 1999, Jain et al. 1999, Berkhin 2004) is one of the most important field in Data Mining. It consists in grouping together similar items while dissimilar ones are asserted to different groups, without any a priori knowledge on the obtained groups. Alternatively, when the prior knowledge on the data is present, the technique is called *supervised classification* or *discriminant analysis*. These input groups of items then are used for the classification of other sets of items. In both techniques, the groups of items are usually called *clusters* or *classes* and are used to synthesize the information contained in the initial set.

The two common approaches in statistical clustering are *partitioning clustering* and *hierarchical clustering*. In the partitioning clustering, one starts with the whole initial analyzed items in a same cluster, which is then split into G clusters. Usually, the number G has to be specified before the analysis. Model selection techniques can be used to determine the most appropriate

values of G, based usually on the selection of the "optimal" partition for a range of values of G. Partitioning methods usually produce clusters by optimizing a criterion function, and due to the combinatorial number of possibilities, the algorithm is run repeatedly. Among the most known partitioning methods, we cite the k-means (MacQueen et al. 1967), the dynamic clustering (Diday 1973) or the minimum spanning trees (Graham & Hell 1985).

The purpose of hierarchical classification is to produce a tree whose nodes represent clusters of the initial dataset. In particular, the initial set of items is the root of the tree while the leaves represent the singletons (clusters with one element/item). This type of structure thus provides an enhanced visual representation than the partitioning methods. The investigator can then select the suitable partitioning from its point of view, by making a trade off between the number of clusters and their homogeneity degree.

Different types of hierarchical clustering methods can be classified according to their initial items/clusters and the steps of their algorithm:

- **agglomerative hierarchical methods**: starting from the initial singletons/items, the clusters are successively merged into higher level clusters, until the entire set of items becomes a cluster. These methods are also called Ascending Hierarchical Classification (AHC).
- **hierarchical divisive methods**: starting from a single cluster (of all items), successive splits of clusters are performed to obtain smaller clusters (Rao 1971).

At the beginning of a clustering process, we have to select the appropriate items for clustering. In some cases, this choice of the dataset E of items to be clustered is apparent from the nature of the task at hand.

In many situations, the data is in the form of a table $\mathbf{X}$ of n individuals (in rows) described by p variables (in columns):

$$\mathbf{X} = (\mathbf{X}_{ij})_{n\times p} = \begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ n \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & \dots & j & \dots & p \end{array} \\ \begin{pmatrix} & & . & & \\ & & \vdots & & \\ \dots & & \mathbf{X}_{ij} \in \mathbb{R} & & \dots \\ & & \vdots & & \\ & & . & & \end{pmatrix} \end{array}.$$

In that case, the clustering approach is generally applied to the individuals but can easily extended to the set of variables.

In this chapter, we will focus on the problem of clustering the set E of variables $\mathbf{X}_{.1}, \mathbf{X}_{.2}, \ldots, \mathbf{X}_{.p}$.

The next section will be devoted to AHC methods which have been studied extensively (Sokal et al. 1963, Hartigan 1975) during the last decades and have a wide number of applications in many different fields such as bioinformatics, signal processing or web mining.

3.2.2 Agglomerative hierarchical methods

3.2.2.1 Measure of similarity

Performing an agglomerative hierarchical clustering requires to define a measure of *similarity* between the items to be clustered.

A similarity function measures the link between items of a set.

Definition 1. A similarity measure on a set E is a function:

$$\mathrm{Sim} : E \times E \to \mathbb{R}^+$$

having the properties:
$\forall (x, y) \in E^2$ such that $x \neq y$,

- $\mathrm{Sim}(x, y) = \mathrm{Sim}(y, x)$
- $\mathrm{Sim}(x, x) = \mathrm{Sim}(y, y) = \mathrm{Sim}_{\max} \geq \mathrm{Sim}(x, y)$,

where $\mathrm{Sim}_{\max}$ is a positive scalar.

To each similarity measure Sim may be associated a *dissimilarity* measure Diss defined by $\mathrm{Diss}(x, y) = \mathrm{Sim}_{\max} - \mathrm{Sim}(x, y)$. Diss will then have the property $\forall x \in E$, $\mathrm{Diss}(x, x) = 0$. In other words, the less the items x and y are alike, the higher is the value of the score $\mathrm{Diss}(x, y)$.

Definition 2. A *distance* function:

$$\mathrm{Dist} : E \times E \to \mathbb{R}^+$$

has the properties:
$\forall (x, y, z) \in E^3$,

- $\text{Dist}(x, y) = 0$ if and only if $x = y$
- $\text{Dist}(x, y) = \text{Dist}(y, x)$
- $\text{Dist}(x, z) \leq \text{Dist}(x, y) + \text{Dist}(y, z)$.

Ultimately, a distance is a dissimilarity since any distance satisfies the two properties of a dissimilarity as well as triangular inequality.

Some commonly used dissimilarities are :

Euclidean distance	$\|\mathbf{X}_{.i} - \mathbf{X}_{.j}\|_2 = \sqrt{\sum_{l=1}^{n} (\mathbf{X}_{li} - \mathbf{X}_{lj})^2}$
Manhattan distance	$\|\mathbf{X}_{.i} - \mathbf{X}_{.j}\|_1 = \sum_{l=1}^{n} \|\mathbf{X}_{li} - \mathbf{X}_{lj}\|$
Maximum distance	$\|\mathbf{X}_{.i} - \mathbf{X}_{.j}\|_\infty = \max_{l \in 1,\ldots,n} \|\mathbf{X}_{li} - \mathbf{X}_{lj}\|$.

For notation convenience, δ will denote a generic dissimilarity on the set E in the remainder of the manuscript.

3.2.2.2 Linkage criteria

After having chosen the similarity measure between items, we need to decide which agglomerative algorithm to apply. There are several agglomerative procedures and they can be distinguished by the way they define the distance from a newly formed cluster to a certain item, or to other clusters. Such a definition, termed *linkage criterion*, then specifies the distance between clusters as a function of the pairwise dissimilarities between items in the clusters:

$$d : \mathcal{P}(E) \times \mathcal{P}(E) \to [0, \infty[.$$

The most popular linkage criteria are:

- **single linkage**, that is the shortest pairwise dissimilarity between items in two different clusters:

$$d_{\text{sl}}(A, B) = \min\{\delta(\mathbf{X}_{.i}, \mathbf{X}_{.j}); \mathbf{X}_{.i} \in A, \mathbf{X}_{.j} \in B\}.$$

- **complete linkage**, that is the dissimilarity between the most distant pair of items coming from each of the two clusters:

$$d_{\text{cl}}(A, B) = \max\{\delta(\mathbf{X}_{.i}, \mathbf{X}_{.j}); \mathbf{X}_{.i} \in A, \mathbf{X}_{.j} \in B\}.$$

- **average linkage**, the average of the pairwise dissimilarities between all pairs of items coming from each of the two clusters:

$$d_{\text{al}}(A, B) = \frac{\delta(\mathbf{X}_{.i}, \mathbf{X}_{.j})}{p_A p_B}; \mathbf{X}_{.i} \in A, \mathbf{X}_{.j} \in B.$$

  with p_A and p_B the cardinals of A and B respectively.

- **Ward's criterion**:

$$d_{\text{wl}}(A, B) = \frac{p_A \times p_B}{p_A + p_B} \delta(\mathbf{g}_A, \mathbf{g}_B)^2, \tag{3.1}$$

  with $\mathbf{g}_A$ and $\mathbf{g}_B$ the centers of the clusters A and B respectively.

Proposition 1. Ward's linkage criterion can also be written as:

$$d_{\text{wl}}(A, B) = \sum_{i \in A \cup B} \delta(\mathbf{X}_{.i}, \mathbf{g}_{A \cup B})^2 - \sum_{i \in A} \delta(\mathbf{X}_{.i}, \mathbf{g}_A)^2 - \sum_{i \in B} \delta(\mathbf{X}_{.i}, \mathbf{g}_B)^2. \tag{3.2}$$

Indeed, when looked more closely, Ward's criterion (Ward Jr 1963) defines the distance between two clusters A and B as the increase of variability within groups (according to the dissimilarity δ chosen) when we merge them. Proposition 1 with the Euclidean distance is demonstrated in Appendix B.

For the previous four linkage criteria, there are relationships (Lance & Williams 1967) that simplify the calculation of distances between classes which are essential for the practical implementation of the AHC algorithm. Indeed, the generic Lance-Williams formula allows deriving the distance between a pair of clusters $A \cup B$ and C from previously calculated distances $d(A, C)$, $d(B, C)$ and $d(A, B)$:

$$d(A \cup B, C) = \alpha_1 d(A, C) + \alpha_2 d(B, C) + \beta d(A, B) + \gamma |d(A, C) - d(B, C)|,$$

where the parameters α_1, α_2, β and γ are determined by the linkage criterion used. Table 3.2 summarizes the parameter values for the most common linkage functions.

		Coefficients			
		α_1	α_2	β	γ
Linkage	**Single**	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
	Complete	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
	Average	$\frac{p_A}{p_A+p_B}$	$\frac{p_B}{p_A+p_B}$	0	0
	Ward	$\frac{p_A+p_C}{p_A+p_B+p_C}$	$\frac{p_B+p_C}{p_A+p_B+p_C}$	$-\frac{p_C}{p_A+p_B+p_C}$	0

TABLE 3.2: Coefficients in the Lance-Williams formula for different linkage criteria.

3.2.2.3 Graphical representation of a hierarchy

To sum up, the steps of an agglomerative hierarchical clustering procedure applied to the dataset E can be described as follows:

1. Start with p clusters containing each one item. Define a $p \times p$ dissimilarity matrix $D(i,j)$ (initialized to $\mathbf{0}_{p\times p}$) between items $\mathbf{X}_{.i}$ and $\mathbf{X}_{.j}$.
2. Find the most similar pair of clusters A and B in the sense of the linkage function chosen and merge them into a single cluster $A \cup B$.
3. Update the dissimilarity matrix (reduce its order by one) by replacing the individual clusters with the merged cluster.
4. Repeat steps 2. and 3. until a single cluster is obtained (that is $p-1$ times).

Such AHC algorithm builds a *hierarchy*.

Definition 3. A hierarchy H of E is a set of non-empty classes that satisfy:

- $E \in H$
- For all $\mathbf{X}_{.i} \in E$, $\{\mathbf{X}_{.i}\} \in H$
- $\forall (A,B) \in H^2$, $A \cap B \in \{A, B, \emptyset\}$

In other words, a hierarchy H of E contains E and all its items. Moreover, two classes of H are either disjoint or one class contains the other.

In order to be able to represent a hierarchy structure graphically, we need to assess its levels, that is to assign them a height, by means of an *index*.

Definition 4. An *indexed hierarchy* is a pair (H, Ind) where H is a hierarchy and Ind is a function:

$$\mathrm{Ind} : H \to \mathbb{R}^+$$

such that:
$\forall (A, B) \in H^2$, such that $A \neq B$,

- $\mathrm{Ind}(A) = 0 \Leftrightarrow A$ is a singleton
- $A \subset B \Rightarrow \mathrm{Ind}(A) < \mathrm{Ind}(B)$.

An indexed hierarchy can be visualized using a graphic called *classification tree* or *dendrogram* (see Figure 3.2).

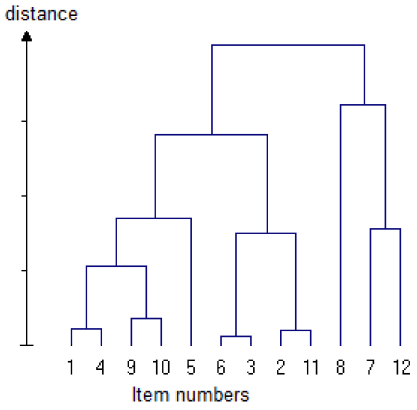


FIGURE 3.2: A dendrogram. *This figure is adapted from Fundamentals of Statistics (Lohninger 2010).*

The relationship generally used to define the index Ind of a hierarchy H is:

$$\forall (A, B) \in H^2, \mathrm{Ind}(A \cup B) = d(A, B),$$

that is the linkage criterion used to build the hierarchy. However, some linkage criteria do not allow to build an indexed hierarchy because they do not grow increasingly with the level of the hierarchy. For instance, the linkage measure defined by the Euclidean distance between the centers of the classes does not verify the second condition in Definition 4. Consequently, the dendrogram of the resulting hierarchy can present inversions (see Figure 3.3).

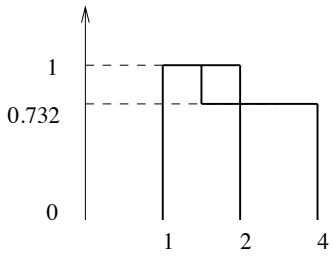


FIGURE 3.3: Example of an inversion in the dendrogram of a hierarchy.

It can be shown that the dendrograms of hierarchies built using the linkage criteria previously presented in this section never present inversions.

3.2.3 Determining the number of clusters

In the general context of data clustering, a major challenge is how to select the right number of clusters, also known as the model selection issue. Indeed, while in some situations the choice of the number of classes may be motivated by prior knowledge of the user, it is in general unknown and needs then to be estimated in a certain way.

Milligan & Cooper (1985) provide a comprehensive survey of 30 methods for estimating the number of clusters, while Gordon (1999) compares the performances of the best five rules exposed in Milligan & Cooper (1985). Gordon (1999) also divides these strategies of estimation of the number of clusters into *global methods* and *local methods*. Local methods are intended to test the hypothesis that a pair of clusters should be merged. While, with global approaches, some measure over the entire dataset is evaluated and optimized as a function of the number of clusters. More recently, Charrad et al. (2014) provide an implementation of an exhaustive list of indices to estimate the number of clusters in a dataset in the R package `NbClust`.

Several model selection approaches use the *within-group dispersion* measure, generally noted W, in order to estimate the number of clusters. Suppose the dataset E has already been clustered into G groups $C_1, \ldots, C_G$ of sizes $p_1, \ldots, p_G$, then the corresponding within-group dispersion measure W_G is defined as:

$$W_G = \sum_{g=1}^{G} \sum_{i=1}^{p_g} \delta(\mathbf{X}_{.i}, \mathbf{g}_{C_g})^2. \tag{3.3}$$

So if the dissimilarity δ is the Euclidean distance, then W_G corresponds to the pooled within-cluster sum of squares around the cluster means.

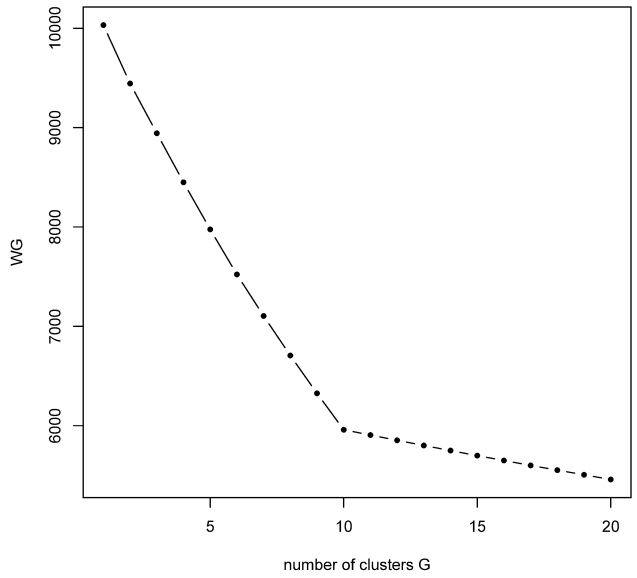


FIGURE 3.4: Observed within-group dispersion measures W_G versus the number of clusters G for a set E of $p = 200$ items in $\mathbb{R}^{100}$ clustered in 10 groups of size 20. The dissimilarity used is the Euclidean distance and the clustering method is the Ward's criterion.

Figure 3.4 shows a typical plot of within-group dispersion measures W_G, with the Euclidean distance and at each step of a Ward's clustering algorithm, as a function of the number of clusters G within the data set. We can notice that W_G decreases monotonically as the number of clusters G increases. Indeed, splitting a class results in two clusters that are more homogeneous and thus with lower dispersion. But this decrease gets slower from some G onward and the presence of such an elbow corresponds exactly to the "true" number of clusters $\hat{G}$ to be estimated (Sugar 1998, Sugar et al. 1999).

Among the global methods using the W_G measures and performing the best according to Milligan & Cooper (1985) was the Calinski and Harabasz index (Caliński & Harabasz 1974)

$$\mathrm{CH}(G) = \frac{(p-G)}{(G-1)} \frac{B_G}{W_G}.$$

$B_G = \sum_{g=1}^{G} \delta(\mathbf{g}_{C_g}, \mathbf{g}_E)^2$ is the between-cluster dispersion measure with g_E the center of gravity of the dataset E. $\mathrm{CH}(G)$ is only defined for G greater than 1 and the optimal number of clusters is $\hat{G}$ that maximizes $\mathrm{CH}(G)$.

Hartigan (1975) proposed the following statistic:

$$\mathrm{H}(G) = \frac{1}{p-G-1}\left(\frac{W_G}{W_{G+1}} - 1\right).$$

The idea is to start with $G = 1$ and to add a cluster if $\mathrm{H}(G+1)$ is significantly large. The decision rule suggested by Hartigan is to add a cluster if $\mathrm{H}(G) > 10$. Hence, the cluster number is best estimated as the smallest G such that $\mathrm{H}(G) \leq 10$. This estimate is defined for $G = 1$ and can then be used for testing the presence of a cluster structure within the dataset.

Another approach for estimating the number of clusters using the W_G measures is the proposal of Krzanowski & Lai (1988) which defined:

$$\mathrm{DIFF}(G) = (G-1)^{2/p} W_{G-1} - G^{2/p} W_G,$$

and chose $\hat{G}$ to maximize the quantity:

$$\mathrm{KL}(G) = \left|\frac{\mathrm{DIFF}(G)}{\mathrm{DIFF}(G+1)}\right|.$$

More recently, Tibshirani et al. (2001) proposed an approach to estimate the number of clusters in a dataset via the Gap statistic. This method is designed to be applicable to any clustering strategy and dissimilarity measure. The idea is to compare the within-cluster dispersion measure of the observed dataset to its expectation under an appropriate reference null distribution. To do so, they define:

$$\mathrm{Gap}(G) = E_p^{\star}[\log(W_G)] - \log(W_G),$$

where $E_p^{\star}$ denotes the expectation under a sample of size p from the reference null distribution. The optimal number of clusters $\hat{G}$ corresponds then to the value maximizing $\mathrm{Gap}(G)$.

The choice of an appropriate reference null distribution is important for applying the Gap statistic method. Tibshirani et al. have chosen the uniformity hypothesis to create the reference null distribution and considered two approaches to construct the support of such distribution. In the first approach, each reference variable $\mathbf{X}_{\cdot j}$, $1 \leq j \leq p$ is generated uniformly over the range of the observed values for that variable. In the second approach, the variables are sampled from a uniform distribution over a box aligned with the principal components of the centered design matrix $\mathbf{X}$. The uniformly generated design matrix is then back-transformed to obtain the reference dataset. In both strategies, the items of the reference dataset are generated independently.

Whereas the first approach has the advantage of simplicity, the second strategy may be more effective in recovering the underlying cluster structure since it takes into account the shape of the data distribution.

More particularly, the computational steps of the Gap method are the following:

1. For each number of clusters G, $1 \leq G \leq G_{max}$, compute the within-cluster dispersion measure W_G.

2. Generate B reference datasets in the way described above. Cluster each of the B reference datasets and calculate W_G^b for $b \in \{1, \ldots, B\}$ and $G \in \{1, \ldots, G_{max}\}$.

3. Compute the Gap statistic:

$$\text{Gap}(G) = \frac{1}{B} \sum_{b=1}^{B} \log(W_G^b) - \log(W_G).$$

4. The optimum number of clusters is given by the smallest G such that:

$$\text{Gap}(G) \geq \text{Gap}(G+1) - \text{sd}_{G+1}.$$

where sd_G is the normalized standard deviation of $\log(W_G^b)$:

$$\text{sd}_G = \sqrt{1 + \frac{1}{B}} \left(\frac{1}{B} \sum_{b=1}^{B} \left[\log(W_G^b) - \frac{1}{B} \sum_{b=1}^{B} \log(W_G^B) \right]^2 \right)^{1/2}. \tag{3.4}$$

3.3 The proposed LD block partitioning approach

3.3.1 The kernel trick

In machine learning, kernel methods are a class of algorithms for pattern analysis whose aim is to find and study general types of relations in datasets. Support Vector Machines (SVMs) are the most well known algorithms capable of operating with kernels (Schölkopf & Smola 2002).

Definition 5. A *positive definite kernel* (PDK) on the set E is a symmetric function $\mathbf{K} : E \times E \rightarrow \mathbb{R}$:

$$\forall (x, y) \in E^2,\ \mathbf{K}(x, y) = \mathbf{K}(y, x)$$

that satisfies, for all $N \in \mathbb{N}$, $(x_1, x_2, \ldots, x_N) \in E^N$ and $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j \mathbf{K}(x_i, x_j) \geq 0.$$

This definition is equivalent to that the similarity matrix of the items of E and defined by $\mathbf{K}$ is positive semi-definite.

Definition 6. If $\mathbf{K}$ is a positive definite kernel on a set E, then there is a Hilbert space $\mathcal{H}$ with the dot-product $\langle , \rangle_{\mathcal{H}}$ and a function $\Phi : E \to \mathcal{H}$ such as:

$$\forall (x, y) \in E^2,\ \mathbf{K}(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}\,. \tag{3.5}$$

The relationship 3.5, which allows to replace dot products by kernels' evaluations, is often called *kernel trick*. When looked more closely, Definition 6 implies that any algorithm which involves vectors and that is only expressed in terms of dot-products between these vectors can be performed implicitly in a Hilbert space by replacing each dot-product by an evaluation of a positive definite kernel on another space. In practice, the function Φ does not need to be specified and the main difficulty of the user is the choice of the kernel.

In the next two sections, we will show how the kernel trick can be used in the Ward's hierarchical clustering and the Gap statistic approaches. The steps of the proposed algorithm for inferring the LD blocks will be detailed in the last section of this chapter.

3.3.2 Ward's criterion using the LD kernel

Ward's hierarchical clustering using the Euclidean distance algorithm applied to SNP data falls within the framework of algorithms described above and the kernel trick can then be used. Indeed, let us consider a genotype matrix of p SNPs observed on n individuals, then $\mathbf{X} \in \mathbb{R}^{n \times p}$ with $\mathbf{X}_{ij} \in \{0, 1, 2\}$. Applying Ward's hierarchical clustering using the Euclidean distance to the columns $\mathbf{X}_{.j}$ of the genotype matrix requires the minimization of Ward's criterion (see Equation 3.1):

$$d_{\text{wl}}(A, B) = \frac{p_A p_B}{p_A + p_B} \|\mathbf{g}_A - \mathbf{g}_B\|_2^2.$$

As the main topic of the thesis is incorporating linkage disequilibrium information in GWA studies, it seems natural to use as kernel the pairwise LD measure:

$$\begin{aligned} \mathbf{K} : E \times E &\longrightarrow \mathbb{R} \\ (\mathbf{X}_{.i}, \mathbf{X}_{.j}) &\longmapsto \mathrm{ld}(\mathbf{X}_{.i}, \mathbf{X}_{.j}) \end{aligned}$$

The pairwise LD measure is indeed a PDK as it corresponds to the correlation between the indicators of the presence of the major allele (or minor allele) at the two loci (see Section 2.3.2).

Using the pairwise LD kernel, Ward's criterion can then be rewritten as follows:

$$\begin{aligned} d_{\mathrm{wl}}(A,B) &= \frac{p_A p_B}{p_A + p_B} \|\mathbf{g}_A - \mathbf{g}_B\|_2^2 \\ &= \frac{p_A p_B}{p_A + p_B} \langle \mathbf{g}_A - \mathbf{g}_B, \mathbf{g}_A - \mathbf{g}_B \rangle_{\mathcal{H}} \\ &= \frac{p_A p_B}{p_A + p_B} (\langle \mathbf{g}_A, \mathbf{g}_A \rangle_{\mathcal{H}} + \langle \mathbf{g}_B, \mathbf{g}_B \rangle_{\mathcal{H}} - 2 \langle \mathbf{g}_A, \mathbf{g}_B \rangle_{\mathcal{H}}). \end{aligned}$$

Moreover, we have:

$$\begin{aligned} \langle \mathbf{g}_A, \mathbf{g}_B \rangle_{\mathcal{H}} &= \left\langle \frac{1}{p_A} \sum_{i \in A} \Phi(\mathbf{X}_{.i}), \frac{1}{p_B} \sum_{j \in B} \Phi(\mathbf{X}_{.j}) \right\rangle_{\mathcal{H}} \\ &= \frac{1}{p_A p_B} \sum_{i \in A, j \in B} \langle \Phi(\mathbf{X}_{.i}), \Phi(\mathbf{X}_{.j}) \rangle_{\mathcal{H}} \\ &= \frac{1}{p_A p_B} \sum_{i \in A, j \in B} \mathbf{K}(\mathbf{X}_{.i}, \mathbf{X}_{.j}) \\ &= \frac{1}{p_A p_B} \sum_{i \in A, j \in B} \mathrm{ld}(\mathbf{X}_{.i}, \mathbf{X}_{.j}). \end{aligned}$$

The distance between the two classes can then be written as:

$$d(A,B) = \frac{p_A p_B}{p_A + p_B} \left(\frac{1}{p_A^2} S_{A,A} + \frac{1}{p_B^2} S_{B,B} - \frac{2}{p_A p_B} S_{A,B} \right), \tag{3.6}$$

using the notation:

$$S_{A,B} = \sum_{i \in A, j \in B} \mathrm{ld}(\mathbf{X}_{.i}, \mathbf{X}_{.j}).$$

Note that, in this case, the ld function can be any pairwise linkage disequilibrium measure such as D, D' or r^2.

3.3.3 The within-group dispersion measures using the LD kernel

Similarly to Ward's criterion, the within-group dispersion measure W should be rewritten using the LD kernel as it is the corner stone of the Gap statistic approach for model selection.

Assuming that the data has already been clustered into G classes $C_1, C_2, \ldots, C_G$ of sizes $p_1, p_2, \ldots, p_G$, then W_G can be written as follows:

$$\begin{aligned}
W_G &= \sum_{g=1}^{G}\sum_{j=1}^{p_g} \|\mathbf{X}_{.j} - \mathbf{g}_{C_g}\|_2^2 \\
&= \sum_{g=1}^{G}\sum_{j=1}^{p_g} \left\langle \mathbf{X}_{.j} - \mathbf{g}_{C_g}, \mathbf{X}_{.j} - \mathbf{g}_{C_g} \right\rangle_E \\
&= \sum_{g=1}^{G}\sum_{j=1}^{p_g} \langle \mathbf{X}_{.j}, \mathbf{X}_{.j} \rangle_E + \sum_{g=1}^{G}\sum_{j=1}^{p_g} \left\langle \mathbf{g}_{C_g}, \mathbf{g}_{C_g} \right\rangle_E - 2\sum_{g=1}^{G}\sum_{j=1}^{p_g} \left\langle \mathbf{X}_{.j}, \mathbf{g}_{C_g} \right\rangle_E \\
&= \sum_{g=1}^{G}\sum_{j=1}^{p_g} \langle \Phi(\mathbf{X}_{.j}), \Phi(\mathbf{X}_{.j}) \rangle_{\mathcal{H}} + \sum_{g=1}^{p_g} p_g \left\langle \frac{1}{p_g}\sum_{l=1}^{p_g} \Phi(\mathbf{X}_{.l}), \frac{1}{p_g}\sum_{k=1}^{p_g} \Phi(\mathbf{X}_{.k}) \right\rangle_{\mathcal{H}} \\
&\quad -2\sum_{g=1}^{G}\sum_{j=1}^{p_g} \left\langle \Phi(\mathbf{X}_{.j}), \frac{1}{p_g}\sum_{k=1}^{p_g} \Phi(\mathbf{X}_{.k}) \right\rangle_{\mathcal{H}} \\
&= \sum_{g=1}^{G} p_g + \sum_{g=1}^{G} \frac{1}{p_g} \sum_{j=1}^{p_g}\sum_{k=1}^{p_g} \mathrm{ld}(\mathbf{X}_{.j}, \mathbf{X}_{.k}) - 2\sum_{g=1}^{G} \frac{1}{p_g} \sum_{j=1}^{p_g}\sum_{k=1}^{p_g} \mathrm{ld}(\mathbf{X}_{.j}, \mathbf{X}_{.k}).
\end{aligned}$$

Then:

$$W_G = p - \sum_{g=1}^{G} \frac{1}{p_g} S_{C_g, C_g}. \tag{3.7}$$

Unlike the previous calculation of Ward's criterion, the ld function for assessing the W_G quantity can only be D' or r^2. Indeed, the calculations in Equation 3.7 implicitly require the condition $\mathrm{ld}(\mathbf{X}_{.j}, \mathbf{X}_{.j}) = 1$.

3.3.4 The algorithms

The first step of the proposed partitioning algorithm slightly differs from a classical hierarchical clustering. Indeed, in our context, nearby SNPs (in the sense of the physical distance along the genome) show relatively high LD between them (see Figure 3.1). For this reason, Ward's hierarchical clustering used for grouping the markers should take into account this spatial constraint. In other words, at each step of the clustering algorithm, only two contiguous SNPs/clusters of

SNPs are allowed to be merged. The sketch of the Ward's constrained hierarchical clustering method is presented in Algorithm 1.

More formally, given a genotype matrix $\mathbf{X}$ and a LD similarity Sim, we begin with each of the p SNPs in a separate cluster (line 2). The corresponding distances between these singletons are the $p-1$ dissimilarities between the SNPs (line 3). Then, the two closest **adjacent** clusters (in the sense of the Ward's linkage criterion) are repeatedly merged until all SNPs are members of the same cluster. The main two steps in the algorithm are finding the best fusion corresponding to the pair of clusters with the smallest distance (line 5) and assessing the 2 distances with newly-formed cluster (lines 7 and 8). The objects returned by the `cWard_LD` algorithm are the classification tree T summarizing the clustering process and the within-group dispersion measures W_{vect} at each step of the clustering. These last measures will be used as input at the second step of the proposed approach for inferring the LD blocks : the Gap statistic.

Algorithm 1 The Ward's constrained hierarchical clustering algorithm using the LD similarity

1: **procedure** CWARD_LD($\mathbf{X} \in \{0,1,2\}^{n\times p}$, Sim)
2: $\mathcal{C} \leftarrow \{C_i = \{\mathbf{X}_{.i}\}, i \in 1,\ldots,p\}$ ▷ each SNP in separate cluster
3: $D \leftarrow \{1 - \mathrm{Sim}(\mathbf{X}_{.i}, \mathbf{X}_{.(i+1)}), i \in 1,\ldots,p-1\}$ ▷ the $(p-1)$ vector of dissimilarities ▷ between adjacent SNPs
4: **for** $step = 1$ **to** $p-1$ **do**
5: $i^\star \leftarrow \arg\min_{i\in\{1,\ldots,p-step\}} D(C_i, C_{i+1})$ ▷ find the closest pair of ▷ adjacent clusters
6: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_{i^\star}, C_{i^\star+1}\} \cup \{C_{i^\star} \cup C_{i^\star+1}\}$ ▷ updating the clustering
7: $d_1 \leftarrow D(C_{i^\star-1}, C_{i^\star} \cup C_{i^\star+1})$ ▷ using Equation 3.6
8: $d_2 \leftarrow D(C_{i^\star} \cup C_{i^\star+1}, C_{i^\star+2})$
9: $D \leftarrow D \setminus \{D(C_{i^\star-1}, C_{i^\star}), D(C_{i^\star}, C_{i^\star+1})\} \cup \{d_1, d_2\}$ ▷ updating D
10: $W_{\text{vect}}[step] \leftarrow W_{p-step}$ ▷ using Equation 3.7
11: **end for**
12: $cW.T \leftarrow T$ ▷ the clustering tree
13: $cW.W_{\text{vect}} \leftarrow W_{\text{vect}}$ ▷ the vector of within-group dispersion measures
14: **return** cW
15: **end procedure**

The Gap statistic approach starts by computing the within-group dispersion measures of the observed genotype matrix (line 2) for each number of clusters G within the interval $[\mathrm{minG}, \mathrm{maxG}]$, and then compares it to its expectation under an appropriate null reference distribution obtained by applying the `cWard_LD` algorithm to uniformly simulated genotype matrices (line 6). After assessing the quantity $\mathrm{Gap}(G)$ for each $G \in [\mathrm{minG}, \mathrm{maxG}]$ (line 10), the `GapStatistic` algorithm returns the optimal number of clusters $\hat{G}$ in the sense of the Gap statistic approach.

Algorithm 2 The Gap statistic algorithm

procedure GAPSTATISTIC($\mathbf{X} \in \{0,1,2\}^{n \times p}$, Sim, minG, maxG, B)
  $W_{\text{obs}} \leftarrow \texttt{cWard\_LD}(\mathbf{X}, \text{Sim}).W_{\text{vect}}[\text{minG} : \text{maxG}]$
  for $G = \text{minG}$ **to** maxG **do**
    for $b = 1$ **to** B **do**
      generate uniformly $\mathbf{X}_{\text{sim}} \in \mathbb{R}^{n \times p}$ ▷ each SNP independently
      $W_G^b \leftarrow \texttt{cWard\_LD}(\mathbf{X}_{\text{sim}}, \text{Sim}).W_{\text{vect}}[\text{minG} : \text{maxG}]$
    end for
    $\text{Gap}(G) \leftarrow \frac{1}{B} \sum_{b=1}^{B} \log(W_G^b) - \log(W_G)$
  end for
  Choose $\hat{G}$ as the smallest G such as
  $\text{Gap}(G) \geq \text{Gap}(G+1) - \text{sd}_{G+1}$ ▷ using Equation 3.4
  return $\hat{G}$
end procedure

3.4 Conclusions

The detection of linkage disequilibrium blocks in the human genome is a recent research field and the methods for their definition are still under development. In this chapter, we have shown how existing block partitioning algorithms are mainly based on two approaches: either the pairwise LD measures are used to detect regions of high or little historical recombination or blocks are defined by employing a haplotypic diversity criterion, where a small number of haplotypes account for a high proportion of the observations. The proposed block partitioning method consists in (i) performing a spatially-constrained hierarchical clustering using the Ward's linkage criterion and the LD similarity (ii) applying the Gap statistic approach to the obtained hierarchy to estimate the number of groups.

Applications of these haplotype blocks detection methods include at least three contexts differing in their objectives. First, the definition of htSNPs in haplotype blocks can reduce genotyping efforts in GWA studies, while much of genetic variation within these blocks is summarized by the htSNPs. Second, grouping markers on the basis of prior biological knowledge can improve the interpretability of the results in medical studies. Third, incorporating the block structure of the SNPs can improve the power of association studies by considering blocks of markers in LD rather than a single SNP at a time. The next chapter falls in the latter category. Indeed, it illustrates how the proposed LD block partitioning approach can be used to incorporate the LD information for efficient marker selection in GWA studies. In chapter 5, a generalized and efficient implementation of the `cWard_LD` algorithm will be presented.

Chapter 4

Performance of a blockwise approach in variable selection using linkage disequilibrium information

4.1 Introduction

With recent advances in high-throughput genotyping technology, genome-wide association studies (GWAS) have become a tool of choice for identifying genetic markers underlying a variation in a given phenotype – typically complex human diseases and traits. In GWAS, information on genetic polymorphisms is collected across the genome and single nucleotide polymorphisms (SNPs) are typically used due to their abundance in the genome. However, common genetic variants identified by GWAS only account for a relatively small proportion of the heritability of diseases (Manolio et al. 2009).

The most widely used approach for selecting causal SNPs is to perform univariate tests of association between the phenotype of interest and the genotype of each marker (Burton et al. 2007, Sham & Purcell 2014). Following Yi et al. (2015), this type of approach will be referred to as Single Marker Analysis (SMA). The results of SMA are often refined in two-ways. First, due to linkage disequilibrium (LD) between SNPs, combining the p-values obtained by SMA into gene-level statistics may yield more interpretable results (Li et al. 2011). Second, candidate markers selected by SMA may be incorporated into a *multi-variable linear models of*

association. In the field of feature-subset selection, the sequential forward approach is a standard. Starting from an empty set, this greedy search algorithm sequentially adds features that maximize a given objective function when combined with features that have already been selected. This method remains however not widely used for identifying a set of associated genetic markers. Conversely, recent studies suggest that penalized regression methods such as Lasso (Tibshirani 1996) and Elastic-Net (Zou & Hastie 2005) may be appropriate to identify the additive effect of several genetic markers (Abraham et al. 2013, Waldmann et al. 2013, de Maturana et al. 2014, Yi et al. 2015). Such methods allow multi-variable linear models to be estimated in high-dimensional situations such as GWAS, where the number p of variables (i.e., SNP markers) exceeds the number n of observations (i.e., individuals) by several orders of magnitude. In this chapter, we propose a penalized regression approach tailored to the dependence between markers in GWAS induced by linkage disequilibrium (LD). Our goal is to identify common variants which may have been missed by SMA because their individual effect size is not large enough to pass genome-wide significance thresholds.

As a motivating example for our contribution, the LD (r^2 coefficients, upper triangular part) and the sample genotype correlations (lower triangular part) between the first 256 SNPs of chromosome 6 in a study on 605 HIV-infected patients are represented in Figure 4.1 (Dalmasso et al. 2008). A blockwise structure can be distinguished, where the average LD within blocks of 12 to 15 SNPs is approximately $r^2 = 0.2$. The LD values are notably more contrasted than the correlation values, as many r^2 coefficients are very close to 0. In order to account for, and take advantage of this strong dependency structure between adjacent or nearby SNPs, it makes sense to focus on the scale of LD blocks, and to explicitly look for *sets of LD blocks jointly associated to the phenotype of interest*.

In order to do so, we propose a three-step method which consists in (i) inferring groups of SNP – that is, LD blocks– using a spatially-constrained hierarchical clustering algorithm, (ii) applying a model selection approach to estimate the number of groups, and (iii) identifying associated groups of SNPs using a Group Lasso regression model (Yuan & Lin 2005). This approach is described in Section 4.2.1. Sections 4.2.2 to 4.2.4 cover a description of the competing approaches the evaluation methods used for performance assessment. In Sections 4.3.1 and 4.3.2, the proposed method is compared to state-of-the-art competitors on simulated and semi-simulated data. Section 4.3.3 describes the application of the proposed method on microarray data from a specific GWA study on HIV.

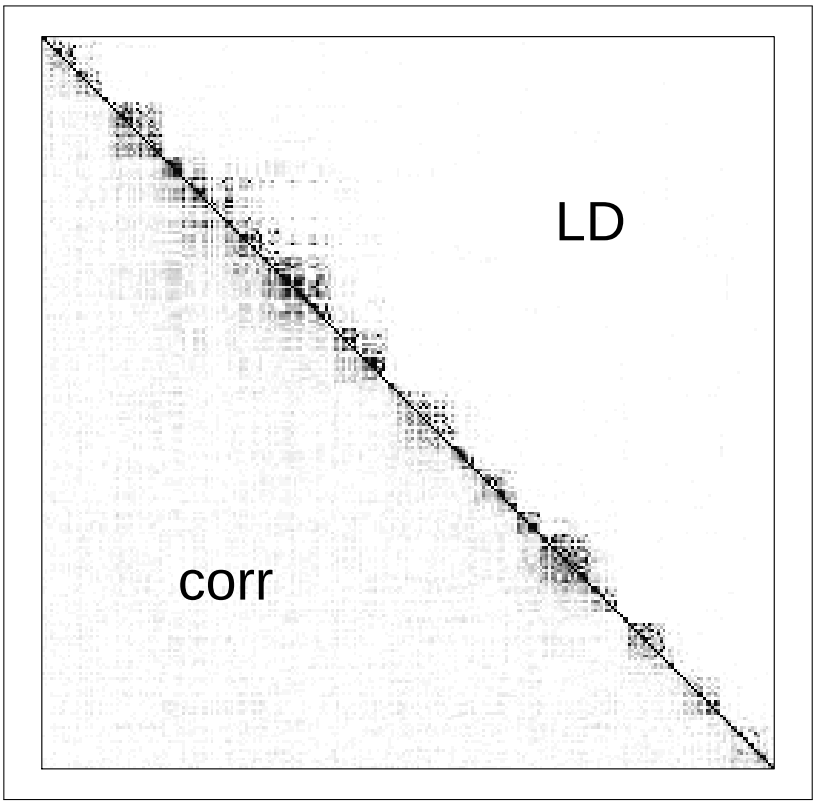


FIGURE 4.1: Blockwise dependency in real genotyping data: 256 SNPs spanning the first 1.45 Mb of Chromosome 6 in Dalmasso et al. (2008). The average distance between two successive SNPs is approximately 5 kb. The upper triangular part of the matrix displays measures of LD (r^2 coefficients) between pairs of SNPs, while its lower triangular part displays absolute sample correlations between pairs of SNP genotypes. Colors range linearly from 0 (white) to 0.4 (black).

4.2 Methods

4.2.1 A three-step method for GWAS

The problem of selecting causal SNPs can be cast as a problem of high-dimensional variable selection. We consider the problem of predicting a continuous response $\mathbf{y} \in \mathbb{R}^n$ from covariates $\mathbf{X} \in \mathbb{R}^{n \times p}$. For $i \in \{1, \ldots, n\}$, $\mathbf{X}_{i\cdot}$ is a p-dimensional vector of covariates for observation i and for $j \in \{1, \ldots, p\}$, $\mathbf{X}_{\cdot j}$ is a n-dimensional vector of observations for covariate j. In GWAS, the covariates are ordinal and correspond to SNP genotypes: $X_{ij} \in \{0, 1, 2\}$ correspond to the number of minor alleles at locus j for observation i. For each $i \in \{1, \ldots, n\}$, we assume that $\mathbf{X}_{i\cdot}$ has a block structure with G non-overlapping blocks of sizes $p_1, \ldots p_G$, with $\sum_{g=1}^{G} p_g = p$. Thus $\mathbf{X}_{i\cdot} = (\mathbf{X}_{i\cdot}^1, \ldots, \mathbf{X}_{i\cdot}^G)$ with $\mathbf{X}_{i\cdot}^g \in \mathbb{R}^{p_g}$ for $g = 1, \ldots, G$.

We propose a three-step method consisting in (i) performing a spatially constrained hierarchical clustering of the covariates $\mathbf{X}$, (ii) estimating the number of groups using (a modified version

of) the Gap statistic (Tibshirani et al. 2001), and (iii) performing a Group Lasso regression to identify which of the inferred groups are associated with the response $\mathbf{y}$.

4.2.1.1 Inference of LD blocks from genotypes

This first step has been presented in details in Section 3.3. It consists in inferring LD blocks using a spatially constrained hierarchical clustering algorithm. Only the genotype data $\mathbf{X}$ are used at this step.

The proposed clustering procedure is based on the one of the most widely used methods for cluster analysis: Ward's incremental sum of squares algorithm (Ward Jr 1963). Nevertheless, it differs from it in two aspects. First, instead of the standard Euclidean distance, we use a measure of the dissimilarity between two SNPs j and j' based on LD: $1 - r^2(j, j')$. Second, we take advantage of the fact that the LD matrix can be modeled as block-diagonal (see Figure 4.1) by only allowing groups of variables that are *adjacent on the genome* to be merged.

4.2.1.2 Estimation of the number of groups

This second step of applying the Gap statistic has been presented in details in Section 3.3.

We have chosen to use a modified version of the Gap statistic (Tibshirani et al. 2001) as a model selection criterion. The Gap statistic compares W_G to its expectation under an appropriate reference null distribution of the data. For a clustering into G groups, we calculate the following quantity:

$$Gap^\star(G) = \frac{1}{B}\sum_{b=1}^{B}\left(W_G^b - W_G\right), \tag{4.1}$$

where for $b = 1 \ldots B$, W_G^b denotes the within-cluster dispersion of clustering the reference data set b in G groups.

In the classical version of the Gap statistic (see Section 3.2.3), the logarithm of W_G is used instead of W_G, and several alternatives to this original definition have been investigated recently (Mohajer et al. 2011). We decided to use the definition in Equation 4.1 as we noticed that it led to better estimation of the number of groups in our simulation studies, which were performed

under a variety of parameters and on several data sets. For the reference distribution, we followed the initial strategy proposed in the original Gap statistic paper (Tibshirani et al. 2001) and simulated each reference feature according to a uniform distribution over the discrete set $\{0, 1, 2\}$. We chose to simulate $B = 100$ reference samples since we empirically observed that it was sufficient to provide a stable estimation of the number of groups.

4.2.1.3 Selection of groups associated with the response

Once LD blocks have been identified, we use Group Lasso regression (Yuan & Lin 2005), presented in Section 2.1.2.3, to identify blocks associated with the phenotype. In the context of GWAS, the explanatory variables are then the SNP markers and the response is the phenotype of interest.

As discussed earlier, the Group Lasso is a group selection method: by construction, the estimated coefficients within a group tend to be either all zero or all nonzero. In practice, the columns of the design matrix $\mathbf{X}$ are scaled before performing Group Lasso regression.

4.2.2 Competing methods

Various approaches have been proposed to select causal SNPs from GWAS data. The method described in Section 4.2.1 is compared to two groups of methods:

- three methods that do not explicitly take a block-structure information into account: SMA, and two penalized regression approaches: Lasso (Tibshirani 1996) and Elastic-Net (Zou & Hastie 2005).
- two methods that do explicitly take the block-structure information into account: the haplotype association module of the PLINK genome association analysis tool (Purcell et al. 2007), and the Group Lasso applied to the true SNP groups. The latter approach cannot be applied in practice, but is very useful to analyze the contribution of the different steps of the proposed method. We will refer to this approach as the "oracle Group Lasso".

Single Marker Analysis. In the standard SMA , for each variable $\mathbf{X}_{.j}$, we fit a single-predictor equation $\mathbf{y} = \beta_0 + \beta_j \mathbf{X}_{.j}$ and a p-value from a t-test against an intercept-only model is calculated.

Multi-variable approaches. The Lasso (Tibshirani 1996) and Elastic-Net (Zou & Hastie 2005) regression models (see Section 2.1.2.3) were compared to the proposed approach. As seen previously, the Lasso encourages sparsity by setting many regression coefficients for irrelevant SNPs to exactly zero. However, this method tends to select only one variable in each group of correlated variables. The estimator of Elastic-Net incorporates some prior information regarding the block structure of the data. However, unlike the proposed method, it does not take advantage of the fact that in the particular case of GWAS, LD blocks are adjacent along the genome. In this chapter, we chose a large value for the ridge parameter ($\lambda_2 = 0.8$) in order for the Elastic-Net estimate to be substantially different from the Lasso estimate (which corresponds to $\lambda_2 = 0$).

Haplotype association. This competing grouping strategy includes 4 steps, the first 3 being performed using the PLINK genome association analysis tool. The first step consists in inferring the LD blocks following the confidence intervals procedure (Gabriel et al. 2002). Then within each LD block, haplotypes are estimated using an accelerated EM algorithm similar to the partition/ligation method (Qin et al. 2002). In the third step, haplotype-specific tests (with 1 degree of freedom) for a quantitative trait are performed with PLINK using the option *–hap-assoc*. Finally, we define a block-adjusted p-value by performing a (Bonferroni) Family-Wise Error Rate correction within each block. The p-value of a SNP is then defined as the adjusted p-value of the block it belongs to.

4.2.3 Performance evaluation

Our performance assessment aims at evaluating the ability of our proposed method to retrieve causal SNPs. Performance is evaluated using partial Areas Under the Curve (AUC) of the Receiver Operator Characteristics (ROC) curve. This measure will be denoted by pAUC. We first evaluate, for each method, the True Positive Rate (TPR) and False Positive Rate (FPR) for a grid of underlying regularization parameter values and for each simulation in order to obtain a ROC curve. Then we calculate the pAUC in the range FPR $\in [0, lim]$ for each ROC curve, where lim is defined as the maximum value of FPR below which the ROC coordinates of all methods are well defined.

4.2.4 SNP and block-level evaluation

A SNP may be detected by a given method either because it is a causal SNP, that is truly associated with the phenotype, or because it is in LD with such a causal SNP. This issue is intrinsic to the design of GWAS and thus requires adapted definitions of true and false positives. A relevant recent contribution is the recently-introduced notion of "threshold-specific FDR" (tFDR) (Yi et al. 2015). tFDR relies on an alternate definition of true positives that incorporates not only "causal true positives" but also "linked true positives". In a similar spirit, we consider two definitions of associated SNPs in our simulation setting. We define a *causal SNP* as a SNP that is simulated with a non-zero regression parameter, and a *block-associated SNP* as a predictor that is not a causal marker but simulated in the same LD block that a causal SNP. This is illustrated by Figure 4.2. Importantly, and contrary to tFDR, our definition of a block-associated SNP does not depend on a correlation threshold.

FIGURE 4.2: Schematics of covariance matrices for illustration of the proposed definition of "causal SNPs" (red area in the left panel) and "block-associated SNPs" (red area in the right panel) on a toy example with $p = 12$ SNPs in 3 blocks of size 4, 6, and 2, respectively.

Therefore, we consider two types of evaluation differing in their objective. In the SNP-level evaluation (left panel in Figure 4.2), the statistical unit considered is the SNP, and a true positive (in red) is the discovery of a causal SNP; the discovery of any other SNP (in blue) is considered as a false positive. In the block-level evaluation (right panel in Figure 4.2), the statistical unit considered is the LD block, and a true positive (in red) is the discovery of a block-associated SNP; the discovery of any other SNP (in blue) is considered as a false positive. Given these definitions, we expect better results from the three classical approaches (SMA, Lasso, and Elastic-Net) for the SNP-level evaluation, and better results from the group-based methods for the block-level evaluation.

4.2.5 Simulation settings

Our simulation setting is adapted from Wu et al. (2009). For all $i \in \{1, \ldots, n\}, \mathbf{X}_{i\cdot}$ is generated from a p-dimensional multivariate normal distribution whose covariance matrix is block-diagonal. If $j \neq j'$ are in the same group, $cov(\mathbf{X}_{\cdot j}, \mathbf{X}_{\cdot j'}) = \rho$ else $cov(\mathbf{X}_{\cdot j}, \mathbf{X}_{\cdot j'}) = 0$. Then, we set X_{ij} to $0, 1$ or 2 according to whether $X_{ij} < -c$, $-c \leq X_{ij} \leq c$ or $X_{ij} > c$, where c is a threshold determined for producing a given minor allele frequency. For example, choosing c as the first quartile of a standard normal distribution corresponds to fixing the minor allele frequency of the corresponding SNP to 0.5. The associated continuous phenotype vector is finally generated according to the linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is a gaussian error term.

4.3 Results

4.3.1 Results on simulated data

We set $n = 100$ and $p = 2,048$, with 192 groups of sizes $2, 2, 4, 8, 16$, and 32, replicated 32 times. The ordering of the groups is drawn at random for each simulation. Figure 4.3 illustrates the type of dependency structure that is obtained in this setting, using the same type of representation as in Figure 4.1.

In our simulation, the difficulty of the problem is calibrated according to the coefficient of determination R^2 of the model, that is, the ratio of the variance explained by the model to the total variance. This coefficient quantifies the ability of a multi-variable model to explain the phenotype using the combined effect of all the relevant markers. It is also called the total heritability h^2 in the context of genetics (Yi et al. 2015). This coefficient is not to be mistaken with the squared Pearson linear correlation coefficient r^2 between the phenotype and the genotypes of a single marker. Thus, in our simulation setting, the absolute value of the regression coefficients of causal SNPs does not influence the performance of the methods. In the experiments reported below, the regression coefficients of the causal SNPs were randomly set to 1 or -1, and to 0 for all other SNPs; R^2 is set to 0.2, which appeared to be a realistic value for GWA studies in

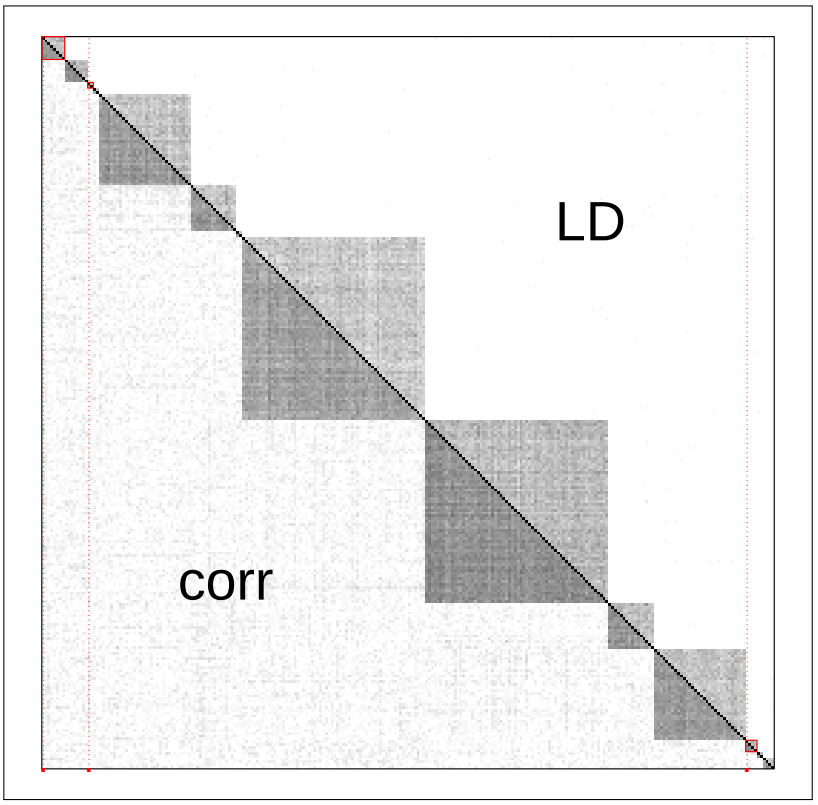


FIGURE 4.3: Blockwise dependency for a simulation run, with $\rho = 0.4$, using the same representation and color scale as in Figure 4.1. The average r^2 within LD block is approximately 0.2. Red dots correspond to causal SNPs. The blocks in which they are located are highlighted by red squares.

comparison with the number of individuals $n = 100$. The other parameters of the simulation are the within-LD-block correlation coefficient ρ, the number `causalSNP` of causal SNPs and the size `sigBlock` of the associated block.

We have performed an extensive simulation study, where `causalSNP` $\in \{1, 2, 4, 6, 8\}$ and `sigBlock` $\in \{2, 4, 8, 16, 32\}$. We report average pAUC across 300 simulation runs. We mainly focus on cases where the correlation coefficient $\rho \in \{0.2, 0.4\}$ as these values yield an average LD within a block that is consistent with what is typically observed in real data (see Figure 4.1).

4.3.1.1 Block-level versus SNP-level evaluation

We consider a setting where a single SNP is truly associated with the phenotype. Figure 4.4 displays the pAUC versus the size `sigBlock` of the "associated block" (that is, the LD block

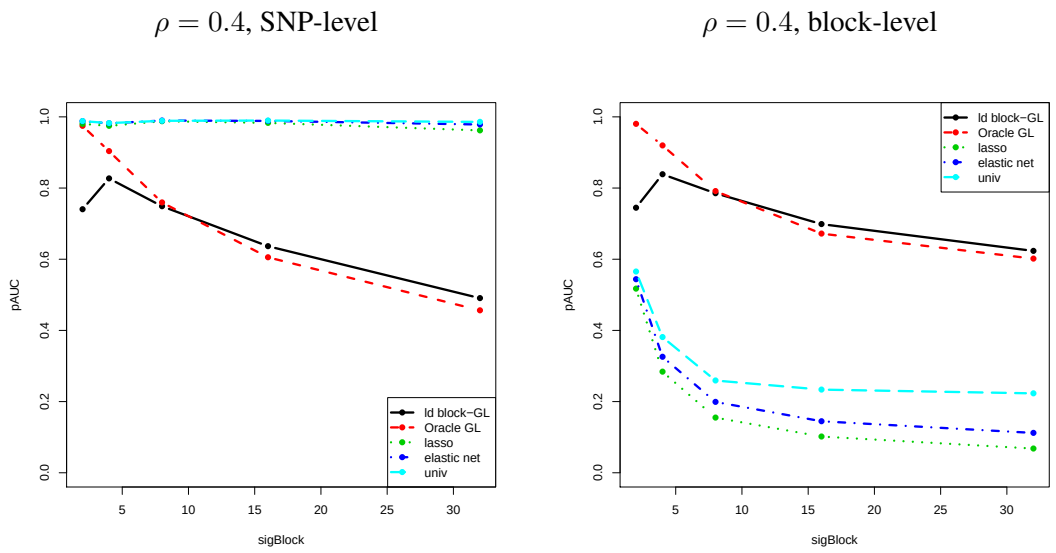


FIGURE 4.4: The mean pAUC versus the size of the LD block containing a single causal SNP `sigBlock` for the proposed method ("ld block-GL", black solid lines), oracle Group Lasso (dashed red lines), Lasso (dotted green lines), Elastic-Net (dash-dotted blue lines) and SMA ("univ", dashed light blue lines), for $\rho = 0.4$. Left: SNP-level evaluation. Right: block-level evaluation.

containing the causal SNP) for both SNP- and block-level evaluations. With SNP-level evaluation (left panel), group-based approaches are outperformed by the three competitors, and increasingly so as the size of the associated block increases. This is mainly due to the high number of false positive SNPs generated by the group selection. Indeed, selecting a group with only one causal SNP causes all other SNPs of the group to be declared as false positives. Conversely, with group-level evaluation (right panel of Figure 4.4), group-based methods show a clear superiority, showing that multi-variable SNP-based methods (Lasso or Elastic-Net) are generally unable to select all of the causal SNPs due to the presence of correlation between the SNPs of the block. The poor performance of Lasso under correlated designs is not new (Zou & Hastie 2005), but Figure 4.4 suggests that the proposed approach even outperforms Elastic-Net. Although the Elastic-Net has been designed specifically for correlated designs and has recently been shown to have good performance in GWAS (Yi et al. 2015), it seems that it does not take full advantage of the characteristic block structure of the predictors in GWAS.

As the size of the associated block increases, the performance of all methods decrease. Indeed, for a given level of within-block correlation (here, $\rho = 0.4$), the larger the size of the block, the more diluted the information about the causal SNP becomes. Thus, a larger LD block in our simulation setting results in a more difficult problem. This increase in complexity explains the general decrease in performance. This decrease in performance is more severe for the Group

lasso. Indeed, it tends to select small groups of SNPs because its default penalty increases with block size. The drop in performance of the proposed approach compared to that of the "oracle" Group Lasso for $\texttt{sigBlock} \in \{2, 4\}$ is discussed in the next subsection when assessing the efficiency of the block inference step.

In the remainder of this section, we focus on SNP-level evaluation, which is *a priori* more favorable to SNP selection methods than to group selection methods. We are interested in comparing the methods under this evaluation setting which is particularly challenging for the proposed approach.

4.3.1.2 Efficiency of LD block inference

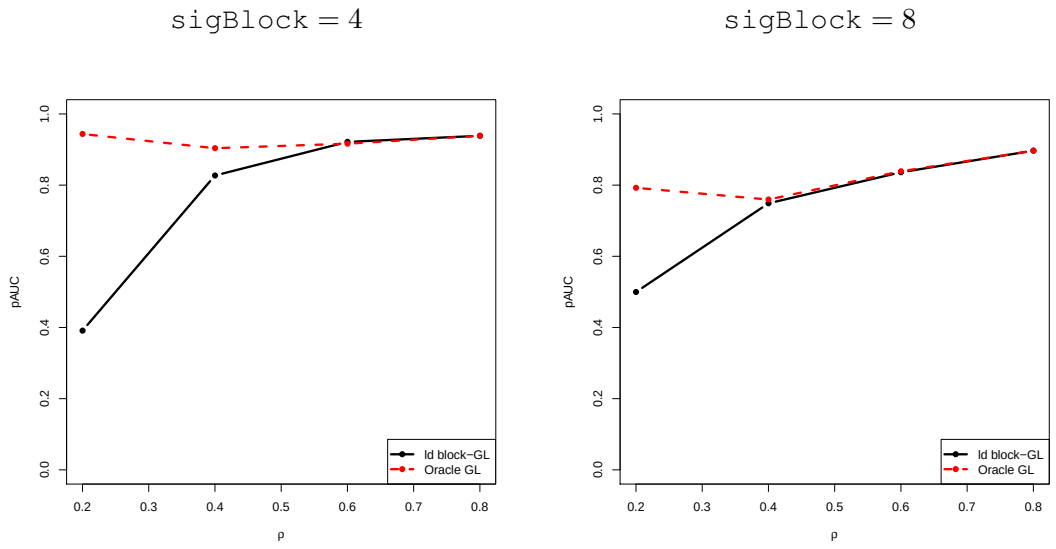


FIGURE 4.5: Average pAUC versus correlation level ρ for the proposed method ("ld block-GL", black solid lines) and an oracle version where the LD blocks are assumed to be known (dashed red lines), for $\texttt{sigBlock} \in \{4, 8\}$.

The goal of this section is to quantify the inference of the LD blocks (the first two steps in Section 4.2.1) on the global performance of the proposed method. In order to do so, we compare the performance of the proposed method to that of the "oracle" version where the Group Lasso is applied to the true LD blocks, that is, those defined by the simulation settings. Figure 4.5 displays the mean pAUC versus the correlation level for both methods. When the level of correlation is less than 0.4, we note that the proposed approach is outperformed by the "oracle" Group Lasso. In fact, for low correlation levels, the block inference procedure tends to under-estimate the number of blocks leading to an estimated group structure with big blocks and thus a high number of false positives selected by the Group Lasso. However, the difference

between the performance of the two group-based methods becomes insignificant when the level of correlation is above 0.4 and when the size of the associated block is greater than 4. This indicates that the proposed LD block inference method, which combines constrained clustering and model selection, efficiently captures the underlying dependency structure in this case.

4.3.1.3 Influence of the number of causal SNPs per block

$$\rho = 0.4$$

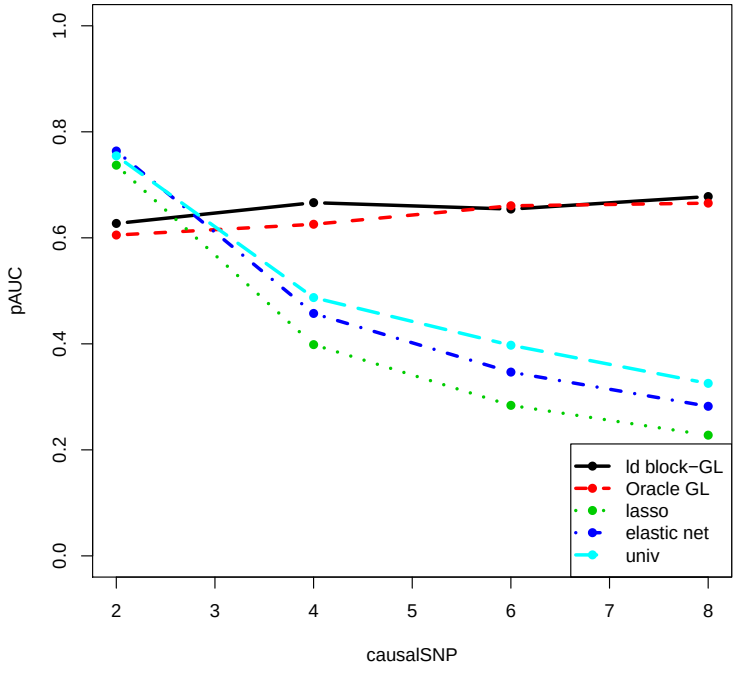


FIGURE 4.6: The mean pAUC as a function of the number of causal SNPs `causalSNP` within a block of size 8, for the proposed method ("ld block-GL", black solid lines), oracle Group Lasso (dashed red lines), Lasso (dotted green lines), Elastic-Net (dash-dotted blue lines) and SMA ("univ", dashed light blue lines), for $\rho = 0.4$.

We investigate the robustness of the 5 approaches to the parameter `causalSNP`, that is, the number of relevant variables within a block of size 8. Figure 4.6 displays the pAUC as a function of `causalSNP` for $\rho = 0.4$.

These results illustrate the robustness of the proposed group approach to an increasing number of causal SNPs, which is not the case of its 3 competitors. Indeed, the performance of the group strategies remain constant when that of the classical approaches deteriorate significantly as soon as the number of relevant SNP within the block exceeds 2. More specifically, the Group Lasso selects the associated block of 8 SNPs for both correlation levels. On the contrary, the

Lasso fails to recover the true relevant SNPs if there are correlations among the variables. As expected, the Elastic-Net performs a little better than the Lasso when the correlation structure is strong enough for the grouping effect of this model to be effective ($\rho \geq 0.4$).

4.3.1.4 Influence of the Minor Allele Frequency distribution

Our simulation model adapted from Wu et al. (2009) allows to reproduce the group-structured correlation that characterizes the GWAS data (see Figure 4.3). However, as noted by a reviewer, fixing the cutoff parameter c at the first quantile of the standard normal distribution as in Wu et al. (2009) generates unrealistic Minor Allele Frequency (MAF) distributions. To address this point, we simulated genotype matrices where the MAF of the SNPs are uniformly sampled between 0.05 and 0.5. This roughly corresponds to the MAF distribution observed in a real GWA study (Dalmasso et al. 2008), and MAF$= 0.05$ is a commonly-used threshold to partition variants into rare and common.

We then performed the same simulation study presented above adapting the dimension parameters to the new range of MAF. Specifically, we used $n = 1,000$ in order for variants with a low MAF to be observed frequently enough. Accordingly, the R^2 ratio was lowered to 0.01 in order for the difficulty of the problem to be similar. The number of markers was increased to $p = 4,096$ in order to maintain $p \gg n$. Finally, groups of sizes $2, 2, 4, 8, 16$, and 32 were replicated 64 times, yielding a total of 384 groups.

The results are shown in Figures 4.7 and 4.8 and conclusions are almost identical to those of the previous subsections. Firstly, for the scenario with an isolated causal SNP as in Section 4.3.1.1 and for the scenario with an increasing number of causal markers as in Section 4.3.1.3, the ordering of the performance of all the methods remained unchanged with a general increase for all the approaches due to the less stringent high-dimensionality ratio n/p compared to the ratio used in the previous subsections. Secondly, the first two steps of the proposed approach were able to perfectly retrieve the underlying block structure, even with low values of the correlation. In contrast, performance curves in scenario 4.3.1.2 were superimposed only for $\rho \geq 0.4$. This difference can be explained by the fact that increasing the number of individuals n led to a more salient LD block structure.

$\rho = 0.4$, SNP-level $\rho = 0.4$, block-level

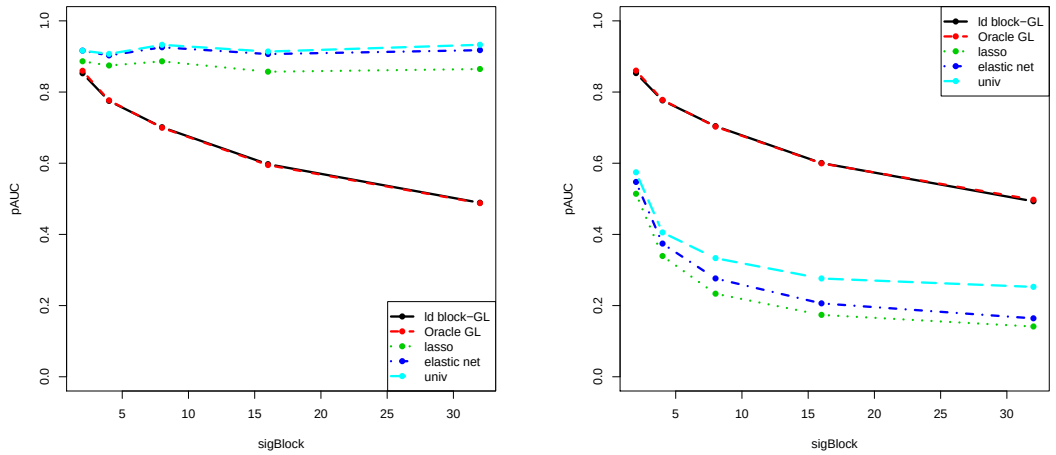


FIGURE 4.7: The mean pAUC versus the size of the LD block containing a single causal SNP `sigBlock` for the proposed method ("ld block-GL", black solid lines), oracle Group Lasso (dashed red lines), Lasso (dotted green lines), Elastic-Net (dash-dotted blue lines) and SMA ("univ", dashed light blue lines), for $\rho = 0.4$. Left: SNP-level evaluation. Right: block-level evaluation.

$\rho = 0.4$

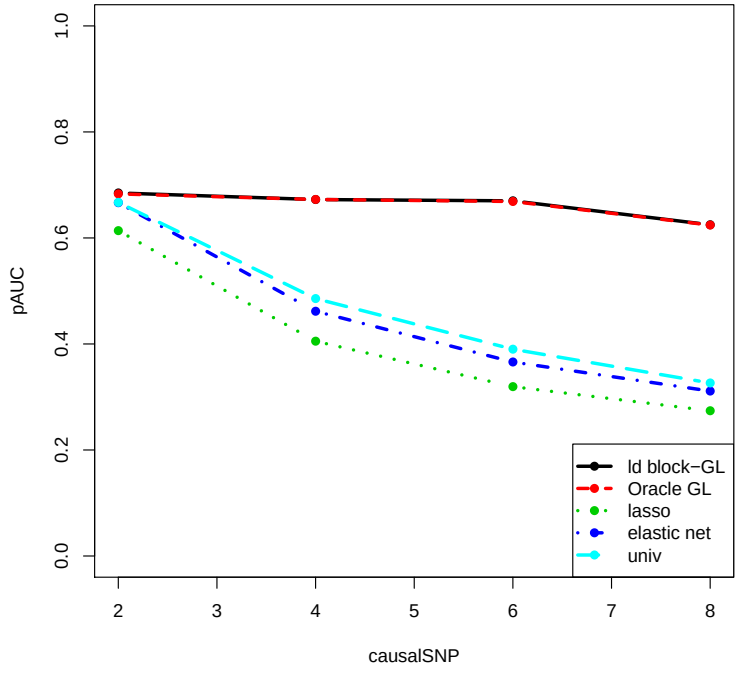


FIGURE 4.8: The mean pAUC as a function of the number of causal SNPs `causalSNP` within a block of size 8, for the proposed method ("ld block-GL", black solid lines), oracle Group Lasso (dashed red lines), Lasso (dotted green lines), Elastic-Net (dash-dotted blue lines) and SMA ("univ", dashed light blue lines), for $\rho = 0.4$.

4.3.2 Results on semi-simulated data

In order to control the causal SNPs while considering a realistic dependance structure among the SNPs, we used semi-simulated data, where the genotypes come from a real GWA study and the phenotypes are simulated using the linear model presented in Section 4.2.5 with pre-determined causal SNPs. This type of simulation allows to study a data set with a real linkage disequilibrium structure while having a ground truth. The genotype data correspond to the first $p = 2,048$ SNPs of chromosome 22 for $n = 100$ individuals from a GWA study on HIV (Dalmasso et al. 2008). This data set is described in more detail in Section 4.3.3. The LD block structure was firstly inferred using the first steps of the two group-based approaches:

- CHC-Gap : the proposed constrained hierarchical clustering followed by the Gap statistic.
- CI : the default confidence intervals method used in PLINK.

The procedure CHC-Gap estimated 225 blocks and the procedure CI inferred 993 blocks including 555 blocks of size 1 (single SNPs). Similar to the previous simulation study, 300 continuous phenotypes were generated by increasing the number of relevant variables `causalSNP` within a block of size 8. Figure 4.9 displays the pAUC as a function of `causalSNP`.

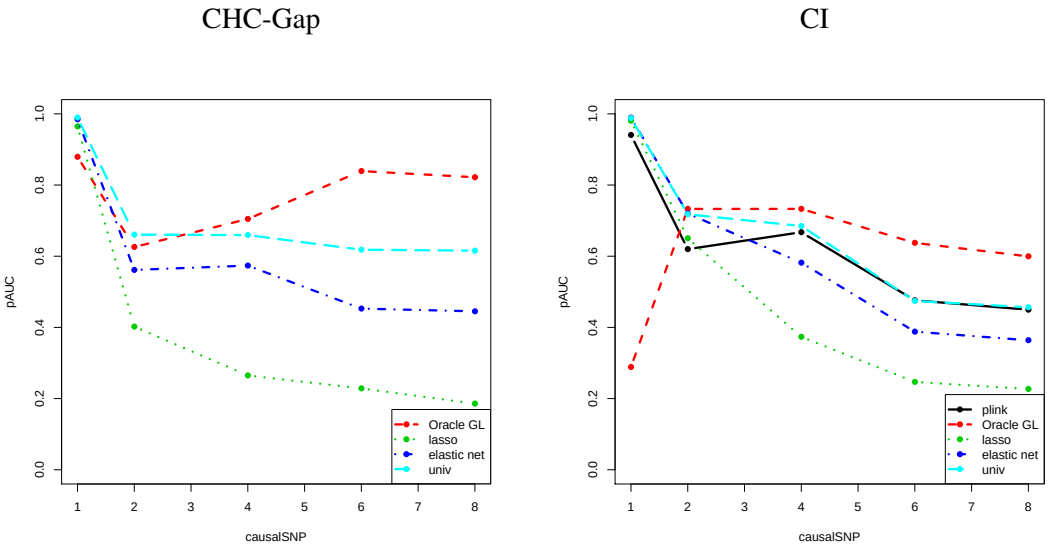


FIGURE 4.9: The mean pAUC as a function of the number of causal SNPs `causalSNP` within a block of size 8, for the haplotype association method ("plink", black solid line), oracle Group Lasso (dashed red lines), Lasso (dotted green lines), Elastic-Net (dash-dotted blue lines) and SMA ("univ", dashed light blue lines). Left: The LD blocks inferred using CHC-Gap. Right: The LD blocks inferred using CI.

Given the blocks estimated with CHC-Gap, we compared the performance of the proposed method to that of the non-grouping approaches (left panel of Figure 4.9). As in Section 4.3.1.1, for `causalSNP` $\in \{1, 2\}$, the proposed approach is outperformed by its competitors because of the high number of false positives generated by the group selection. Conversely, the performance of the competing methods deteriorate significantly as soon as `causalSNP` > 2 which is not the case of the Group Lasso. This result is also consistent with those obtained in Section 4.3.1.3.

Similarly, given the block structure inferred with CI, we investigated the robustness of the oracle Group Lasso, the haplotype association approach and the 3 non-grouping methods to the parameter `causalSNP` (right panel of Figure 4.9). Comparing haplotype association and Group Lasso approaches, we observe a difference of performance when one unique causal SNP is included in a block. The drop in performance of the Group Lasso is due to the difference in the block structure: as explained in Section 4.3.1.1, the Group Lasso penalty increases with block size, making it difficult for this method to select the correct block in presence of many smaller blocks. In practice, this is not problematic as the block selection step in the proposed approach yields larger blocks. On the contrary, the haplotype association method performs a p-value correction that takes the block structure into account, but the p-value of the causal SNP is so small that the adjustment hardly reduces the significance of the block. Furthermore, as in Section 4.2.5, it is remarkable that Group Lasso outperforms competing approaches as soon as `causalSNP` > 2 even for SNP-level evaluation.

The consistency between the results of Sections 4.3.1 and 4.3.2 suggests that the simulation setting used in Section 4.3.1 efficiently mimics a realistic genotyping data set.

4.3.3 Analysis of HIV data

4.3.3.1 Data set

The HIV data set consists of $p = 20,811$ SNPs genotyped for $n = 605$ Caucasian subjects and the plasma HIV-RNA level as phenotype. It corresponds to the phenotype and the genotype data related to the chromosome 6 of the GWA study conducted by Dalmasso et al. (2008). A small number of SNPs were discarded from the study because they generated undefined values of LD. The filtered data set thus contained $20,756$ SNPs. Missing values were imputed using the Bioconductor R package `snpStats` (Clayton 2013). For the proposed approach, this imputation

was performed after the constrained clustering described in Sections 4.2.1.1 and 4.2.1.2, as the proposed constrained clustering algorithm handles missing values. The same data set was used to perform the haplotype association approach. Each of the compared models was adjusted for the gender of the patient.

4.3.3.2 Block inference

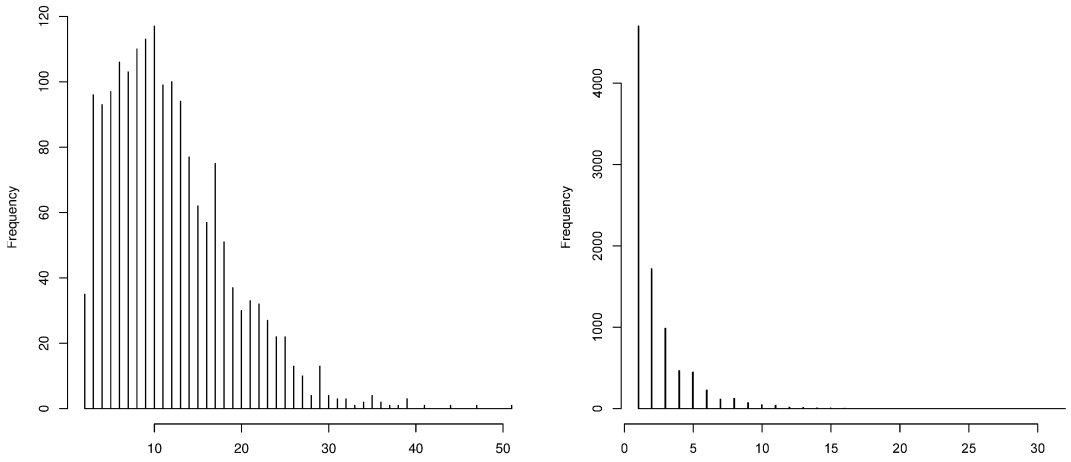


FIGURE 4.10: Histograms of the estimated block sizes of the HIV data. Left: the histogram of the block sizes estimated by the first 2 steps of the proposed method. Right: the histogram of the block sizes estimated by the first step of the haplotype association approach.

The first step of inferring the LD blocks applied to the HIV data estimated 1,756 blocks with $B = 500$ null reference data sets generated in the Gap statistic algorithm. The distribution of the sizes of the obtained blocks is represented in the histogram of Figure 4.10 (left panel). The median block size is close to 10, and the size of the vast majority of blocks is less than 30. The first step of the haplotype association method estimated 9,003 haplotype blocks including 4,699 single SNPs. The size distribution of the obtained blocks is represented in the histogram on the right panel of Figure 4.10. Unlike the LD structure inferred by the proposed approach, the haplotype blocks are much smaller with an average block size of 2.

4.3.3.3 Results on HIV data

We were able to reproduce the results of Dalmasso et al. (2008): the SNPs identified by SMA correspond to the 15 SNPs selected by Dalmasso et al. (2008) at a target False Discovery Rate (FDR) level of 25%. Most of these SNPs are located in the major histocompatibility complex

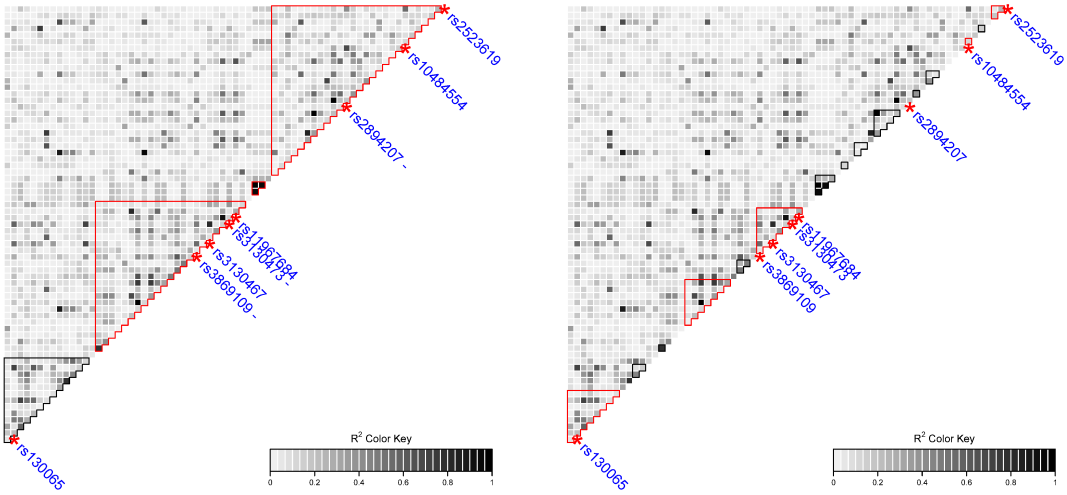


FIGURE 4.11: A linkage disequilibrium (r^2) plot with the inferred block structures (black and red contour lines) for a set of 68 contiguous SNPs located on the MHC region. Left: within the structure inferred by the proposed method, the blocks selected by the Group Lasso are delimited with a red contour line. The SNPs selected by SMA are plotted with a red star (*), and the SNPs missed by Lasso with a blue dash (-). Right: within the structure inferred by the haplotype association method, the blocks selected by the competing method are delimited with a red contour line.

(MHC) region 6p21. A linkage disequilibrium plot of a set of 68 contiguous SNPs within this region is represented in Figure 4.11. The SNPs marked with a red star (*) are those selected by SMA. The first 20 SNPs selected by the Lasso are the same as those selected by the univariate model except for 3 SNPs; the names of these 3 SNPs are marked with blue dashes (-) in the left panel of Figure 4.11.

The local block structures inferred by both the clustering and model selection steps of the proposed method and the competing haplotype association method are also highlighted in this figure (contour lines). The mean LD within the largest two blocks of the left panel is $r^2 = 0.41$ and $r^2 = 0.55$, respectively. The Lasso was able to recover two of the four SNPs identified by Dalmasso et al. (2008) in the first block, and two of the three SNPs identified by Dalmasso et al. (2008) in the second block. This is consistent with the fact that the Lasso is not designed to select correlated variables, as already discussed in Section 4.2.2.

Among the four blocks inferred by the proposed method in this region, the three blocks with a red contour line are among the first 15 blocks selected by the Group Lasso. Almost all of them are of size more than 10 SNPs, except for the two blocks containing 3 and 4 SNPs already identified by Dalmasso et al. (2008), as displayed in Figure 4.11. Each of the 8 remaining SNPs selected by SMA are located in a different LD block of average size around 18 SNPs. The fact

that these SNPs have not been detected by the Group Lasso is consistent with the results of our simulation data. Indeed, Figure 4.4 showed that the Group Lasso tends to select small groups of SNPs because of its default penalty.

Contrary to the Lasso or the Elastic-Net, the proposed approach detected groups of SNPs that had not been identified by Dalmasso et al. (2008). Some of these groups of SNPs may contain interesting candidates, as further discussed below in the description of Figure 4.12.

Similarly to the proposed method, we focused on the first 15 blocks (including single SNPs) selected by the haplotype association approach. The 5 blocks selected by the haplotype association method in the same region represented in Figure 4.11 are represented with a red contour line. The competing approach was able to recover all of the 7 SNPs identified by Dalmasso et al. (2008) and located in this region. However, it detected one group of SNPs that had not been identified in the previous study. This difference could be due to the strong LD ($r^2 = 0.81$) between the SNPs of this block and the SNPs of the block containing 4 markers previously identified as associated with the phenotype.

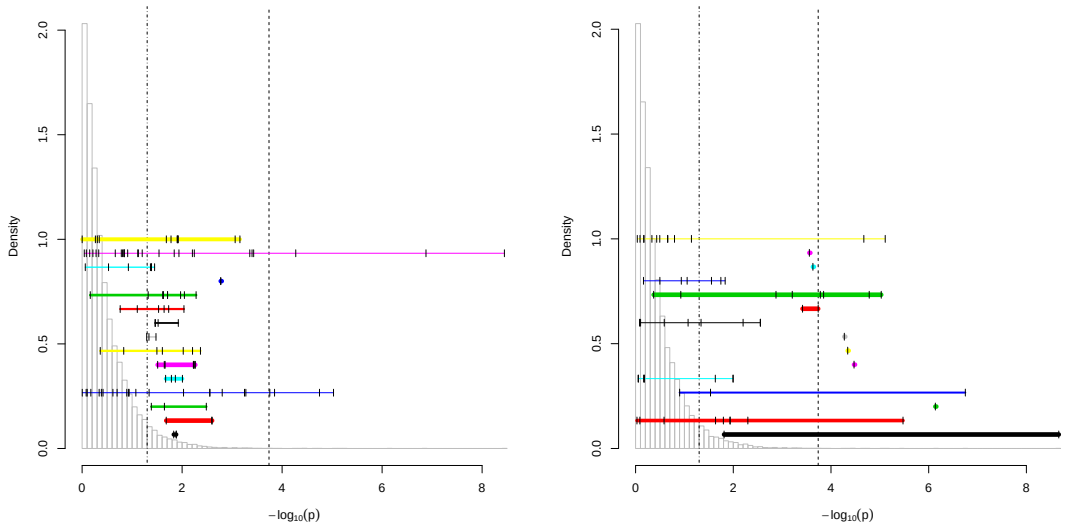


FIGURE 4.12: A comparison between the results of Dalmasso et al. (2008) and the grouping methods on HIV data. The gray histogram represents the distribution of the ($-\log_{10}$-transformed) SMA p-values obtained by Dalmasso et al. (2008). Each of the first 15 blocks selected by the proposed approach (left panel) and the first 15 blocks selected by the haplotype association method (right panel) are represented by a colored horizontal segment ranging from the smallest to the largest SMA p-value of the block. Vertical black segments indicate SMA p-values of each SNP in these LD blocks. Vertical lines highlight the significance threshold used in Dalmasso et al. (2008) (dashed line) and the standard (non multiplicity-corrected) level of 0.05 (dash-dotted line).

Each of the first 15 LD blocks selected by the two grouping strategies are represented as a colored horizontal segment in Figure 4.12, where the x axis corresponds to the ($-\log_{10}$-transformed) SMA p-values obtained by Dalmasso et al. (2008).

For the haplotype association approach (right panel of Figure 4.12), 6 of the 15 blocks consist of a single SNP, that had already been identified in Dalmasso et al. (2008). Moreover, for several of the 15 LD blocks selected by the proposed approach (left panel of Figure 4.12), *all of the SMA p-values of the block* are smaller than the (non multiplicity-corrected) 0.05 level (vertical dash-dotted line at $-\log_{10}(p) = 1.3$). Therefore, although we do not claim that all of these groups of SNPs are relevant to HIV, we believe that some of them might contain interesting candidates. The dashed vertical line highlights the significance threshold used in Dalmasso et al. (2008). Therefore, the 4^{th} and 14^{th} blocks which cross the vertical dotted line correspond to the two largest blocks in the left panel of Figure 4.12, which respectively contain 4 and 3 SNPs previously identified by Dalmasso et al. (2008). We also believe that Figure 4.12 is an interesting diagnostic plot to pinpoint candidate groups of SNPs associated with the disease. Further replication or meta-analysis work would be required to confirm the association between these novel candidates and the phenotype.

4.4 Conclusions

In this chapter, we have proposed a three-step approach that takes into account the biological information of the linkage disequilibrium between variables by firstly inferring LD blocks, then estimating the number of such blocks, and finally performing Group Lasso regression on these inferred groups. This method is implemented as an `R` package. Although we have used a continuous phenotype in our simulations and applications, the approach described in this chapter can be extended to the study of categorical phenotypes, by using the logistic version of each regression model.

We have demonstrated using simulations that the proposed approach efficiently retrieves the underlying block structure for realistic levels of LD between SNPs. Moreover, state-of-the-art SMA and penalized regression approaches Lasso and Elastic-Net are outperformed by our proposed method for the purpose of identifying *blocks containing causal SNPs*. We have argued that selecting *blocks* (rather than individual SNPs) associated with a phenotype is a sensible goal in the GWAS context, where the proportion of heritability explained by individual SNPs is generally low. Interestingly, although the proposed method can only select groups of SNPs and not individual SNPs, our results on simulated data suggest that this approach performs better than state-of-the-art approaches in terms of selection of causal SNPs as soon as the number of associated SNPs within the same LD block exceeds 2.

We have also applied this method to semi-simulated data with a real genotype matrix and a simulated phenotype. As soon as the number of causal markers within a block exceeds 2, the proposed approach shows remarkable performance compared to non-grouping classical strategies, and to an haplotype association method that explicitly takes the block structure information into account. This result suggests that the proposed method is adapted to a real linkage disequilibrium structure.

Finally, an application of this method to HIV data illustrates the ability of the method to (i) partly recover the signal identified by single-locus methods, and (ii) pinpoint previously overlooked associations. These results demonstrate the relevance of the approach, and thereby illustrate the importance of tailored integration of biological knowledge in high-dimensional genomic studies such as GWAS.

A limitation of our proposed method is that it does not provide a significance assessment for the selected groups. Deriving reliable p-values for regression coefficients in high-dimensional,

correlated settings is a challenging research area in the machine learning and statistics fields in general (Bühlmann 2013, Chatterjee & Lahiri 2011). This issue is discussed in the conclusion section of the manuscript.

Chapter 5

An efficient implementation of adjacency-constrained hierarchical clustering of a band similarity matrix

5.1 Algorithmic complexity

An *algorithm* is a procedure that solves a general, well-specified problem. An algorithmic problem is specified by describing the set of inputs it must work on and of its outputs after running the algorithm. An algorithm is then a set of rules that describes a finite sequence of steps to transform these input values into the required output values.

Let us take a toy example of a simple algorithm in order to illustrate the link between problem and algorithm. Algorithm 3 describes the steps for incrementing the elements of an array T of n integers by a.

Algorithm 3

```
1: procedure IncremTab(An array T of size n, an integer a)
2:     for i = 1 to n do
3:         T[i] ← T[i] + a
4:     end for
5:     return T
6: end procedure
```

It is indeed a finite sequence of operations, browsing the elements of the array T, incrementing them by a and providing an output responsive to the problem, namely the array T with the new

values. Note that an algorithm is not a *computer program*. The implementation of an algorithm in a programming language produces a program.

An algorithm defined for a given problem is *correct* if it solves it. The *efficiency* of an algorithm typically considers two criteria:

1. Its **running time**. The longer the algorithm takes to complete, the less efficient it is considered.
2. The **memory space** it uses. The more memory the algorithm uses, the less efficient it is considered.

The optimization of these two criteria is generally antagonistic. Indeed, it is common to increase the memory space used by an algoritm in order to reduce its execution time, by storing a set of previously calculated results for example. We will see later that this chapter illustrates well this tradeoff.

In the previous example of algorithm for incrementing the values of an array, the time for executing Algorithm 3 for an array of n elements of course depends on the speed of increment of each element of the array. But it depends primarily on the size n of the array. Thus, incrementing the elements of an array of size $2n$ will take twice longer than incrementing the elements of an array of n elements, and this regardless of the speed of the operation of increment. Therefore, we seek to measure the efficiency of an algorithm, in terms of execution time, independently of the processor speed and the programming language used to implement the algorithm. To do so, we use the notion of *complexity*. The complexity is the number of *elementary operations* (indivisible operations) necessary for the execution of the algorithm. This number is generally expressed as a function of the size of the input. In the example of Algorithm 3, the operation of increment must be performed n times, whatever the increment speed.

The exact number of elementary operations required to perform an algorithm is generally a numerical function of the size of the input. This number of operations can vary according to certain specific input values. It may also depend on implementation details, creating a constant number of operations in addition to perform. For this reason, we are usually studying the *asymptotic* efficiency of algorithms. That is, we are concerned with how the running time of an algorithm increases as the size of the input increases. For instance, indicating that the complexity of an algorithm is of $T(n) = 1250n^2 + 15n + 830\log_{10}(n) + 50$ provides us little extra information

than the observation that "the time grows quadratically with n". It is then much easier to talk in terms of simple upper bound of complexity functions using the *Big-O* ($\mathcal{O}$) notation. The $\mathcal{O}$ simplifies the complexity analysis by ignoring levels of detail that do not impact this complexity as the size of the input grows. Mathematically speaking, for two functions f and g, $f = \mathcal{O}(g)$ if there is a positive constant c such as $f(n) \leq cg(n)$ for n sufficiently large. In other words, the notation $f = \mathcal{O}(g)$ expresses that f is smaller or equal to g if the constant factors (additive and multiplicative) are ignored. For instance, if $f(n) = 100n^2 + 10n$ and $g(n) = n^2$, then $f = \mathcal{O}(g)$, that is $f(n) = \mathcal{O}(n^2)$. More details and examples of algorithms and complexities are provided in Skiena (1998) and in Sedgewick (1988).

In addition to its asymptotic complexity, the efficiency of an algorithm is also evaluated by knowing how it works over all possible inputs. For this reason, we are generally interested in the *worst-case complexity* of the algorithm, that is the function of n defined by the maximum number of elementary operations required for any input of size n. All the complexities discussed in this chapter are worst-case complexities.

The notion of *space complexity* also exists. It is the maximum amount of space used at any step of the algorithm, ignoring the space used by the input. The notation for space complexity is the same as the notation for time complexity: it is expressed as a function of the input size. Thereafter, when the term complexity is used without any specification, it will refer to time complexity, that is the number of elementary operations.

Some types of worst-case complexities are named as follows, where n is the size of the input:

- **Sublinear algorithms:**
  - Algorithms in **constant time** have a complexity of $T(n) = 1$. The number of operations remains constant, regardless of the size of the input.
  - **Logarithmic algorithms** have a complexity of $T(n) = \log(n)$. The complexity increases slightly with n. This type of complexity is found in a loop where the input size is divided at each iteration. As a matter of fact, the base of the logarithm is usually ignored when analyzing algorithms because multiplicative terms are ignored in the $\mathcal{O}$ notation.
- **Linearithmic algorithms** have a complexity of $T(n) = n\log(n)$. Such complexity grows faster than a linear one but slower than any polynomial in n with exponent strictly greater than 1.

- **Polynomial algorithms** have a complexity of $T(n) = n^c$, where c is a constant. Some polynomial algorithms are named specifically:
  - **Linear algorithms** have a complexity of $T(n) = n$. This is typically the complexity of a loop performing n iterations, with a constant number operations at each iteration, as for Algorithm 3.
  - **Quadratic algorithms** have a complexity of $T(n) = n^2$. It is the complexity of two nested loops: a loop performing n times a loop performing n iterations.
- **Exponential algorithms** have a complexity of $T(n) = c^n$, where c is a constant. For example, listing all the subsets of the set $\{1, 2, \ldots, n\}$ is optimally performed in $\mathcal{O}(2^n)$.

5.2 The adjacency-constrained hierarchical clustering algorithm

Now that we have introduced some concepts of algorithmics, let us study the complexity of the basic algorithm of a hierarchical clustering with adjacency constraint using the similarity Sim, applied to a set of p items $\mathbf{X}_{.1}, \mathbf{X}_{.2}, \ldots, \mathbf{X}_{.p}$.

5.2.1 Time and space complexities

Algorithm 4 presents the basic steps of an adjacency-constrained hierarchical clustering. It is in fact a high-level description of the clustering algorithm, in the sense that it does not show all the details of the clustering procedure such as the linkage criterion used. We will see later that an implementation-level description of this algorithm is required to precisely determine its time and space complexities.

In order to determine the complexity of the constrained hierarchical clustering algorithm, the number of elementary operations required at each step of the clustering will be counted. Table 5.1 details the complexities of Algorithm 4 line by line. As regards lines 7 and 8, their complexity depends on the implementation of the algorithm: if all distances between all possible pairs of clusters have already been calculated and stored then lines 7 and 8 would have complexity of $\mathcal{O}(1)$. And the number of elementary operations necessary to pre-calculate these distances should be added to the complexity of the whole algorithm. Otherwise, lines 7 and 8 have a worst-case complexity of $\mathcal{O}(p)$.

Algorithm 4 The constrained hierarchical clustering algorithm

1: **procedure** CAHC($\mathbf{X} \in \mathbb{R}^{n \times p}$, Sim)
2: $\mathcal{C} \leftarrow \{C_i = \{\mathbf{X}_{.i}\}, i \in 1, \ldots, p\}$ ▷ each item in separate cluster
3: $D \leftarrow \{1 - \text{Sim}(\mathbf{X}_{.i}, \mathbf{X}_{.(i+1)}), i \in 1, \ldots, p-1\}$ ▷ the $(p-1)$ vector of dissimilarities between adjacent items
4: **for** $step = 1$ **to** $p - 1$ **do**
5: $\{C_{i^\star}, C_{i^\star+1}\} \leftarrow \arg\min_{i \in \{1,\ldots,p-step\}} D(C_i, C_{i+1})$ ▷ find the closest pair of adjacent clusters
6: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_{i^\star}, C_{i^\star+1}\} \cup \{C_{i^\star} \cup C_{i^\star+1}\}$ ▷ updating the clustering
7: $d_1 \leftarrow D(C_{i^\star-1}, C_{i^\star} \cup C_{i^\star+1})$
8: $d_2 \leftarrow D(C_{i^\star} \cup C_{i^\star+1}, C_{i^\star+2})$
9: $D \leftarrow D \setminus \{D(C_{i^\star-1}, C_{i^\star}), D(C_{i^\star}, C_{i^\star+1})\} \cup \{d_1, d_2\}$ ▷ updating D
10: **end for**
11: **return** T
12: **end procedure**

Line number in Algorithm 4	Number of elementary operations
line 2	p
line 3	$p-1$
line 5	$p - step$
line 6	$p - step$
line 7 (same for line 8)	depends on the implementation
line 9	$p - step$

TABLE 5.1: Number of elementary operations required at each step of Algorithm 4.

Hence, given that the operations of lines 5 to 9 are executed within a loop of $p-1$ steps then the number of elementary operations necessary for the performance of Algorithm 4 is:

$$T_4(p) = \underbrace{p}_{\text{line 2}} + \underbrace{p-1}_{\text{line 3}} + \underbrace{\sum_{step=1}^{p-1}}_{\text{the loop}} \left(\underbrace{(p - step)}_{\text{line 5}} + \underbrace{(p - step)}_{\text{line 6}} + \underbrace{2(p-1)}_{\substack{\text{the worst-case complexity} \\ \text{of lines 7 and 8}}} + \underbrace{(p - step)}_{\text{line 9}} \right).$$

As seen in Section 5.1, the expression of T_4 can be summarized by: the computation time of the constrained hierarchical clustering algorithm increases quadratically with the number p of items. In other words, the complexity of Algorithm 4 is in $\mathcal{O}(p^2)$.

The space complexity of the high-level `cAHC` algorithm depends on its implementation. For instance, if the p^2 distances between all possible pairs of clusters are calculated and stored beforehand then the space complexity of Algorithm 4 is in $\mathcal{O}(p^2)$. Such an implementation is used in the function `chclust` of the R package `rioja`. Conversely, if the p^2 distances are

calculated on the fly, only the size of internally created objects such as the vectors of distances and clusters will count, resulting in a space complexity of $\mathcal{O}(p)$.

Note that Algorithm 1 is a particular case of Algorithm 4. More particularly, the `cWard_LD` algorithm uses $\mathrm{ld}(\mathbf{X}_{.i}, \mathbf{X}_{.j})$ as a similarity between the items $\mathbf{X}_{.i}$ and $\mathbf{X}_{.j}$ (that are SNPs) and the Ward's criterion as a linkage function. Therefore, the `cWard_LD` algorithm has the same algorithmic properties as the `cAHC` algorithm, that is, it has a quadratic behavior in computation time with the number p of SNPs (see Figure 5.1).

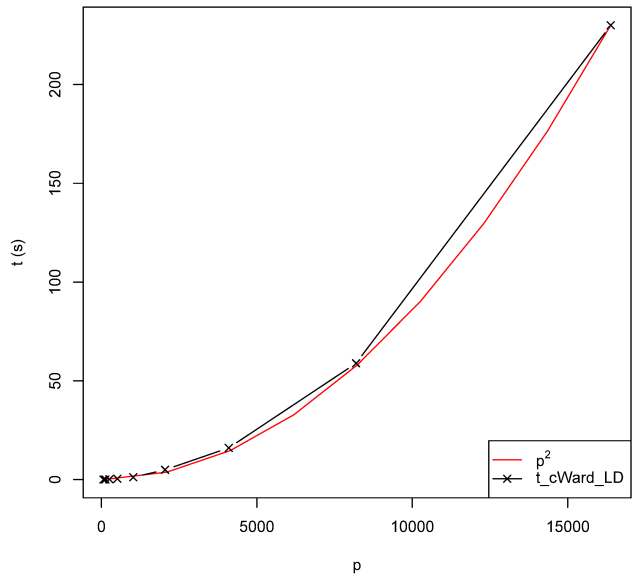


FIGURE 5.1: Computation time (in seconds) of the `cWard_LD` algorithm (in black) and theoretical time complexity (p^2, in red) as functions of the number of markers p.

Concerning the implementation of Algorithm 1, a naive implementation would consist in (i) calculating the $p(p-1)/2$ LD measures for each pair of SNP and (ii) performing constrained hierarchical clustering on the obtained similarity matrix. As p is typically of the order of 10^4 to 10^5 for a single chromosome in a standard GWAS, such an implementation with space complexity of $O(p^2)$ is not appropriate. Indeed, for a single chromosome of length $p = 10^5$, this algorithm would require storing of the order of 10^{10} numeric values of LD before applying the clustering algorithm. To overcome this difficulty, our implementation of the constrained clustering takes as input the $n \times p$ matrix of genotypes $\mathbf{X}$, and calculates the LD measures incrementally as they are required by the clustering. The LD measures are calculated directly from genotypes using the Bioconductor `R` package `snpStats` (Clayton 2013, Clayton & Leung 2007), which handles missing values.

5.2.2 Scalability to high-dimensional data

The scalability of the adjacency-constrained hierarchical clustering algorithm to high-dimensional data were evaluated on SNP data and using LD similarity.

By first applying the function `chclust` (R package `rioja`) to simulated genotype matrices with an increasing number of SNPs, we noticed that this implementation could not handle more than 8000 SNPs to be clustered because of memory issues.

We then run the `cWard_LD` algorithm using the implementation described in Section 5.2.1. Table 5.2 presents the computation times of this procedure applied to p SNPs genotyped in 100 individuals for different values of p. More particularly, we can notice that applying the `cWard_LD` algorithm to 32768 SNPs takes more than 22 minutes. Furthermore, it takes 4.5 hours (on a standard 2.2 Ghz single CPU) to analyze a whole genome of $500k$ simulated SNPs (for Affymetrix $500k$ arrays) genotyped on 100 individuals, when the algorithm is applied chromosome by chromosome since it uses the LD measure as a similarity.

Number of SNPs	256	1024	4096	16384	32768
Running time (min)	0.004	0.0278	0.3617	5.5168	22.067

TABLE 5.2: Running time of the `cWard_LD` algorithm applied to p SNPs genotyped in 100 individuals.

These computation times of the `cWard_LD` algorithm, and more generally the quadratic complexity of the adjacency-constrained hierarchical clustering remain unsatisfactory for many reasons such as:

- such complexity makes the BALD approach, presented in the preceding chapter, computationally intensive. Indeed, $B = 500$ adjacency-constrained hierarchical clustering procedures needed to be run within the Gap statistic approach for estimating the optimal number of clusters.
- the number of SNPs clustered in 4.5 hours is very small compared to the 160 million of SNPs that have been identified in humans and that could be genotyped using high-throughput sequencing technologies.
- other potential applications, such as HiC data analysis, also require the use of algorithms that are adapted to high-dimensional data.

Therefore, the remainder of this chapter will be dedicated to the description of an optimized implementation of the adjacency-constrained hierarchical clustering algorithm.

5.3 A generalized and efficient implementation of the adjacency-constrained hierarchical algorithm

The proposed implementation of the `cWard_LD` algorithm takes advantage of the fact that SNPs which are far apart along a chromosome are not in linkage disequilibrium. Using this feature, it is possible to decrease the time complexity of the `cWard_LD` algorithm from $\mathcal{O}(p^2)$ to $\mathcal{O}(p\log(p) + ph)$ and maintain its space complexity linear in p, where $h \ll p$ is a user-defined parameter such that the LD between two SNPs distant from more than h can safely be set to 0.

As a matter of fact, the following improvements can be applied to any adjacency-constrained hierarchical clustering algorithm that uses Ward's criterion, regardless of the similarity (or dissimilarity) chosen between the items. Consequently, we will be considering thereafter a similarity matrix in general so the input variable Sim used in the generalized `cWard` algoritm (Algorithm 5) is not necessarily a LD similarity and the items $\{\mathbf{X}_{.i}, i = 1, \ldots, p\}$ we aim to cluster are not necessarily SNPs. Note also that, unlike the `cWard_LD` algorithm, the generalized `cWard` algorithm takes as input the parameter h. This algorithm is, again, a high-level description of the clustering procedure which explains the fact that h does not explicitly appear in the body of the algorithm.

Algorithm 5 Ward's constrained hierarchical clustering applied to a h-band similarity matrix

procedure CWARD($\mathbf{X} \in \mathbb{R}^{n\times p}$, Sim, h)
  $\mathcal{C} \leftarrow \{C_i = \{\mathbf{X}_{.i}\}, i \in 1, \ldots, p\}$ ▷ each SNP in separate cluster
  $D \leftarrow \{1 - \text{Sim}(\mathbf{X}_{.i}, \mathbf{X}_{.(i+1)}), i \in 1, \ldots, p-1\}$ ▷ the $(p-1)$ vector of dissimilarities between adjacent SNPs
  for $step = 1$ **to** $p - 1$ **do**
    $\{C_{i^\star}, C_{i^\star+1}\} \leftarrow \arg\min_{i\in\{1,\ldots,p-step\}} D(C_i, C_{i+1})$ ▷ find the closest pair of adjacent clusters
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_{i^\star}, C_{i^\star+1}\} \cup \{C_{i^\star} \cup C_{i^\star+1}\}$ ▷ updating the clustering
    $d_1 \leftarrow D(C_{i^\star-1}, C_{i^\star} \cup C_{i^\star+1})$ ▷ using Equation 3.6
    $d_2 \leftarrow D(C_{i^\star} \cup C_{i^\star+1}, C_{i^\star+2})$
    $D \leftarrow D \setminus \{D(C_{i^\star-1}, C_{i^\star}), D(C_{i^\star}, C_{i^\star+1})\} \cup \{d_1, d_2\}$ ▷ updating D
  end for
  return T
end procedure

Unlike Algorithm 4, the introduction of the user-parameter h in Algorithm 5 incorporates a biological constraint that allows to reduce the number of similarities to be calculated from p^2 in the `cAHC` algorithm to ph in the `cWard` procedure. Nevertheless, similarly to Algorithm 4, this high-level description of Algorithm 5 still has a complexity of $\mathcal{O}(p^2)$ since many steps within its main loop (lines 4 to 10), and more specifically those at lines 5, 7 and 8, are linear in p. We then need to make these steps sublinear in time (constant or logarithmic) in order to avoid the quadratic growth of the clustering complexity with p.

To this end an implementation trick will be described, in Section 5.3.2, in order to reduce the complexity of the operations in lines 7 and 8 of Algorithm 5. Then, Section 5.3.3 presents a new data structure that is used to optimize the complexity of the line 5 step of Algorithm 5. These improvements will result in an implementation-level description of the optimized `cWard` algorithm in Section 5.3.4.

5.3.1 The h-band similarity matrix

The introduction of a parameter h aims to control the maximum lag between items $\mathbf{X}_{\cdot i}$ and $\mathbf{X}_{\cdot j}$ for similarity calculations. Thus, the similarity measures are computed between these two items only if i and j differ by no more than h. This lag of size h leads to a similarity matrix in band with non zero coefficients within a band of width $2h$. An example of such a h-band similarity matrix is illustrated in Figure 5.2.

Recall that, by using the Ward's criterion coupled with the LD kernel trick, the distance between two clusters A and B equals to:

$$d_{\text{wl}}(A,B) = \frac{p_A p_B}{p_A + p_B}\left(\frac{1}{p_A^2}S_{A,A} + \frac{1}{p_B^2}S_{B,B} - \frac{2}{p_A p_B}S_{A,B}\right), \tag{5.1}$$

with

$$S_{A,B} = \sum_{i\in A, j\in B} \text{ld}(\mathbf{X}_{\cdot i}, \mathbf{X}_{\cdot j}).$$

A condition for using this formula with a similarity kernel other than the LD only requires that the similarity between an item and itself equals to 1, which corresponds to diagonal elements of the similarity matrix in Figure 5.2.

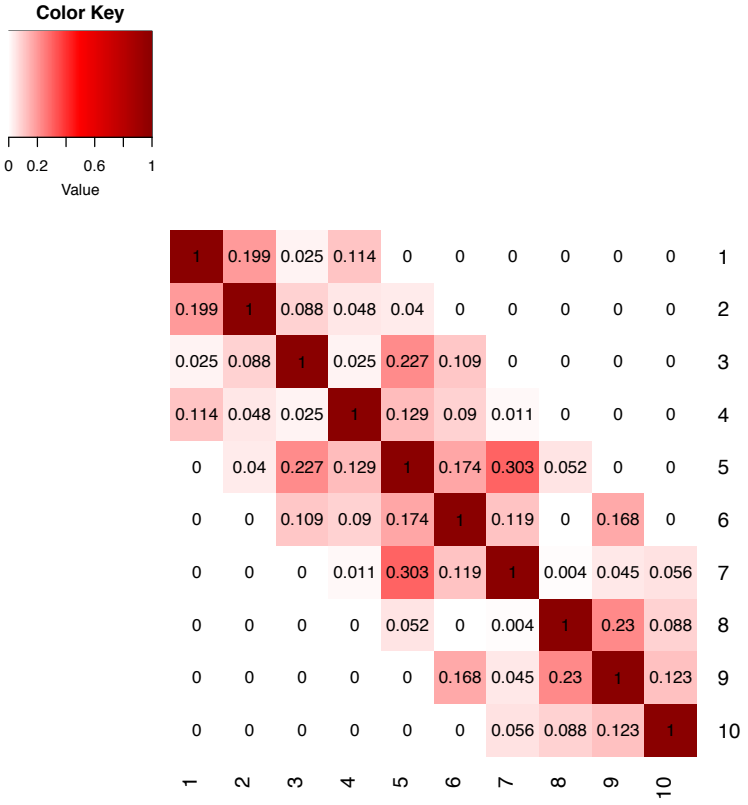


FIGURE 5.2: Heatmap of a similarity matrix with $p = 10$ and $h = 3$.

5.3.2 The pencils' trick

The implementation trick described in this section ensures that the steps in lines 7 and 8 of Algorithm 5 are each performed in constant time, that is the complexity of the calculation of the new distances step is of $\mathcal{O}(1)$.

According to Equation 5.1, S_{AA} corresponds to the sum of the similarity measures between the items of cluster A. Similarly S_{BB} equals to the sum of the similarity measures between the items of cluster B and S_{AB} is the sum of the similarity measures between items of cluster A and those of cluster B (see Figure 5.3).

Therefore, as the distance $d_{\text{wl}}(A, B)$ between any pair of clusters A and B is fully known as soon as the quantities S_{AA}, S_{BB} and S_{AB} are calculated, the latter will be computed at the beginning of the clustering algorithm, for all possible pairs of clusters A and B using *pencil-shaped* areas that appear in the similarity matrix. More particularly, as displayed in Figure 5.4, S_{AA} can be calculated by adding the sum of the similarity measures contained in the red-outlined pencil-shaped area (a right oriented *pencil*) to the sum of the similarity measures contained in the green-outlined pencil-shaped area (a left oriented pencil) and deducting from the total the sum

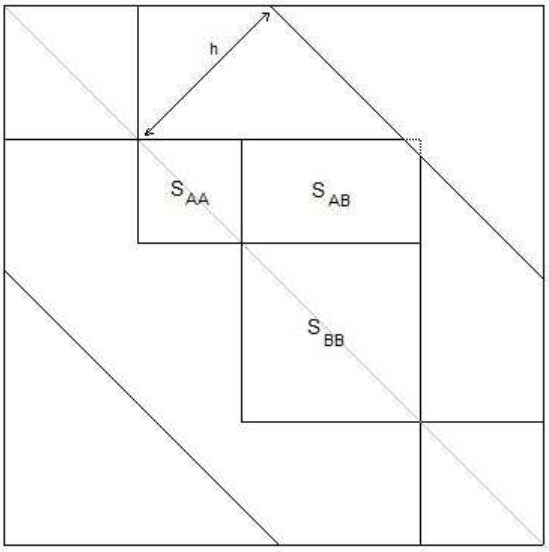


FIGURE 5.3: A schematics of the h-band similarity matrix for illustration of the quantities S_{AA}, S_{BB} and S_{AB} used in the calculation of $d_{\text{wl}}(A, B)$.

of the similarity measures contained in the full diagonal band of width $2(\max(A) - \min(A))$. $\max(A)$ $(\min(A))$ denotes the maximum (minimum) of the positions (from 1 to p) of the items contained in A.

The same calculation method can be applied to compute S_{BB}. Once S_{AA} and S_{BB} calculated, as shown in Figure 5.5, S_{AB} corresponds to half of the sum of the similarity measures contained in the blue-outlined pencil-shaped area (left oriented pencil) plus the sum of the similarity measures contained in the yellow-outlined pencil-shaped area (right oriented pencil) minus the sum of the similarity measures on the full diagonal band of width $2h$ minus S_{AA} minus S_{BB}.

Finally, it appears that, at each step of the clustering, computing Ward's criterion between any two clusters requires only the calculation of sums of similarity measures within *pencil-shaped areas* that can be fully defined by three parameters:

- their width: `hLoc`.
- their end-point: `lim`.
- their orientation: `sense="right''` or `"left''`.

For example, the red-outlined pencil-shaped area presented in Figure 5.4 is well defined by setting: `hLoc` $= 2(\max(A) - \min(A))$, `lim` $= \max(A)$ and `sense` `="right''`. Consequently, the beginning of the clustering algorithm will consist in calculating the sums of all the

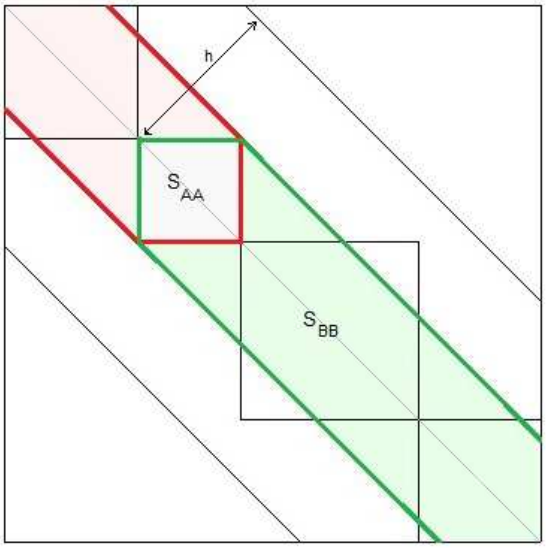


FIGURE 5.4: A schematics of the pencil-shaped areas used for calculating S_{AA}. S_{AA} equals to the sum of the similarity measures contained in the pencil-shaped red-outlined area to the sum of the similarity measures contained in the pencil-shaped green-outlined area and deducting from the total the sum of the similarity measures contained in the full diagonal band of width $2(\max(A) - \min(A))$.

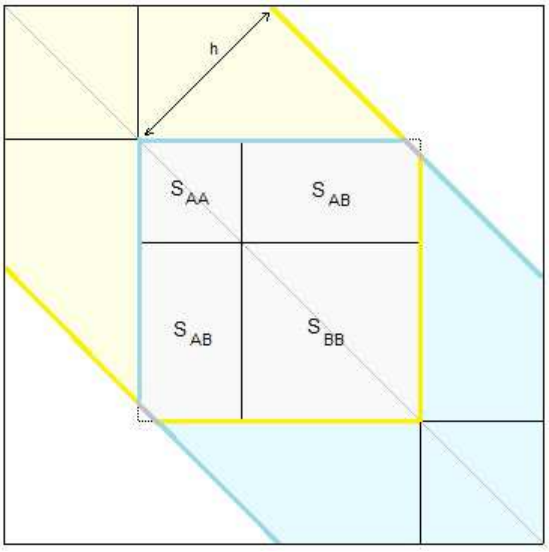


FIGURE 5.5: A schematics of the pencil-shaped areas used for calculating S_{AB}. S_{AB} equals to half of the sum of the similarity measures contained in the pencil-shaped blue-outlined area plus the sum of the similarity measures contained in the pencil-shaped yellow-outlined area minus the sum of the similarity measures on the full diagonal band of width $2h$ minus S_{AA} minus S_{BB}.

pencil-shaped areas of depth $\texttt{hLoc} \in \{1, \ldots, h\}$, of end points $\texttt{lim} \in \{1, \ldots, p\}$ and of the two possible orientations. These values are stored in two arrays of sizes $p \times h$ corresponding to the right-oriented and left-oriented "pencils". As a result, at each step of the clustering, assessing the two distances with the newly-merged cluster (lines 7 and 8 of Algorithm 5) will consist in a simple access to the adequate values in the two arrays of pencils sums.

More details on calculating sums of similarities within pencil-shaped areas are described in Appendix C.

Note that, the improvement suggested so far is not yet sufficient since the line 5 operation of Algorithm 5 is still linear in time, resulting in a quadratic complexity of the whole algorithm.

5.3.3 Reducing the time complexity of finding the best fusion

The purpose of this section is to introduce the concept of *binary heaps* and describe the usage of this data structure within the `cWard` algorithm in order to optimize the complexity of the line 5 step of Algorithm 5, that is finding the best fusion in the sense of the Ward's criterion.

5.3.3.1 Binary heaps

Heaps are, after the search trees, the second most studied type of data structure (Brass 2008). As abstract structures they are also called *priority queues*. The heap structure was originally introduced by Williams (1964) for the very special application of sorting, although he did already present it as a separate data structure with possibly further applications.

A *binary heap* is an array object that can be viewed as a *partially ordered* complete binary tree.

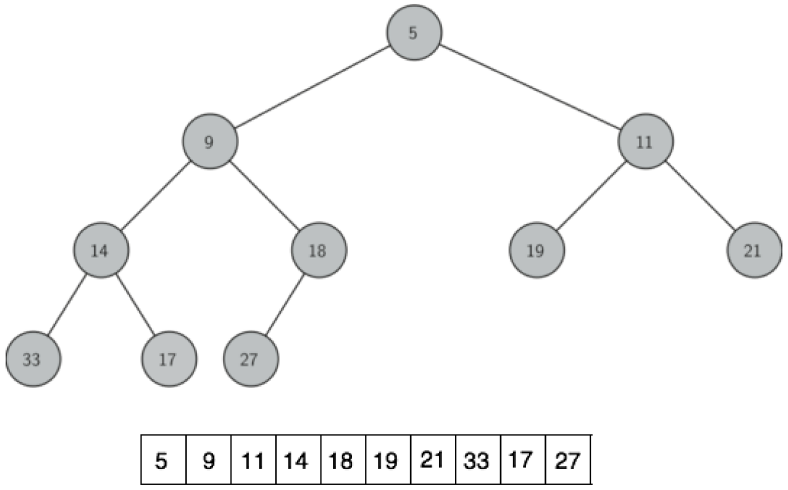


FIGURE 5.6: A min-heap viewed as a binary tree and an array. The min-heap has height 3.

It is a binary tree means that, it is a tree data structure in which each node has at most two children, which are referred to as *left child* and *right child*. Then, when the heap is stored as an

array A, the root of A is $A[1]$, and given the index i of any node in A, the indices of its parent and children can be determined as follows:

$$\begin{aligned} \text{Parent}(i) &= \lfloor i/2 \rfloor \\ \text{Left}(i) &= 2i \\ \text{Right}(i) &= 2i + 1. \end{aligned}$$

The height of a binary heap is defined as the number of edges of the longest simple path from the root to a leaf. Since a heap of p elements is based on a complete binary tree, its height is $\log_2(p)$.

The binary heap is partially ordered means that there is a partial order relationship between the value of a node and the values of its children. In a *min-heap*, as shown in Figure 5.6, the value of a node is less than or equal to the values of its children. In a *max-heap*, the value of a node is greater than or equal to the values of its children. Consequently, the smallest (largest) value in a min-heap (max-heap) is at the heap's root. In the rest on the manuscript, we will focus on min-heaps.

Some basic procedures are applied to min-heap data structures such as:

- `DeleteMin` which consists in deleting the minimum element (which is the root) from the min-heap and restoring its properties.
- `InsertHeap` which is the procedure for adding an element to a min-heap by maintaining the min-heap properties.
- `BuildHeap` which produces a min-heap from an unordered input array.

The `DeleteMin` procedure: There is a conventional approach to delete the minimum element from a min-heap. As the root of the min-heap contains the minimum element, deletion always happens from the root which is replaced by the last element of the min-heap. However, at that root position the new element may violate the min-heap property if it is greater than one of its children. In that case, an operation of *percolation down* must be applied to the newly-placed element in order to maintain the min-heap property. The steps of this procedure are detailed in Algorithm 6.

Algorithm 6 Percolation down algorithm

```
1:  procedure PERCDOWN(H, p, i)
2:      l ← Left(i)
3:      r ← Right(i)
4:      if l ≤ p and H[l] < H[i] then
5:          min ← l
6:      else min ← i
7:      end if
8:      if r ≤ p and H[r] < H[min] then
9:          min ← r
10:     end if
11:     if min ≠ i then
12:         aux ← H[i]
13:         H[i] ← H[min]
14:         H[min] ← aux
15:         PercDown(H, p, min)
16:     end if
17: end procedure
```

Given a min-heap H of p elements and a position $i \leq p$, at each step of the percolation down procedure, the minimum of the elements $H[i]$, $H[\text{Left}(i)]$ and $H[\text{Right}(i)]$ is determined. And its index is stored in min (lines 2 to 10). If $H[i]$ is the minimum, then the subtree rooted at node i is already a min-heap and the procedure terminates. Otherwise, one of the two children has the minimum element, and $H[i]$ is swapped with $H[min]$ (lines 12 to 14), which causes node i and its children to satisfy the min-heap property. The node indexed by min, however, now has the original value $H[i]$, and thus the subtree rooted at min might violate the min-heap property. Consequently, we call `percDown` recursively on that subtree (line 15).

The `DeleteMin` procedure applied to a heap of height 3 is illustrated in Figure 5.7. The root of the min-heap, which contains the minimum element 2 (panel a), is first replaced by the last element 14 of the min-heap (panels b and c). Given that the element 14 is greater than its two children 3 and 4, it is then swapped with the smallest of its children 3 (panel d). Similarly, in its new place, 14 is also greater than its two children 8 and 10, it is then swapped with 8 (panel e). It is then in the right place since it has no more children. In this example, the operation of percolation down was carried out in 2 steps of swaps within a heap of height 3. More generally, at a percolation down algorithm are performed at most $h = \log_2(p)$ swap operations for a min-heap of p elements, resulting in a time complexity of $\mathcal{O}(\log_2(p)) = \mathcal{O}(\log(p))$ for the `DeleteMin` procedure.

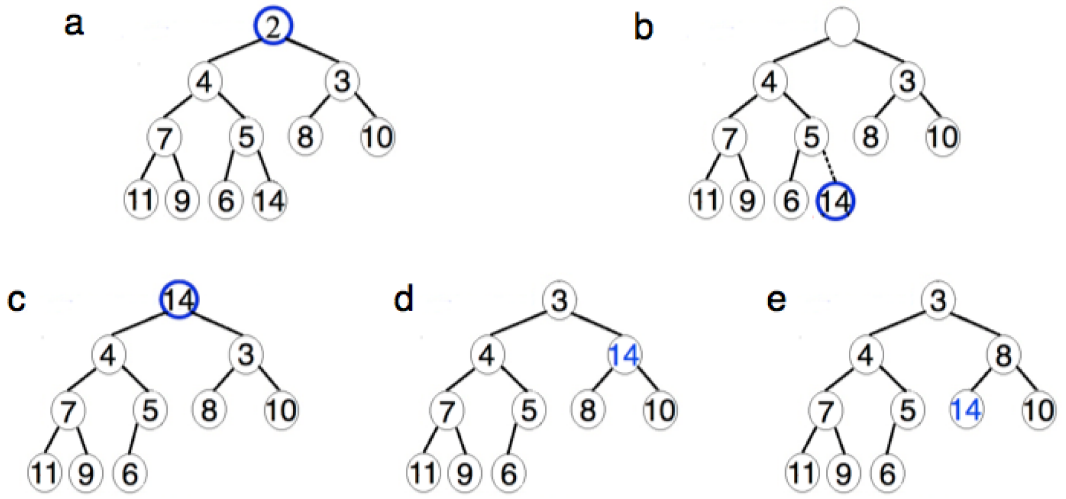


FIGURE 5.7: The procedure `DeleteMin`(H). The root of the tree 2 (a) is deleted and replaced by the last element of the tree (b and c). The min-heap property is restored by successively swapping 2 with the smallest of its children 3 (d) and 8 (e).

The `InsertHeap` procedure: To add a new element to a min-heap H of p elements, first a new node is put at the right of the last leaf of the tree. Given that this insertion may break the order property of the min-heap, it is then necessary to perform a *percolation up* operation, that is the newly-added node is successively swapped with its parent, until the value of the parent node is less than that of the inserted node. The percolation up operation applied to the new min-heap of size $p' = p + 1$, at its last position p', is described in Algorithm 7.

Algorithm 7 Percolation up algorithm

```
1:  procedure PercUp(H, p')
2:      pos ← p'
3:      parent ← Parent(pos)
4:      while pos > 1 do
5:          if H[parent] > H[pos] then
6:              aux ← H[parent]
7:              H[parent] ← H[pos]
8:              H[pos] ← aux
9:          else pos ← 1
10:         end if
11:         pos ← parent
12:         parent ← Parent(pos)
13:     end while
14: end procedure
```

In the example of Figure 5.8, the element 2 put at the last position is less than its parent 5, it is then exchanged with it (panel c). Similarly, 2 is less than its successive parents 4 and 3 so it is successively swapped with them (panels d and e) and thus reaching the root of tree. For

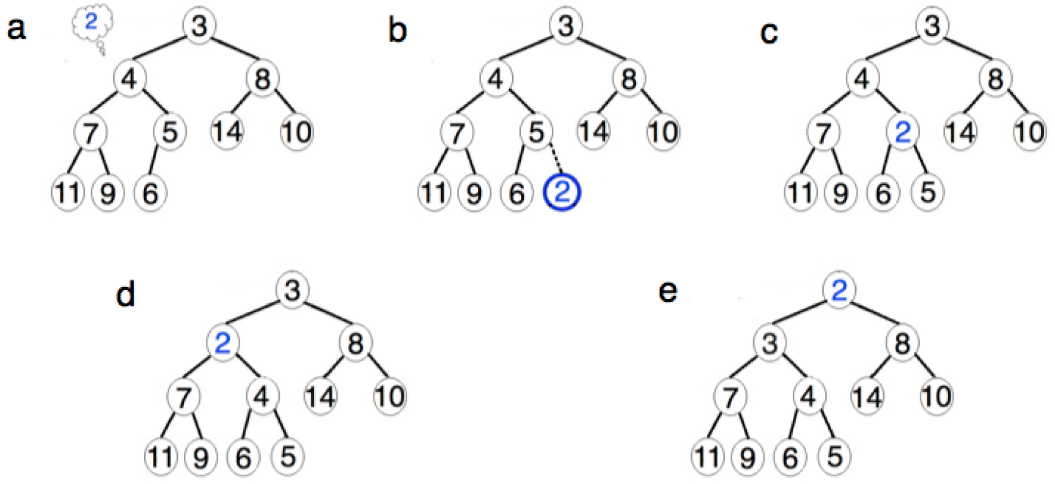


FIGURE 5.8: The procedure of `InsertHeap`$(H, 2)$. The element 2 is inserted at the first free node of the tree (b). The min-heap property is restored by successively exchanging 2 with its parents 5 (c), 4 (d) and 3 (e).

the same reasons that for the `DeleteMin` algorithm, at most $h = \log_2(p)$ swap operations are performed at the percolation up for a min-heap of p elements, resulting in a time complexity of the `InsertHeap` procedure of $\mathcal{O}(\log_2(p)) = \mathcal{O}(\log(p))$.

The `BuildHeap` procedure: In order to convert an array A of p elements into a min-heap, successive percolation down procedures must be performed in a bottom-up manner starting from the middle of the tree as follows:

```
1: procedure BUILDHEAP(A, p)
2:     for i = ⌊p/2⌋ down to 1 do
3:         PercDown(A, p, i)
4:     end for
5: end procedure
```

For instance, in the example of Figure 5.9, the 10-element array is first represented as a tree. Then, starting from the position $\lfloor 10/2 \rfloor = 5$ of the array, corresponding to the node of the element 2 (blue rectangle 1), down to the top of the tree (that is following the blue rectangles 2 to 5 of Figure 5.9), percolation down operations are successively performed. The proof of obtaining a min-heap using this algorithm is detailed in Cormen (2009).

Each call of the percolation down procedure costs $\mathcal{O}(\log(p))$ in time and the `BuildHeap` algorithm makes $\mathcal{O}(p)$ such calls. Thus, the running time of the `BuildHeap` procedure is in $\mathcal{O}(p \log(p))$.

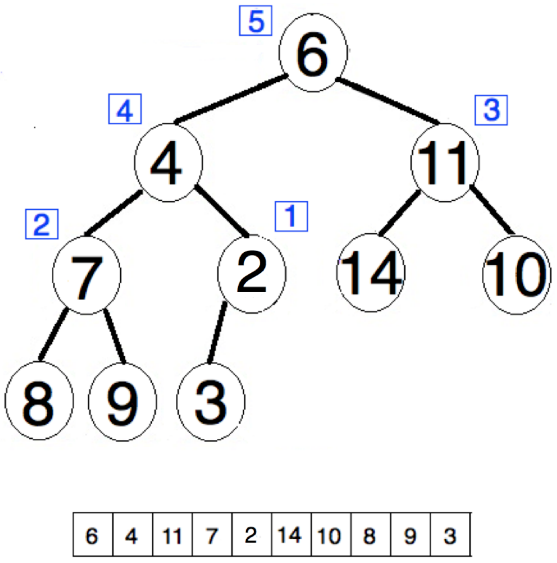


FIGURE 5.9: The procedure of `BuildHeap`, showing the data structure before the successive percolation down operations. The blue rectangles 1 to 5 refer to the steps of the building the min-heap.

To sum up, Table 5.3 summarizes the complexities of the preceding elementary operations applied to min-heaps and unordered arrays of size p. These complexities justify the interest in the min-heap structure. Indeed, when applied to min-heaps, these elementary operations are at worst linearithmic when integrated within a loop of p operations.

	findMin	**insert**	**deleteMin**
unordered array	$\mathcal{O}(p)$	$\mathcal{O}(1)$	$\mathcal{O}(p)$
min-heap	$\mathcal{O}(1)$	$\mathcal{O}(\log(p))$	$\mathcal{O}(\log(p))$

TABLE 5.3: A comparison of time complexities of finding the minimum element, inserting an element and deleting the minimum element operations applied to an unordered array and a min-heap of size p.

5.3.3.2 Finding the best fusion using a min-heap

The first intuitive idea for using the min-heap structure in order to reduce the complexity of the step at line 5 of Algorithm 5 consists in storing the vector of distances D in a min-heap and update it (by deleting the minimal element and adding distances with the newly-formed cluster) at each step of the clustering. Besides, at each step of the `cWard` algorithm, the relative position (from 1 to p) of the pair of clusters corresponding to the minimal Ward's distance has to be known for the calculation of the two distances between the newly-formed cluster and its adjacent clusters. Nevertheless, the simple implementation of D as a min-heap breaks the link

between the distance between a pair of clusters and the relative position of these clusters among other clusters. So, in order to maintain this connection, a structure of a "chained array" in addition to the min-heap has been used. These structures and the relationship between them are illustrated in Figure 5.10.

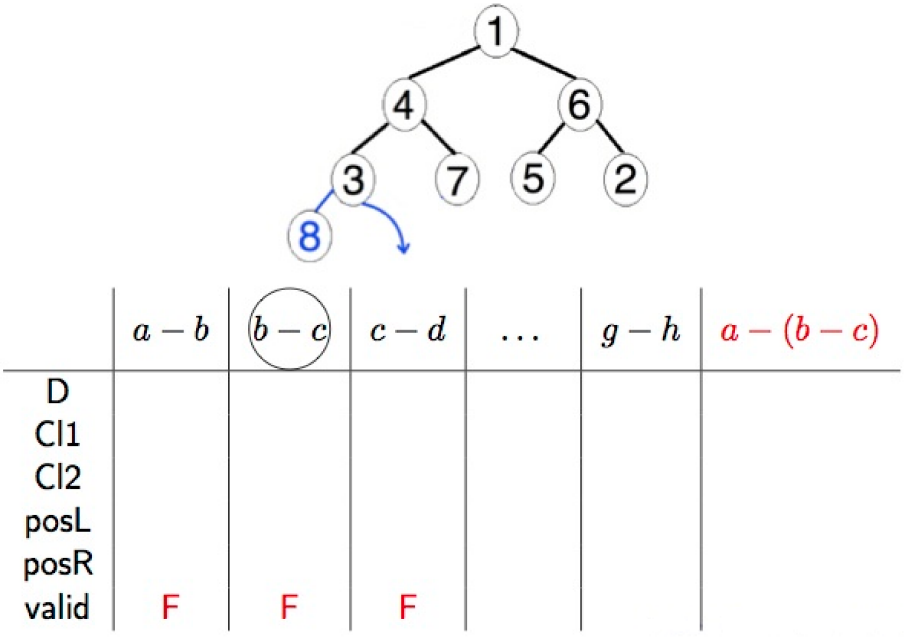


FIGURE 5.10: The data structures used in the `cWard` algorithm and the relationship between them. The chained array contains information about the pairs of adjacent clusters which are candidates to fusion (bottom panel): the Ward's distance between them ("D"), the items contained in the first cluster of the pair ("Cl1"), the items contained in the second cluster of the pair ("Cl2"), the position of the left-neighbor of the pair ("posL"), the position of the right-neighbor of the pair ("posR"), and the validity of the fusion ("valid"). The positions in the chained array of these potential fusions are stored in a min-heap (top panel) according to their corresponding distances.

More particularly, let consider 8 items $\{a, b, c, d, e, f, g, h\}$ to be clustered using the `cWard` algorithm. The proposed implementation of this algorithm starts by storing information about the pairs of adjacent clusters which are candidates to fusion: the Ward's distance between them (first arrow of the array in Figure 5.10), the items contained in the first cluster of the pair (second arrow), the items contained in the second cluster of the pair (third arrow), the position of the left-neighbor of the pair (fourth arrow), the position of the right-neighbor of the pair (fifth arrow) and finally the validity of the fusion (last arrow). This last logical parameter is initialized to `TRUE`, which means that the fusion of the corresponding pair of clusters is still possible. In addition, the positions in the chained array of the potential fusions $\{1, 2, \ldots, 7\}$ are stored in a min-heap according to the corresponding distances of the fusions. For instance, in Figure 5.10, the node of the position 4 has as children the positions 3 and 7 because the distance between the clusters

d and e (at position 4 in the chained array) is less than or equal to each of the distances between the clusters c and d (at position 3 in the chained array) and the clusters g and h (at position 7 in the chained array).

Now suppose that, at the first step of Algorithm 5, the pair of clusters $\{b, c\}$ (black-circled in Figure 5.10) appears to be the best fusion in the sense of the Ward's criterion. Then, (i) the position in the chained array of this best fusion (2 in the example of Figure 5.10) is removed from the min-heap (ii) the new two potential fusions $a - (b - c)$ and $(b - c) - d$ are added at the end of the chained array with all information about these fusions (iii) the positions in the chained array of these new two fusions are added to the min-heap (iv) the arrow "valid" of the chained array corresponding to the fusions $a - b$, $b - c$ and $c - d$ are set to `FALSE` given that the singletons b and c no longer exist and therefore these fusions are no longer possible.

5.3.4 Implementation and complexity of the `cWard` algorithm

Algorithm 8 The **optimized** algorithm of the Ward's constrained hierarchical clustering applied to a h-band similarity matrix

procedure cWARD($\mathbf{X} \in \mathbb{R}^{n\times p}$, Sim, h)
  Calculate the two $p \times h$ arrays of pencils sums ▷ $\mathcal{O}(ph)$
  Initialize the chained array Tab
  $heap \leftarrow$ `buildHeap`$(1 : (p-1), D)$ ▷ $\mathcal{O}(p\log(p))$
  $jj \leftarrow p$
  for $step = 1$ **to** $p - 1$ **do**
    while $(!Tab[valid, heap[1]])$ **do**
      $heap \leftarrow$ `deleteMin`$(heap)$ ▷ $\mathcal{O}(\log(p))$
    end while
    $posMin \leftarrow heap[1]$
    $i^\star \leftarrow Tab[Cl1, posMin]$
    $heap \leftarrow$ `deleteMin`$(heap)$ ▷ $\mathcal{O}(\log(p))$
    $d_1 \leftarrow D(C_{i^\star-1}, C_{i^\star} \cup C_{i^\star+1})$ ▷ $\mathcal{O}(1)$
    $d_2 \leftarrow D(C_{i^\star} \cup C_{i^\star+1}, C_{i^\star+2})$
    Add the distances d_1 and d_2 to Tab
    $heap \leftarrow$ `insertHeap`$(heap, jj, D)$ ▷ $\mathcal{O}(\log(p))$
    $heap \leftarrow$ `insertHeap`$(heap, jj + 1, D)$ ▷ $\mathcal{O}(\log(p))$
    Update the neighbors of $C_{i^\star-1}$ and $C_{i^\star+2}$ in Tab
    Set $Tab[valid, posMin]$, $Tab[valid, posL]$ and $Tab[valid, posR]$ to `FALSE`
    $jj \leftarrow jj + 2$
  end for
end procedure

The detailed steps of the optimized `cWard` algorithm applied to a h-band similarity matrix are presented in Algorithm 8. First, the sums of similarity measures within pencil-shaped areas are

assessed and stored in two arrays of sizes $p \times h$ in order to be used later to calculate any distance between any pair of clusters, as described in Section 5.3.2 (line 3). Then, as for a classical adjacency-constrained hierarchical clustering algorithm, the optimized implementation of the `cWard` algorithm starts with each of the p items in a separate cluster. Thus, the chained array Tab is initialized with the first $p-1$ potential fusions $1-2$, $2-3$, $\ldots$, $(p-1)-p$ and the corresponding information about it (line 4). For instance, the first column of Tab related to the fusion $1-2$ is initialized to the vector $(1-\mathrm{Sim}(1,2); 1; 2; -1; 2; \texttt{TRUE})$. So, as the item 1 has not a left-neighbor, the arrow "posL" of the fusion $1-2$ is set to -1. Besides, given that potential fusions are successively added at each step of the classification, the size of the chained array Tab is initialized to $6 \times 3p$. Third, the positions in the chained array of the potential fusions are stored in a min-heap structure (line 5). Similarly to the chained array, the size of the min-heap is initialized to $3p$. The heap property of this latter is built according to the distance between each pair of clusters (first arrow of Tab). Finally, the counter jj, which stores the first empty column of the chained array Tab, is initialized to p and updated as the classification progresses.

At each step of the clustering, the validity of the fusion located at the root of the min-heap is first checked. If `FALSE`, that is the related fusion is not possible, then this position corresponding to the minimal distance is removed from the min-heap and the validity of the new root is re-verified (lines 8 to 10). If `TRUE`, then the merge is made effective by (i) the position of this fusion is removed from the min-heap (line 13), (ii) the new 2 distances with the newly-formed cluster are calculated using the 2 arrays of pencils sums (lines 14 and 15), (iii) these distances with the related information are added to the chained array Tab at the positions jj and $jj+1$ (line 16), (iv) these positions are added to the min-heap (lines 17 and 18), (v) the rows "posL" and "posR" of the neighbors of the newly-formed cluster are updated (line 19), (vi) the row "valid" of the merged pair of clusters and its neighbors are set to `FALSE` (line 20), (vii) the position counter jj is incremented by 2 (line 21).

Lastly, we can notice that the structure of chained array is essential for maintaining the connection between the distance between a pair of clusters and its relative position among other clusters. More particularly, the update of the rows "posL" and "posR" at each step of the clustering allows the knowledge of the neighbors of all the pair of adjacent clusters through their positions in the same array. Hence the naming *chained array* of such a structure.

The number of elementary operations necessary for the performance of Algorithm 8 equals to:

$$T_8(p,h) = \underbrace{2ph}_{\text{line 2}} + \underbrace{p-1}_{\text{line 3}} + \underbrace{p\log(p)}_{\text{line 4}} + \underbrace{\sum_{step=1}^{p-1}}_{\text{the loop}} \left(\underbrace{\log(p)}_{\text{lines 7 to 9}} + \underbrace{\log(p)}_{\text{line 12}} + \underbrace{\log(p)}_{\text{lines 16 and 17}} \right).$$

The theoretical complexity of the optimized implementation of the `cWard` algorithm is then in $\mathcal{O}(p\log(p) + ph)$.

Besides, the amount of space used by Algorithm 8, without counting the space used by the input, equals to:

$$S_8(p,h) = \underbrace{2ph}_{\text{line 2}} + \underbrace{18p}_{\text{line 3}} + \underbrace{3p}_{\text{line 4}}.$$

Thus, the space complexity of the optimized implementation of the `cWard` algorithm is in $\mathcal{O}(ph)$.

Firstly, Algorithm 8 was completely coded in `R`. Nevertheless, we noted that, in the `R` language, the operation of accessing to a value of a vector is not executed in a constant time. More importantly, this operation becomes quadratic in time when integrated within a for/while loop. For these reasons, we chose to implement the main loop of the optimized algorithm in `C` (lines 6 to 21 of Algorithm 8) and to interface it with the remaining steps of the algorithm coded in `R` using the function `.Call`.

`.Call` is a built in `R` function designed as a way to call external code precompiled such as `C` or `C++` into a shared object file from `R`. It is one of the most basic ways to call external functions from `R` and comes with only the minimum amount of support. It can operate on the so-called `SEXP` objects, which stands for pointers to `S` expression objects. More specifically, everything inside `R` is represented as such a `SEXP` object, and by permitting exchange of these objects between the `C` (or the `C++`) language and `R`, programmers have the ability to operate directly on `R` objects. An example of the `.Call` function usage is presented in Appendix D.

5.3.5 Computation time of the optimized implementation of the `cWard` algorithm

The optimized implementation of the `cWard` algorithm described in the previous section is compared to two implementations:

Implem. \ p	64	128	256	512	1024	2048	4096
+pencils +heap	0.01	0.01	0.01	0.02	0.04	0.09	0.19
+pencils -heap	0.02	0.02	0.04	0.09	0.19	0.47	1.36
-pencils -heap	0.03	0.07	0.17	0.43	1.22	4.98	16.07

Implem. \ p	8192	16384	32768	65536	131072	262144	524288
+pencils +heap	0.38	0.70	1.27	2.48	5.15	10.09	23.04
+pencils -heap	4.12	14.50	59.37	263.79	1055.16	4220.64	16882.55
-pencils -heap	58.91	230.03	942.49	3769.97	15079.87	60319.48	241277.9

TABLE 5.4: Running times in seconds of the **+pencils +heap**, the **+pencils -heap** and the **-pencils -heap** implementations applied to randomly simulated genotype matrices of 100 individuals and p SNPs. The parameter h was set to 30 and the computation times were averaged across 20 simulation runs. The values shown in red correspond to computation times derived from the theoretical complexity (quadratic in p) and the running times for $p = 8192$ SNPs of the **+pencils -heap** and **-pencils -heap** implementations.

- **-pencils -heap**, which is described in Section 5.2.1. To this implementation was integrated the user-parameter h. Thus the similarities calculated on the fly are computed only if the lag between the two items does not exceed h.
- **+pencils -heap**, which uses the pencils' trick but not the min-heap and the associated chained array structures. To this implementation was also integrated the user-parameter h. The pencils sums are then calculated on a h-band similarity matrix.

Using this notation, the optimized implementation described in Algorithm 8 can be referred to as **+pencils +heap**.

In addition to the **-pencils -heap** implementation described in Section 5.2.1, the optimized implementation of the `cWard` algorithm has been added to the `R` package `BALD`. A detailed description how to use the different functions of this package is available in Appendix E.

Note that running the three implementations on the same set of items and with the same value of h leads to the same clustering tree.

The scalability for high-dimensional data of Algorithm 8 compared to the two other implementations can be assessed from Table 5.4, where we measure the computation time, averaged across 20 runs, of the **+pencils +heap**, **+pencils -heap** and **-pencils -heap** implementations applied to randomly simulated genotype matrices of 100 individuals and p SNPs. The parameter h was

set to 30. The computation times shown in red in Table 5.4 were derived from the theoretical complexity of each implementation (quadratic in p for the **+pencils -heap** and the **-pencils -heap** implementations) and using computation times for 8192 SNPs. Indeed, they correspond to running times greater than 3 minutes and it would then take more than an hour to assess the average computation time across the 20 simulated matrices for only one value of p.

The computation times from $p = 64$ to $p = 16384$ are illustrated in Figure 5.11.

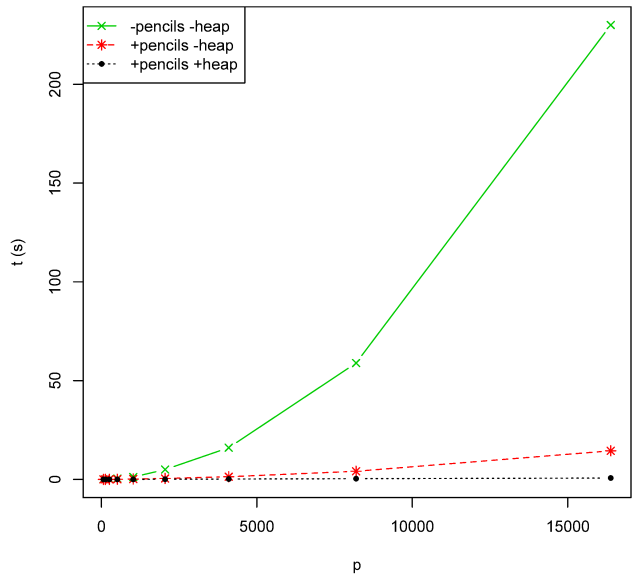


FIGURE 5.11: The running time (in seconds) as a function of the number of SNPs p, for each of the three implementations applied to randomly simulated genotype matrices of 100 individuals and p SNPs. The parameter h was fixed at 30. The running times were averaged across 20 runs.

Algorithmically speaking, the two implementations **+pencils -heap** and **-pencils -heap** are quadratic in p but Figure 5.11 shows that the pencils' trick used in the **+pencils -heap** implementation makes the clustering algorithm much more efficient. Furthermore, Figure 5.11 shows the efficiency of the **+pencils +heap** implementation in terms of computation time compared the two other implementations due to the improvement from a complexity of $\mathcal{O}(p^2)$ to a complexity of $\mathcal{O}(p \log(p) + ph)$. This difference in running times is even more important for high values of p (see Figure 5.12). For instance, it takes around 23 seconds for the optimized **+pencils +heap** implementation to cluster 524288 SNPs while this running time is $\sim 150\%$ longer for the **+pencils -heap** implementation to classify only 32768 SNPs and for the **-pencils -heap** implementation to analyze 8192 markers.

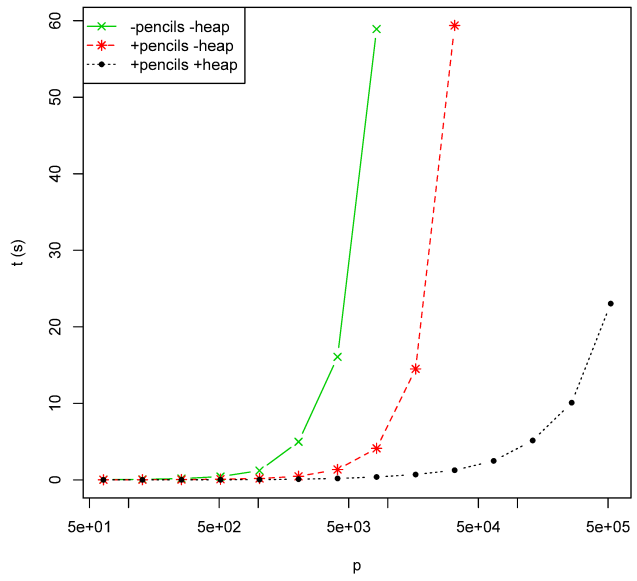


FIGURE 5.12: Difference in computation times between the three implementations for high values of p. The parameter h was fixed at 30. The running times were averaged across 20 runs.

To extend the comparison with the computation time cited in Section 5.2.2, the optimized implementation **+pencils +heap** takes now 9.92 seconds and 38.8 seconds to cluster a whole genome of $500k$ simulated SNPs genotyped on 100 individuals with $h = 30$ and $h = 100$ respectively. Consequently, unlike the quadratic complexity of the former implementations, the time complexity of the optimized implementation presented in Algorithm 8 is adapted to high-dimensional data.

The number of individuals genotyped n also influences the complexity of the adjacency-constrained hierarchical clustering using LD similarity. Indeed, it is involved in the calculations of the two arrays of pencils sums in the implementations **+pencils +heap** and **+pencils -heap**, and intervenes at each LD calculation within the main clustering loop in the implementation **-pencils -heap**. Nevertheless, more generally, the impact of the number of individuals on the complexity of the adjacency-constrained hierarchical clustering algorithm remains difficult to determine as it is specific to the similarity used, and thus to the way in which is implemented such a similarity. Concerning the LD similarity, our experiments show that, thanks to an optimized implementation of LD calculations in the `snpStats` package, the influence of n is less critical than that of the number of SNPs p. Indeed, as shown in Table 5.5, increasing n by a factor of 10 increases the total run time of the method by a factor of ~ 3.5, implying a sublinear complexity of the three implementations in n. The optimized implementation described in Algorithm 8 presents,

Implem. \ p	64	128	256	512	1024	2048	4096
+pencils +heap	0.02	0.03	0.05	0.09	0.2	0.39	0.67
+pencils -heap	0.02	0.03	0.08	0.13	0.30	0.73	1.82
-pencils -heap	0.06	0.16	0.55	1.68	5.61	20.6	75.82

TABLE 5.5: Running times in seconds of the **+pencils +heap**, the **+pencils -heap** and the **-pencils -heap** implementations applied to randomly simulated genotype matrices of 1000 individuals and p SNPs. The parameter h was set to 30 and the computation times were averaged across 20 simulation runs.

nonetheless, the advantage of separating the operations of similarity calculations from the main clustering loop. This allows to locate the influence of n wich intervenes only while running the line 2 operation.

In order to evaluate the influence of the parameter h on the clustering results of Algorithm 8, while considering a realistic dependance structure among the items, we used the $p = 20756$ SNPs of chromosome 6 for $n = 605$ individuals from the GWA study on HIV (Dalmasso et al. 2008). We first applied the `cWard` algorithm using LD similarity and with $h = p$, followed by the Gap statistic approach. These two steps estimated a "true" block structure including 1756 blocks of SNPs with the following distribution of sizes:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2	7	11	11.82	16	51.

We then compared this block structure to that estimated by applying Algorithm 8 with $h < p$ followed by the Gap statistic approach, for different values of h. We noticed that the true block structure and that inferred with $h < p$ are identical as soon as $h \geq 45$. These observations can be explained by the fact that the lower value h_0 of h leading to the same block structure as with $h = p$ is closely linked to the maximum size of the "true" blocks. This result is also consistent with the assumption that h_0 is several orders of magnitude smaller than the number of items to be clustered.

Finally, this chapter illustrates well the tradeoff between minimizing the complexity in time and space of an algorithm. Indeed, starting from an adjacency-constrained hierarchical clustering algorithm quadratic in time and in $\mathcal{O}(p)$ in space (if the similarities are calculated on the fly), we could obtain an implementation-level description of the same algorithm that has a complexity

of $\mathcal{O}(p\log(p) + ph)$ in time and of $\mathcal{O}(ph)$ in space. These improvements allow thus to be sub-quadratic both in time and space, which was the main objective to enable the algorithm to scale high-dimensional data.

5.4 Conclusions

In this chapter, we have proposed an efficient implementation of the adjacency-constrained hierarchical clustering algorithm according to Ward's criterion and using a band similarity matrix. This work is in fact an improvement of the approach proposed in the preceding chapter. Indeed, an adjacency-constrained hierarchical clustering of the genetic markers have been performed as a first step for inferring the LD blocks. However, given that such an algorithm is intrinsically quadratic in time in the number of SNPs, it appeared to be not adapted to the high-dimensionality of GWAS data.

An efficient implementation of such an algorithm has then been proposed in the general context of any similarity measure. This implementation assumes that items which are far apart have a null similarity between them. This property is reflected in a user-defined parameter $h \ll p$ such that the similarity between two items distant from more than h is set to 0, leading to a h-band similarity matrix. By means of (i) a "simple" expression of the Ward's criterion using the kernel trick (ii) a pre-calculation of the distances between all pairs of clusters thanks to the pencils' trick and (iii) the min-heap structure, we could reduce the time complexity of the adjacency-constrained hierarchical clustering algorithm to $\mathcal{O}(p\log(p) + ph)$ while keeping its space complexity linear in p.

The interest of the proposed implementation has been illustrated by applying it to simulated genotype matrices with an increasing number of SNPs to be clustered. It indeed allows a dramatic reduction on the computation time of the adjacency-constrained hierarchical clustering algorithm compared to two former implementations. Furthermore, the results demonstrate the scalability of the proposed implementation to high-dimensional data.

A potential criticism of the implementation proposed in this chapter could be the "arbitrary" choice of the parameter h. Generally speaking, h can be chosen according to the fact that this parameter corresponds to the maximum size of the blocks the user would have in his data. More specifically, when dealing with the linkage disequilibrium similarity, the user's choice can be guided by the lengths in kb of the LD blocks found out in previous studies (see references cited in Table 3.1). Indeed the extent of the linkage disequilibrium varies from population to population and from one chromosome to another. For instance, the sizes of the LD blocks identified by Gabriel et al. (2002) varied from $<$ 1kb to 173kb in European samples. With an average density of 1 SNP each 2kb, an LD block then contains at most 86 SNPs.

Chapter 6

Conclusions and perspectives

The advent of high-throughput genotyping technologies, including SNP genotyping chips, was a key turning point for the genetic analysis of multifactorial diseases. This analysis is a major public health issue (diagnosis, prevention, therapy, ...). In this context, the Genome-wide association studies (GWAS) have emerged as relevant approaches for the precise localization of genetic markers involved in the mechanisms of these diseases. Unfortunately, the high complexity of the data used, combined to their large volume are, among others, tricky issues that have raised doubts about the relevance of these studies' findings.

This PhD work focused on GWAS and their related problematics. It aimed to provide indications and guidelines that answer to some questions raised by the analysis of complex genetic data and to develop both statistical methods and software tools that allow to improve certain aspects of GWAS.

This final chapter is dedicated to presenting the main points that we evoked in this manuscript as well as the conclusions regarding the methodological and software developments that we conducted.

6.1 General conclusions

The work carried out in this PhD is at the interplay of three main scientific fields: statistics, genetics and computer science. Thus, in order to facilitate the understanding of the manuscript, we paid particular attention to introduce the basic concepts of each area in accordance with the topics tackled.

We first introduced basic statistical concepts such as hypothesis testing and multivariate regression models. Secondly, after a presentation of essential concepts of cellular biology and fundamental notions of genetics, we could define the linkage disequilibrium, its most common measures, two approaches to estimate it, and the group structure it induces in the human genome. Thirdly, Genome-wide association studies have been introduced through the type of data and analytical model used in these studies. Finally, through the presentation of haplotype association studies, emphasis has been put on the importance of taking into account the block-like structure of the human genome induced by linkage disequilibrium, in order to identify sets of loci that are associated with complex traits. It was then concluded that LD constitutes a central parameter in both the design and the proper conduct of GWAS. More importantly, incorporating LD information could improve the power of these studies.

Several LD block partitioning approaches have been proposed in order to infer the linkage disequilibrium structure underlying the human genome. As a reminder, these methods can be classified into two main groups: those that make use of pairwise LD measures to distinguish high LD regions from recombination hotspots and those that define blocks as regions with limited haplotype diversity. A comprehensive list of existing definitions of LD blocks have been presented with a detailed description of the most commonly used methods of both classes. The absence of information about haplotypes from unphased GWAS data on one hand, and the subjective and arbitrary choices usually made on the thresholds of pairwise LD approaches on the other hand, led us to present the first contribution of this manuscript. An LD block partitioning method which consists in: (i) performing a spatially-constrained hierarchical clustering using Ward's linkage criterion and LD similarity (ii) applying the Gap statistic approach to the obtained hierarchy to estimate the number of groups. After an introduction to cluster analysis and model selection approaches, a detailed description of the two steps of the proposed LD block partitioning method have been presented. In conclusion, the choice of a statistical and automated approach for inferring LD blocks was motivated by the aim of improving the power of GWA studies by incorporating the estimated block structure in a block-based regression model.

The remaining contributions of this manuscript can be divided in two phases. A first methodological phase focused on incorporating LD block information in GWAS, followed by a second practical phase that improves the computational performances of the method proposed in the first phase.

The methodological contribution was to propose an approach that selects sets of markers that

are associated with a phenotype of interest. This method, called BALD, consists in inferring the LD blocks using the two steps described earlier and then identifying associated groups of SNPs using the Group Lasso regression model. The BALD approach is based on the idea of taking advantage of the block structure induced by LD and select associated markers that could have been missed by single-marker analysis by explicitly looking for sets of LD blocks jointly associated with the phenotype of interest. We investigated the efficiency of the proposed approach compared to state-of-the-art regression methods: three non grouping methods that select associated SNPs –single-marker analysis, Lasso and Elastic-Net– and two grouping methods that select sets of associated markers –the haplotype association module of PLINK and the Group Lasso applied to the true SNP groups–. Our results on simulated data showed that BALD performs better than state-of-the-art approaches as soon as the number of causal SNPs within an LD block exceeds 2. Furthermore, our results on semi-simulated and real data illustrated the robustness of the proposed method to a real LD structure. Thus, our three-step method reaches satisfying performances both at the level of LD blocks by inferring well the underlying block structure but also in terms of SNP selection. Furthermore, these results highlight the importance of incorporating biological knowledge in high-dimensional genomic studies such as GWAS.

The practical contribution of this manuscript consisted in the design and the implementation of a generalized and efficient spatially-constrained hierarchical clustering algorithm called `cWard`. The idea of introducing such an algorithm arose from the observation that the computation times of the adjacency-constrained hierarchical clustering procedure used in the BALD, and more generally its quadratic time complexity, were not adapted to GWAS data. The improvements made to the `cWard` algorithm were nonetheless presented in the general context of any similarity measure, and not necessarily the LD similarity. A user-parameter h, which controls the maximum lag between items for similarity calculations, was first introduced. By means of the pencils' trick, we could make the complexity of calculating the distance between any pair of clusters constant in time and obtain a space complexity in $\mathcal{O}(ph)$ of the `cWard` algorithm, where p is the number of items to be clustered. Then, by using the min-heap structure, we could make the operation of finding the best fusion sublinear (logarithmic) in time. These two improvements resulted in a clustering algorithm with a time complexity of $\mathcal{O}(p\log(p) + ph)$. The efficiency of this novel algorithm were investigated with applications to simulated SNP data, by comparing its computation times to those of former algorithms that did not include some or all of the implementation improvements used in `cWard`. Our results showed that the sub-quadratic complexity of the proposed implementation allowed a dramatic reduction on the computation times of the

spatially-constrained hierarchical clustering algorithm, especially for high values of p. These results also demonstrated the scalability of the `cWard` algorithm to high-dimensional data.

The work carried out in the framework of this PhD has been accompanied by the development of the R package `BALD`, available at the webpage `http://www.math-evry.cnrs.fr/logiciels/bald`. In addition to the application of the BALD method and the `cWard` algorithm presented in this manuscript, this package allows generating structured GWAS data, that is genotypes of SNPs with a group structure along the genome and continuous phenotypes associated with these genotypes. It also provides functions for the use of several existing regression methods via a unified interface, the evaluation of their performances in terms of variable selection using ROC curves and a graphical representation of the results. These software developments put the emphasis on the importance of furnishing software tools that allow to conduct a GWA study and that are computationally adapted to the high-dimensionality of the data.

Eventually, the different results that we obtained allowed to draw some interesting conclusions regarding certain aspects of the conduct of GWA studies that could be added to the many research and discussions on the topic. In addition, we designed a novel clustering algorithm that showed promising results and is susceptible to be further used to cluster either genetic or other types of data.

6.2 Perspectives

The research work that we presented in this manuscript has led to several conclusions and also pointed out some interesting research perspectives.

6.2.1 SNP/block-level p-values through hierarchical testing

The proposed BALD approach showed promising results in selecting sets of SNPs that are associated with the phenotype. Nevertheless, a limitation of this method is that it does not provide a significance assessment for the selected groups. Deriving reliable p-values for regression coefficients in high-dimensional, correlated settings is a challenging research area in the machine learning and statistics fields in general (Bühlmann 2013, Chatterjee & Lahiri 2011). However, even if such p-values could be obtained for the groups inferred by the BALD method, we would like to emphasize that providing an interpretable multiple testing risk assessment in GWAS

would remain a difficult question. Although several multi-SNP tests have been proposed to assess the significance of SNP groups (Kwee et al. 2008, Li et al. 2011), no fully satisfactory strategy allows the control of standard multiple testing error rates such as the FWER or the FDR.

Indeed, the presence of correlation among explanatory variables makes causal SNPs indistinguishable from their "neighbors". This issue is not specific to a particular inference method, but intrinsic to the design of GWAS. Therefore, we believe that it should be addressed by adapting the definitions of true and false positives. In Chapter 4, we have considered two types of risk evaluation at different *genomic scales*: SNP-level and block-level evaluations. An alternative strategy in a similar spirit was recently proposed (Yi et al. 2015). Both strategies rely on a prior definition of the scale of the signal of interest. A future research topic then could be to develop an evaluation strategy and an associated inference method that adapts to this scale. A possible direction is to adapt the notion of hierarchical testing of variable importance (Meinshausen 2008, Mandozzi & Bühlmann 2015) to the specific context of GWAS.

6.2.2 Model selection approach

Within the BALD method, we considered the Gap statistic approach to estimate the optimal number of clusters as it provided the best empirical results. Nevertheless, despite the improvements brought to the `cWard` algorithm, the Gap-step remains the computational bottleneck of the proposed three-step approach. Thus, investigating other strategies to estimate the number of clusters could lead to finding an approach that would faster the BALD method.

Inspired from Lévy-Leduc et al. (2014), a future research topic could be to take advantage of the block-like structure of the similarity matrix of the data by considering its log-likelihood at each step of the spatially-constrained clustering. Indeed, under certain assumptions on the means and variances of the similarity measures within and outside the diagonal blocks, and by considering the gain of log-likelihood as distance between two clusters, it can be observed that the log-likelihood curve with respect to the number of blocks of the partition presents a change-point which corresponds to the optimal number of clusters. Thus, by applying an adjacency-constrained hierarchical clustering algorithm according to the log-likelihood criterion, the number of clusters could be estimated both with the clustering tree.

6.2.3 Rare variants analysis

With the advent of new generation sequencing technologies, biologists are increasingly focusing on rare variants that could be involved in the biological mechanisms of common diseases. Since associations highlighted by Genome-wide association studies can account for only a small fraction of the heritability of deseases, rare variants analysis could have the potential to explain some of this missing heritability.

An important issue that raises with rare variants is finding out their associations with diseases. Indeed, the low frequency of these variants renders standard methods used to test for associations underpowered unless sample sizes or effect sizes are very large. Consequently, novel association tests have been specifically designed to overcome this limitation (Morris & Zeggini 2010). Furthermore, the issue of linkage disequilibrium may also be taken into account in a different way. Some methods for rare variant analysis assume that rare variants are not in LD whereas other methods relax this assumption, including methods designed to jointly assess common and rare variants (Yuan et al. 2012).

6.2.4 The `cWard` algorithm

The algorithm that we presented in Chapter 5 has shown promising performances. We think that it could be improved by continuing working on certain aspects.

The `cWard` algorithm can be made computationally even more efficient in two ways. First, a recurrence relationship between the sum of similarities within pencils of parameters (`hLoc`, `lim`, `"right''`), (`hLoc+1`, `lim`, `"right''`), (`hLoc`, `lim+1`, `"right''`) and (`hLoc+1`, `lim+1`, `"right''`) can be highlighted. An equivalent recurrence relationship can be deducted for `"left''` pencils. These relationships would allow to reduce the time complexity of the proposed algorithm.

Besides, the pre-calculated sums of similarities within pencils areas are probably not all used in the clustering process. A possible improvement is then to calculate the pencils sums needed on the fly as the clustering progresses. This can reduce the space complexity of the `cWard` algorithm.

Another perspective with the `cWard` algorithm is its application to other types of data. Indeed, the current implementation of the `cWard` function is designed to be applied to SNP data as it

takes as input a genotype matrix. Since the improvements presented in Chapter 5 do not depend on the similarity considered, developing a generalized implementation `cWard` that takes as input a similarity matrix could render the software suitable for other clustering applications and not only for SNP clustering. One possible direction is to apply this implementation to Hi-C data (Lieberman-Aiden et al. 2009).

Appendix A

Bioinformatic resources for GWAS and haplotype analysis

A.1 Databases

dbSNP. The dbSNP database is a continuously updated public database that lists the genetic variations found in different animal species including Homo sapiens. Despite its name, this database lists not only the SNPs but also other genetic variants such as indels or micro-satellites. A summary of all available data in *dbSNP* on the different species are available at the link http://www.ncbi.nlm.nih.gov/SNP/snp_summary.cgi.

1000genomes Project. The 1000genome project (http://www.1000genomes.org/) (Consortium et al. 2010) was born in order to facilitate the analysis and research of low frequency SNPs. This project was launched in 2008 with the goal of creating a public reference database for DNA polymorphism by sequencing the entire genome of 2500 individuals from 28 different populations. These data are being produced and are expected to soon have a more complete mapping of human variants at MAF 1% and below, their haplotypes and the linkage disequilibrium between them.

International HapMap Project. The International Haplotype Map Project (HapMap: (http://hapmap.ncbi.nlm.nih.gov/) (Consortium et al. 2005), launched in 2002, aimed to create a map of 1 million common SNPs (defined as those whose the MAF is at least 5%), with

not only their genomic locations but also genotype frequencies and LD relationships among each other, in three populations (Europeans, Africans and East Asians). In Phase I, the project involved the genotyping of 1 million SNPs in 270 individuals (90 from each of the three populations). In Phase 2, the HapMap had been extended to include over 3 million SNPs on the same samples (Frazer et al. 2007). And those samples plus additional ones were later genotyped using the latest SNP chip technology in Phase 3 of the project. The HapMap provides not only a very high density of common SNPs which can be used as markers in association studies, but also provides their location in the genome and how they are distributed within populations and among populations in different parts of the world. The other goal of the HapMap project was to describe the dependency relationships between SNPs using linkage disequilibrium information and thus identify common haplotypes (with frequency $> 5\%$).

A.2 Software

PLINK. PLINK is a free, open-source whole genome association analysis toolset, designed to perform a range of basic, large-scale analyses in a computationally efficient manner (Purcell et al. 2007). It can be downloaded for Linux, MS-DOS and Mac OS systems at the PLINK webpage `http://pngu.mgh.harvard.edu/~purcell/plink/`. PLINK provides functions, among other tools, for genotype data management, simple marker association analysis, LD calculations, multi-marker and haplotype association analysis.

Furthermore, PLINK can be used in combination with R to take advantage of an array of statistical tools such as the R package `snpStats`. Furthermore, the R packages `SNPassoc` (González et al. 2007) and `GenABEL` (Aulchenko et al. 2007) are also designed specifically to handle genome-wide association data.

snpStats. The R package `snpStats` (extending the `snpMatrix` package) (Clayton 2012), which is a component of the BioConductor open-source software project, provides classes and statistical methods for large SNP association studies, with the possibility of reading and creating input and output PLINK files. More particularly, this package comprises efficient implementations for LD calculations and SNP data imputation.

Haploview. Haploview (Barrett et al. 2005) is an open-source software package that provides computation of linkage disequilibrium statistics and population haplotype patterns from primary genotype data in an interactive interface. Haploview can also perform association studies and allows choosing tagSNPs and estimating haplotype frequencies. Finally, it provides a convenient viewer for PLINK results generated from genome-wide association studies. It can be downloaded for Linux, MS-DOS and Mac OS systems at the Haploview webpage http://www.broad.mit.edu/mpg/haploview/.

Appendix B

Ward's criterion

According to Equation 3.2, the Ward's criterion using the Euclidean distance equals to:

$$
\begin{aligned}
d_{\text{wl}}(A,B) &= \sum_{i\in A\cup B} \|\mathbf{X}_{.i}, \mathbf{g}_{A\cup B}\|_2^2 - \sum_{i\in A} \|\mathbf{X}_{.i}, \mathbf{g}_A\|_2^2 - \sum_{i\in B} \|\mathbf{X}_{.i}, \mathbf{g}_B\|_2^2 \\
&= \sum_{i\in A\cup B} \left(\|\mathbf{X}_{.i}\|_2^2 - 2\mathbf{g}_{A\cup B}^\top \mathbf{X}_{.i} + \|\mathbf{g}_{A\cup B}\|_2^2\right) - \sum_{i\in A} \left(\|\mathbf{X}_{.i}\|_2^2 - 2\mathbf{g}_A^\top \mathbf{X}_{.i} + \|\mathbf{g}_A\|_2^2\right) \\
&\quad - \sum_{i\in B} \left(\|\mathbf{X}_{.i}\|_2^2 - 2\mathbf{g}_B^\top \mathbf{X}_{.i} + \|\mathbf{g}_B\|_2^2\right) \\
&= 2\sum_{i\in A} \mathbf{X}_{.i}^\top (\mathbf{g}_A - \mathbf{g}_{A\cup B}) + 2\sum_{i\in B} \mathbf{X}_{.i}^\top (\mathbf{g}_B - \mathbf{g}_{A\cup B}) \\
&\quad -p_A\|\mathbf{g}_A\|_2^2 - p_B\|\mathbf{g}_B\|_2^2 + (p_A+p_B)\|\mathbf{g}_{A\cup B}\|_2^2 \\
&= 2(\mathbf{g}_A - \mathbf{g}_{A\cup B})^\top p_A\mathbf{g}_A + 2(\mathbf{g}_B - \mathbf{g}_{A\cup B})^\top p_B\mathbf{g}_B - p_A\mathbf{g}_A^\top\mathbf{g}_A - p_B\mathbf{g}_B^\top\mathbf{g}_B \\
&\quad +(p_A+p_B)\|\mathbf{g}_{A\cup B})\|_2^2.
\end{aligned}
$$

Using the relationship:

$$
\mathbf{g}_{A\cup B} = \frac{p_A\mathbf{g}_A + p_B\mathbf{g}_B}{p_A+p_B},
$$

we obtain:

$$
\begin{aligned}
d_{\text{wl}}(A,B) &= 2\frac{(p_B\mathbf{g}_A - p_B\mathbf{g}_B)^\top}{p_A+p_B}p_A\mathbf{g}_A + 2\frac{(p_A\mathbf{g}_B - p_A\mathbf{g}_A)^\top}{p_A+p_B}p_B\mathbf{g}_B - p_A\mathbf{g}_A^\top\mathbf{g}_A - p_B\mathbf{g}_B^\top\mathbf{g}_B \\
&\quad + \frac{p_A^2}{p_A+p_B}\|\mathbf{g}_A\|_2^2 + \frac{p_B^2}{p_A+p_B}\|\mathbf{g}_B\|_2^2 + 2\frac{p_Ap_B}{p_A+p_B}\mathbf{g}_A^\top\mathbf{g}_B \\
&= 2\frac{p_Ap_B}{p_A+p_B}(\mathbf{g}_A - \mathbf{g}_B)^\top\mathbf{g}_A + 2\frac{p_Ap_B}{p_A+p_B}(\mathbf{g}_B - \mathbf{g}_A)^\top\mathbf{g}_B - p_A\|\mathbf{g}_A\|_2^2 - p_B\|\mathbf{g}_B\|_2^2 \\
&\quad + \frac{p_A^2}{p_A+p_B}\|\mathbf{g}_A\|_2^2 + \frac{p_B^2}{p_A+p_B}\|\mathbf{g}_B\|_2^2 + 2\frac{p_Ap_B}{p_A+p_B}\mathbf{g}_A^\top\mathbf{g}_B \\
&= 2\frac{p_Ap_B}{p_A+p_B}\|\mathbf{g}_A - \mathbf{g}_B\|_2^2 - \frac{p_Ap_B}{p_A+p_B}\|\mathbf{g}_A\|_2^2 - \|\mathbf{g}_B\|_2^2 + 2\frac{p_A^2}{p_A+p_B}\mathbf{g}_B^\top\mathbf{g}_A \\
d_{\text{wl}}(A,B) &= \frac{p_Ap_B}{p_A+p_B}\|\mathbf{g}_A - \mathbf{g}_B\|_2^2
\end{aligned}
$$

Appendix C

Sums of similarities within pencil-shaped areas

Figure C.1 shows that the complementary of the half of a right-oriented (resp. left-oriented) pencil-shaped area within a diagonal band of width $hLoc$ is a rectangle-shaped area located at the bottom (resp. top) of the h-band similarity matrix. The sum of similarity measures within this type of shape can be more easily calculated than a within pencil-shaped areas by using optimized `R` functions as `rowCumsums` and `colCumsums` which are included in the package `matrixStats`. The `R` function `rowCumsums` (resp. `colCumsums`) calculates the cumulative sums for each row (resp. column) of a given matrix.

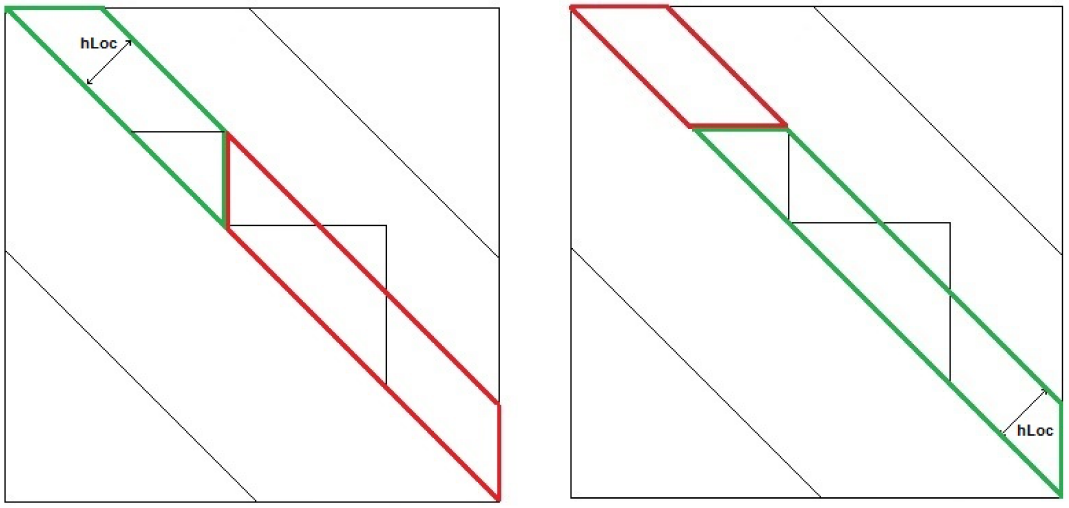
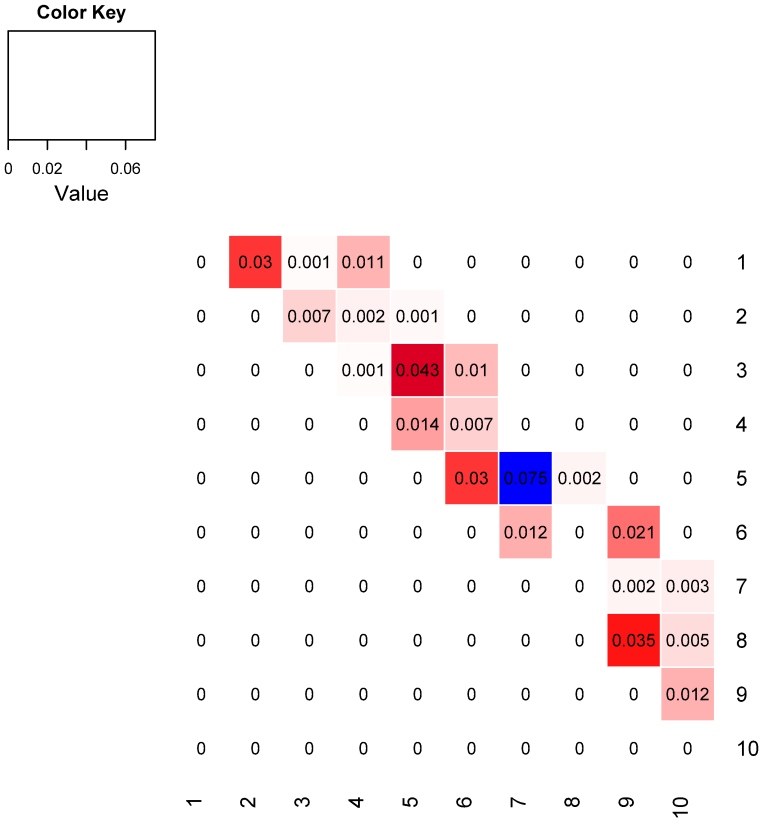


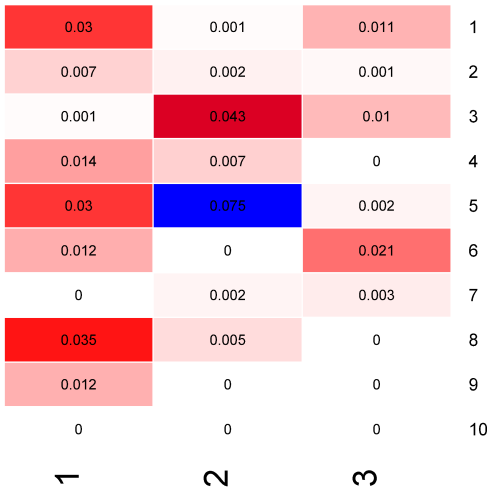
FIGURE C.1: A schematics of the upper side of h-band similarity matrix. The complementary of both right-oriented pencil-shaped area (left panel, in green) and left-oriented pencil-shaped area (right panel, in green) are rectangle-shaped areas (in red).

In order to optimize the computation of the sum of the similarity measures within rectangle-shaped areas, two functions have been implemented (`toMatLeft` and `toMatRight`) which convert the upper side of a h-band similarity matrix to a $p \times h$ matrix with zeros located on the bottom right for top rectangle-shaped areas, and zeros located on the top left for bottom rectangle-shaped areas. The outputs of these two functions are illustrated in Figure C.2.

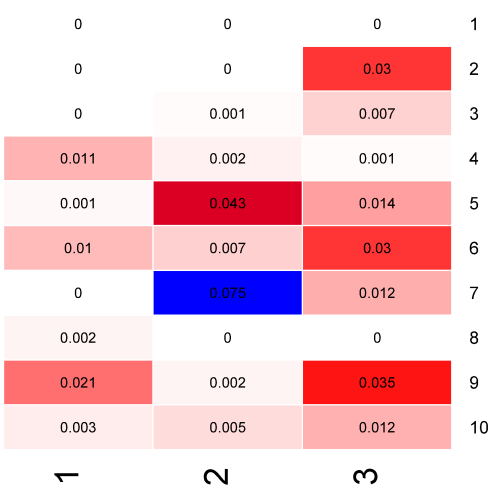
Lastly, applying the `R` functions `rowCumsums` and `colCumsums` one after the other to the top rectangle (resp. to a rotated bottom rectangle) allows to obtain the sum of the elements of all the top rectangles (resp. bottom rectangles) of depth $hLoc \in \{1, \ldots, h\}$ and limit $lim \in \{1, \ldots, p\}$. For instance, given the $p \times h$ similarity matrix M with zeros located on the bottom right, the sum of the similarity measures contained in the top rectangle of deph $hLoc$ and limit lim correponds to the element `rowCumsums(colCumsums(M))[lim,hLoc]`.



(A) upper side of the h-shrinked matrix



(B) top rectangle



(C) bottom rectangle

FIGURE C.2: The output rectangles of the functions `toMatLeft` (left panel) and `toMatRight` (right panel).

Appendix D

R's C interface

Generally, calling a C function from R requires two pieces: a C function and an R wrapper function that uses .Call.

Let take the example of the buildHeap function wich allows to build a min-heap starting from an unordered vector of values (see Section 5.3.3). The wrapper R function is then buildHeap which calls the percDown function coded in C.

```
// ------- In C --------

#include <stdio.h>
#include <R.h>
#include <Rinternals.h>
#include <Rmath.h>
SEXP percDown(SEXP Rpositions, SEXP Rdistances, SEXP Rl, SEXP Rpos){
  int mc, right, left;
  int *positions, *l, *pos;
  double *distances, tmp, val;
  Rpositions = PROTECT(coerceVector(Rpositions, INTSXP));
  positions = INTEGER(Rpositions);
  distances = REAL(Rdistances);
  l = INTEGER(Rl);
  pos = INTEGER(Rpos);
  *pos = *pos - 1;
```

```
  val = distances[positions[*pos]-1];

  while((2**pos+1) < *l) {
    if ((2**pos+2) == *l){
      left = 2**pos+1;
      if (val > distances[positions[left]-1]) {
// swap positions
tmp = positions[*pos];
positions[*pos] = positions[left];
positions[left] = tmp;
// update pos
*pos = left;
      }
      else
*pos = *l;
    }
    else {
      left = 2**pos+1;
      right = 2**pos+2;
      mc = right;
      if (distances[positions[left]-1] < distances[positions[right]-1])
mc = left;

      if (val > distances[positions[mc]-1]) {
// swap positions
tmp = positions[*pos];
positions[*pos] = positions[mc];
positions[mc] = tmp;
// update pos
*pos = mc;
      }
      else
*pos = *l;
    }
```

```
  }
  UNPROTECT (1) ;
  return(Rpositions);
}
```

Note that, for any C function callable from R, all the arguments must be pointers. Also, to coerce objects at the C level, we should use `PROTECT(new = coerceVector(old, SEXPTYPE))`. If the created R objects are not protected, the garbage collector will consider them as unused and delete them. Finally, it must be ensured that every protected object is unprotected. `UNPROTECT()` takes a single integer argument n and unprotects the last n objects that were protected.

This C example code is then put in a file `percDown.c` and compiled using the command:

```
R CMD SHLIB percDown.c
```

Now the code can be dynamically loaded into R by doing:

```
   dyn.load("percDown.so")
```

The following R script is the wrapper `buildHeap` function that uses `.Call`.

```
########## In R ##########
dyn.load("percDown.so")
buildHeap <- function(positions, distances, l){
  for (ii in floor(l/2) : 1){
    positions <- .Call("percDown", positions, distances,
                       as.integer(l), as.integer(ii))
  }
  return(positions)
}
```

Its use is fairly simple, requiring the name of the C function and all the arguments being passed in. Nevertheless, it is up to the programmer to ensure that the correct arguments, with the correct types, are provided.

Appendix E

BALD Vignette

E.1 The BALD package

The `BALD` package (Blockwise Approach using Linkage Disequilibrium) arose out of the need to provide friendly data generation functions and an efficient analysis method of whole genome association studies in `R`.

With recent advances in high-throughput genotyping technology, genome-wide association studies (GWAS) have become a tool of choice for identifying genetic markers underlying a variation in a given phenotype - typically complex human diseases and traits. Whole-genome single nucleotide polymorphism (SNP) data are collected for many thousands of SNP markers, leading to high-dimensional regression problems where the number of predictors greatly exceeds the number of observations. Moreover, these predictors are statistically dependent, in particular due to linkage disequilibrium (LD).

The main function of this package `grplassoCWard` implements a *proposed three-step approach* (Dehman et al. 2015) that explicitly takes advantage of the grouping structure induced by LD: in the first step, LD blocks are inferred by performing a clustering of LD estimates with an adjcency constraint (Ward Jr 1963). In the second step, the Gap statistic model selection approach (Tibshirani et al. 2001) is applied to estimate the number of groups and finally the Group Lasso regression (Yuan & Lin 2005) is performed on the inferred LD blocks.

E.2 Generating genotype and phenotype data

Let n be the number of individuals of our GWA study and p the number of variables (SNPs). The `BALD` package allows to generate block-structured genotype data and associated continuous phenotype according to the linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{y} \in \mathbb{R}^n$ is the phenotype vector, $\mathbf{X} \in \{1, 2, 3\}^{n \times p}$ the SNP genotypes matrix and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ a gaussian error term. The columns of $\mathbf{X}$ are assumed to be block-structured on `nBlocks` non-overlapping blocks.

In order to simulate such GWAS data, we will use the two simulation functions `simBeta` and `simulation`. We will first simulate the association vector $\boldsymbol{\beta}$ using the function `simBeta` as follows:

```
set.seed(2)
blockSizes <- c(2,4,5,3,2,4)
p <- sum(blockSizes)
sig.blocks <- c(3,5)
nb.per.block <- c(2,3)
betas <- simBeta(blockSizes, sig.blocks, nb.per.block)
```

The first element of the output `betas`:

```
betas$blockSizes
```

```
## [1] 4 3 5 2 4 2
```

contains the effective block sizes used for the simulation of $\boldsymbol{\beta}$. The second element of `betas`

```
str(betas$betaMat)
```

```
##  num [1:20, 1:3] 0 0 0 0 1 -1 0 -1 -1 1 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "betaSNP" "betaBl" "groups"
```

is a 20×3 numeric matrix. The first column of the matrix contains the regression vector for the 20 predictors (SNPs) structured in `length(blockSizes)`=6 blocks. We can check that only 5 SNPs were simulated as associated which are the first two (resp. three) SNPs contained in the first block of size 3 (resp. 5).
The second column of `betas$betaMat` contains the "block regression vector".

```
betas$betaMat[,"betaBl"]
```

```
##  [1] 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
```

As we can see, all the coefficients of the predictors contained in `sig.blocks` have been simulated nonzero (TRUE).
Finally, the third column of `betas$betaMat` corresponds to the vector defining the grouping of the variables effectively used for the simulation of the regression vector $\boldsymbol{\beta}$.

Using the regression coefficients in `betas`, we can then simulate the genotype and phenotype data. We begin by fixing the number of individuals of our study `n`, the level of correlation between the SNPs of each block `corr` and the coefficient of determination of the problem `r2`:

```
n <- 50
corr <- 0.5
r2 <- 0.7
sim <- simulation(n, betas$betaMat[, "betaSNP"], betas$blockSizes,
                  corr, r2, minMAF=0.45)
X <- sim$X
str(X)
```

```
##  num [1:50, 1:20] 3 1 3 1 2 1 2 1 3 1 ...
```

```
y <- sim$y
str(y)
```

```
##  num [1:50, 1] 0.857 -1.339 -0.89 0.861 4.194 ...
```

```
sim$r2  ## effective coefficient of determination


## [1] 0.7343056
```

The range of Minor Allele Frequencies (MAF) of the simulated SNP markers can be calibrated using the options `minMAF` and `maxMAF` of the `simulation` function.

E.3 The three-step method

Now that we have generated our genotype and phenotype data, we can apply the proposed three-step method in two different ways: (i) by applying sequentially three functions corresponding to the three steps of the proposed approach or (ii) by using a generic function that comprises these three steps.

Following the first option, we first begin by applying the adjacency-constrained hierarchical clustering to the *columns* of the genotype matrix by running the `cWard` function:

```
cW <- cWard(X, h=p-1, sim=simR2, heaps=TRUE)
str(cW)


## List of 10
##  $ traceW     : num [1:19, 1:2] 1 2 3 4 5 6 7 8 9 10 ...
##  $ gains      : num [1:19] 0.648 0.653 0.67 0.716 0.729 ...
##  $ merge      : int [1:19, 1:2] -13 -6 -19 -5 -3 -17 -1 -8 -11 -15 ...
##  $ height     : num [1:19] 0.648 1.301 1.971 2.687 3.416 ...
##  $ seqdist    : num [1:19] 0.648 1.301 1.971 2.687 3.416 ...
##  $ order      : int [1:20] 1 2 3 4 5 6 7 8 9 10 ...
##  $ labels     : chr [1:20] " 1" " 2" " 3" " 4" ...
##  $ method     : chr "cWard"
##  $ call       : language .cWLD(simMat = LD, p = p, h = h, blMin = blMin,
##  $ dist.method: NULL
##  - attr(*, "class")= chr "hclust"
```

As described in Chapter 5, the parameter `h` controls the maximum lag between the columns of $\mathbf{X}$ considered. Thus, the LD measures are calculated between $\mathbf{X}[,i]$ and $\mathbf{X}[,j]$ only if i and j differ by no more than `h`. `sim=simR2` means that the LD similarity used is the r^2. Finally, if the parameter `heaps` is set to `TRUE`, then the implementation**+pencils +heap** detailed in Chapter 5 will be used. Otherwise, the implementation **-pencils -heap** presented in Section 5.2.1 will be run.

In addition to the tree structure produced by the clustering process, the `cWard` function returns the within-group dispersion measures W_k at each step of the clustering.

Besides, the `gapStatistic` function allows to apply the hierarchical clustering algorithm followed by the Gap statistic procedure:

```
gapS <- gapStatistic(X, min.nc=2, max.nc=p-1, B=50)
gapS$best.k

## [1] 6

## groups inferred using the first two steps of the proposed method
infGroups <- cutree(gapS$tree, gapS$best.k)
names(infGroups) <- NULL
infGroups

##  [1] 1 1 1 1 2 2 2 3 3 3 3 3 4 4 5 5 5 5 6 6
```

The `gapStatistic` function returns the tree structure resulting from the clustering of the genotype martix as well as the optimal number of clusters according to the Gap statistic procedure.

Finally, the Group Lasso regression can be applied to the inferred groups by running the `select` function as follows:

```
nlambda <- 30
reg <- select("groupLasso", X, y, groups=infGroups, nlambda=nlambda)
str(reg)

##  num [1:30, 1:20] 0 0 0 -0.0073 -0.00311 ...
##  - attr(*, "groups")= num [1:20] 1 1 1 1 2 2 2 3 3 3 ...
```

The returned object is a `nlambda` × `p` matrix of the Group Lasso coefficients as well as the group structure inferred by the first two steps of the proposed method as an attribute.

The second and easier option for applying the proposed three-step method consists in running the `grplassoCWard` function using these command lines:

```
B <- 50
## assessing the within-cluster measures for the B reference datasets
traceWB <- lapply(1:B, FUN=function(ii){
  getTraceW(n, p)
})
traceWB <- simplify2array(traceWB)
gl <- grplassoCWard(sim$X, sim$y, groups=NULL, nlambda=nlambda,
                    max.nc=p-1, min.nc=1, traceWB=traceWB, heaps=TRUE)
str(gl)

## List of 2
##  $ coefs : num [1:30, 1:20] 0 0 0 -0.0073 -0.00311 ...
##  $ groups: num [1:20] 1 1 1 1 2 2 2 3 3 3 ...
```

Through the default value `NULL` for the argument `groups`, the user indicates that the group structure needs to be inferred using the constrained Ward's incremental method and the Gap statistic model selection approach. The inferred group structure is returned as a vector of integers from `1` to `nBlocks`. SNPs sharing the same number belong to the same group. Therefore, we can check if the two first steps of inferring groups have well estimated the block sizes:

```
betas$blockSizes
```

```
## [1] 4 3 5 2 4 2
```

```
gl$groups
```

```
##  [1] 1 1 1 1 2 2 2 3 3 3 3 3 4 4 5 5 5 5 6 6
```

```
tab <- table(gl$groups)
dimnames(tab) <- NULL
tab
```

```
## [1] 4 3 5 2 4 2
```

The block sizes were in effect well estimated but this is not always the case above all when the correlation level is less that 0.4.
The second element of the output corresponds to the `nlambda` × `p` matrix of the Group Lasso coefficients.

E.4 Compared to other approaches

The `BALD` package allows the application of several regression methods using the function `select`:

```
coefsGL <- select("groupLasso", X, y, groups=NULL,  nlambda=nlambda,
                  max.nc=p-1, min.nc=2, B=100)
coefsOGL <- select("groupLasso", X, y, groups=betas$groups,
                   nlambda=nlambda)   ## Oracle Group Lasso !
coefsL <- select("lasso", X, y, nlambda=nlambda)
coefsEN <- select("elastic.net", X, y, lambda2=0.5, nlambda=nlambda)
pvalsUniv <- select("univ", X, y, nlambda=nlambda)
```

See vignette in the path :

```
system.file("evaluation/evaluation.Rnw", package="BALD")
```

for an example of performance comparison of different methods.

E.5 Representations of the results

The `BALD` package allows two different representations of the results: the first plot function `plotHeatmap` allows a Heatmap of the linkage disequilibrium blocks within a given region (Shin et al. 2006) and possibly to highlight selected blocks/SNPs. The second plot function `plotGroupsGL` provides a graphical display for interpreting selected blocks in function of the univariate p-values of the SNPs contained in these blocks.

Based on the regression results of the models Group Lasso and Lasso on the previously simulated data set, we can represent the first 3 blocks selected by the Group Lasso and the first SNPs selected by the Lasso as follows:

```
## "true" beta
betas$betaMat[,"betaSNP"]

##  [1]  0  0  0  0  1 -1  0 -1 -1  1  0  0  0  0  0  0  0  0  0  0

groups <- gl$groups
coefsGL <- select("groupLasso", X, y, groups=groups,  nlambda=nlambda)
selSNP <- as.matrix(t(coefsGL)!=0)

## blocks selected by GL at each level of regularization
selBl <- as.matrix(aggregate(selSNP, list(groups=groups), sum)[, -1])
str(selBl)

##  int [1:6, 1:30] 0 0 0 0 0 0 0 3 5 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:30] "V1" "V2" "V3" "V4" ...
```

```
selBl[,1:4]
```

```
##      V1 V2 V3 V4
## [1,]  0  0  0  4
## [2,]  0  3  3  3
## [3,]  0  5  5  5
## [4,]  0  0  0  0
## [5,]  0  0  0  4
## [6,]  0  0  0  0
```

```
## first 2 blocks selected by GL
firstBl <- which(selBl[,2]!=0)

## first 5 SNPs selected by the Lasso
coefsL <- select("lasso", X, y, nlambda=nlambda)
firstSNPs <- which(coefsL[4,]!=0)

## heatmap plot
blockSizes <- betas$blockSizes
```

```
plotHeatmap(X, blockSizes, selBlocks=firstBl, snpNames=as.character(1:p),
            snpStar=as.character(firstSNPs))
```

Figure E.1 displays the linkage disequilibrium measures of the set of 20 contiguous markers. The SNPs shown with a red star (*) correspond to the first markers (in the regularization path) selected by the Lasso. The local block structure inferred by the clustering and model selection steps of the proposed method is also highlighted and the first 3 blocks (in the regularization path) selected by the Group Lasso are delimited by a red outline.

Finally, in order to have a more accurate idea about the relevance of the blocks selected by the Group Lasso, we can display the univariate p-values of the SNPs within them. To do this, we will use the function `plotGroupsGL` that takes as arguments regression coefficients matrix of

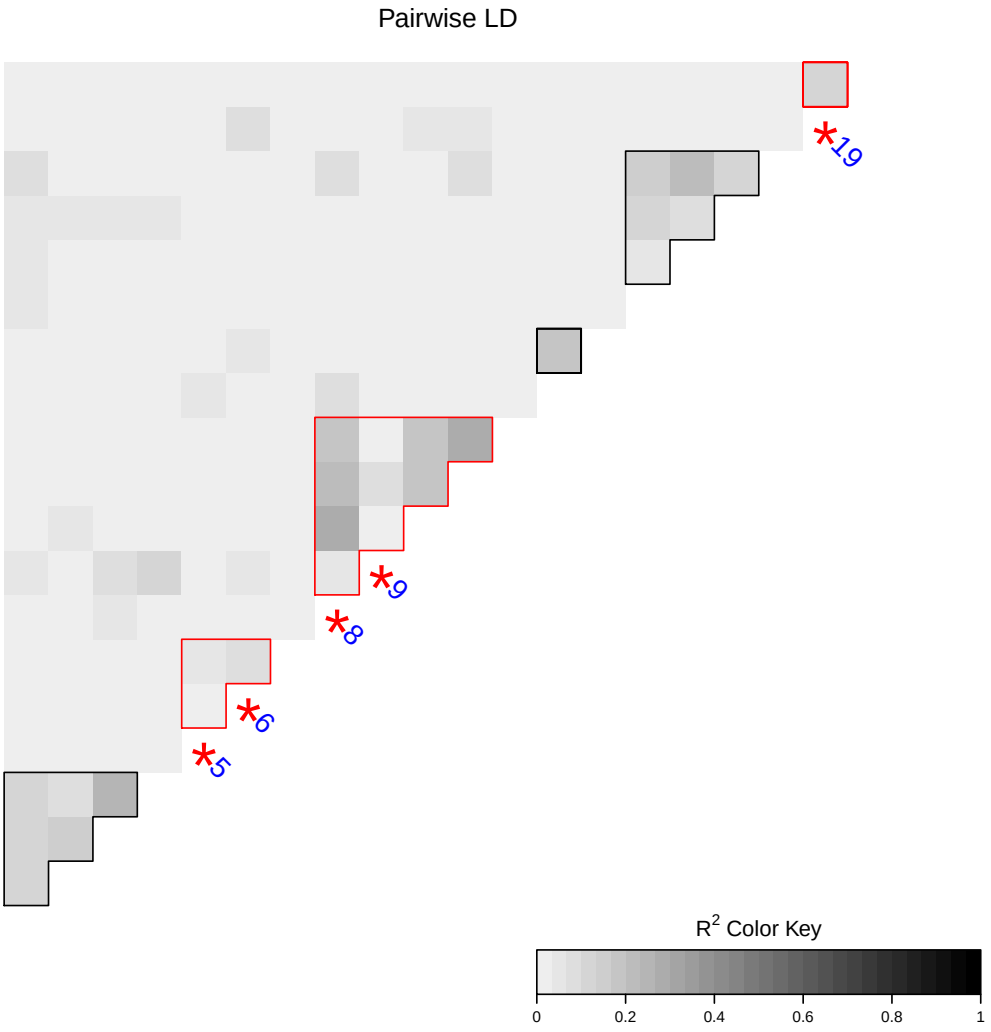
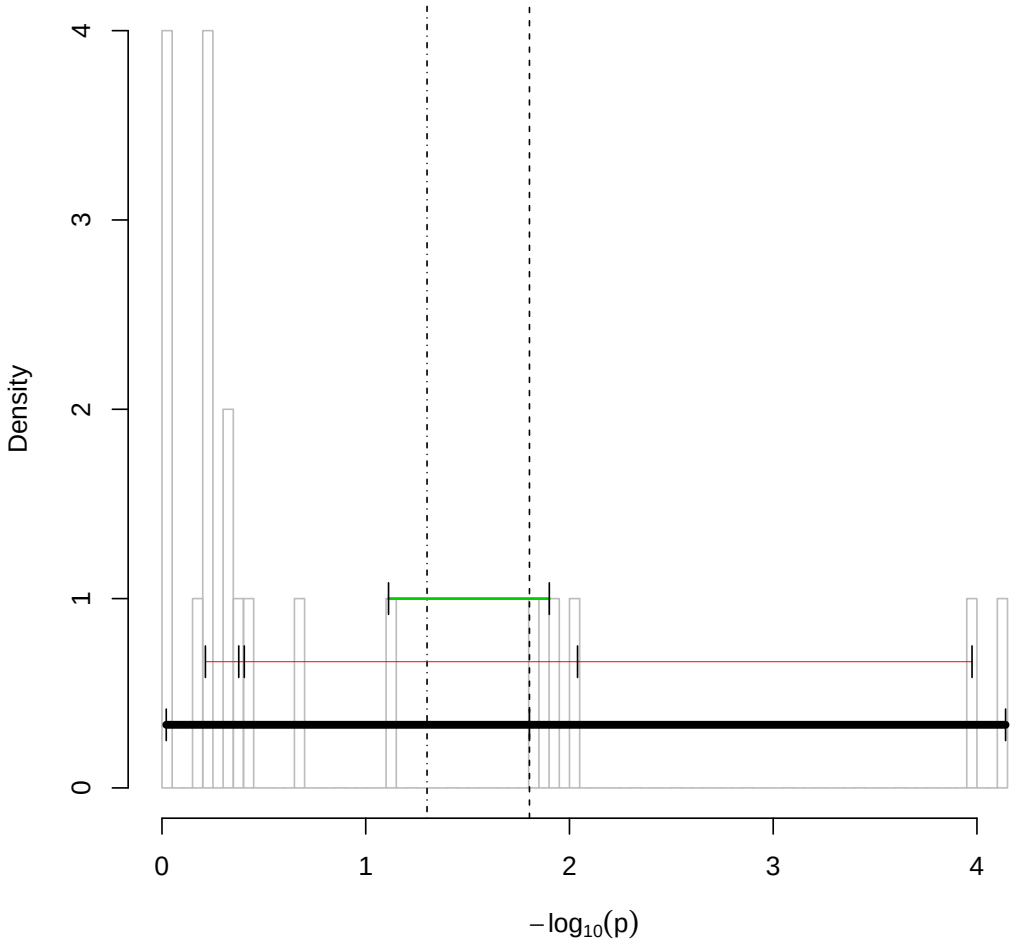


FIGURE E.1: Heatmap plot of the LD blocks.

the Group Lasso, the number of groups to be displayed and the univariate p-values of the 20 markers:

```
## univariate p-values
pvals <- apply(X, 2, FUN=function(vect){
  pv <- summary(lm(y ~ vect))$coefficients[2,4]
})

## first 3 blocks displayed
plotGroupsGL(coefsGL, nbGroup=3, pvals)
```



Each of the first 3 blocks selected by the Group Lasso is represented by a colored horizontal segment ranging from the largest to the smallest univariate p-value of the block. The vertical black segments indicate the univariate p-values of each SNP in these LD blocks and the vertical line highlights the significance threshold (defaults to `t=0.25`).

E.6 Session information

```
sessionInfo()


## R version 3.1.0 (2014-04-10)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
```

```
##
## locale:
## [1] fr_FR.UTF-8/fr_FR.UTF-8/fr_FR.UTF-8/C/fr_FR.UTF-8/fr_FR.UTF-8
##
## attached base packages:
## [1] methods   grid      stats     graphics  grDevices utils     datasets
## [8] base
##
## other attached packages:
## [1] BALD_0.2.1   knitr_1.10.5
##
## loaded via a namespace (and not attached):
##  [1] BiocGenerics_0.12.1 colorspace_1.2-6    digest_0.6.8
##  [4] evaluate_0.7        formatR_1.2         ggplot2_1.0.1
##  [7] grplasso_0.4-5      gtable_0.1.2        highr_0.5
## [10] lattice_0.20-31     LDheatmap_0.99-1    magrittr_1.5
## [13] MASS_7.3-40         Matrix_1.2-0        matrixStats_0.14.0
## [16] munsell_0.4.2       parallel_3.1.0      plyr_1.8.2
## [19] proto_0.3-10        quadrupen_0.2-4     Rcpp_0.11.6
## [22] reshape2_1.4.1      ROC_1.42.0          scales_0.2.4
## [25] snpStats_1.16.0     splines_3.1.0       stringi_0.4-1
## [28] stringr_1.0.0       survival_2.38-1     tools_3.1.0
## [31] zlibbioc_1.12.0
```

Contributions

Article

- Dehman Alia, Ambroise Christophe and Neuvial Pierre (2015), Performance of a block-wise approach in variable selection using linkage disequilibrium information, BMC bioinformatics 16(1), 148.

Conferences

- Dehman Alia, Rigaill Guillem, Neuvial Pierre and Ambroise Christophe. An efficient implementation of adjacency-constrained hierarchical clustering of a band similarity matrix, *International Federation of Classification Societies (IFCS)*. Bologna, 2015.
- Dehman Alia, Ambroise Christophe and Neuvial Pierre. BALD : Etude d'association par blocs de déséquilibre de liaison, *Troisièmes Rencontres R*. Toulouse, 2014.
- Dehman Alia, Ambroise Christophe and Neuvial Pierre. Incorporating linkage disequilibrium blocks in Genome-Wide Association Studies, *Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM)*. Toulouse, 2013.

Poster

- Dehman Alia, Ambroise Christophe and Neuvial Pierre. Incorporating linkage disequilibrium blocks in Genome-Wide Association Studies, *Software and Statistical Methods for Population Genetics (SSMPG)*. Aussois, 2013.

Seminars

- Dehman Alia, Rigaill Guillem, Neuvial Pierre and Ambroise Christophe. Incorporating linkage disequilibrium blocks in Genome-Wide Association Studies, *Séminaire AgroParisTech*. Paris, 2015.

- Dehman Alia, Rigaill Guillem, Neuvial Pierre and Ambroise Christophe. An efficient implementation of adjacency-constrained hierarchical clustering of a band similarity matrix, *Statistics for Systems Biology*. Jouy-en-Josas, 2015.

- Dehman Alia, Rigaill Guillem, Neuvial Pierre and Ambroise Christophe. Incorporating linkage disequilibrium blocks in Genome-Wide Association Studies, *Séminaire des doctorants*. Rennes, 2014.

Bibliography

Abecasis, G. R., Noguchi, E., Heinzmann, A., Traherne, J. A., Bhattacharyya, S., Leaves, N. I., Anderson, G. G., Zhang, Y., Lench, N. J., Carey, A. et al. (2001), 'Extent and distribution of linkage disequilibrium in three genomic regions', *The American Journal of Human Genetics* **68**(1), 191–197.

Abraham, G., Kowalczyk, A., Zobel, J. & Inouye, M. (2013), 'Performance and robustness of penalized and unpenalized methods for genetic prediction of complex human disease', *Genetic Epidemiology* **37**(2), 184–195.

Agapow, P.-M. & Burt, A. (2001), 'Indices of multilocus linkage disequilibrium', *Molecular Ecology Notes* **1**(1-2), 101–102.

Anderson, E. C. & Novembre, J. (2003), 'Finding haplotype block boundaries by using the minimum-description-length principle', *The American Journal of Human Genetics* **73**(2), 336–354.

Ardlie, K. G., Kruglyak, L. & Seielstad, M. (2002), 'Patterns of linkage disequilibrium in the human genome', *Nature Reviews Genetics* **3**(4), 299–309.

Arlot, S., Celisse, A. et al. (2010), 'A survey of cross-validation procedures for model selection', *Statistics surveys* **4**, 40–79.

Armitage, P. (1955), 'Tests for linear trends in proportions and frequencies', *Biometrics* **11**(3), 375–386.

Aulchenko, Y. S., Ripke, S., Isaacs, A. & Van Duijn, C. M. (2007), 'GenABEL: an R library for genome-wide association analysis', *Bioinformatics* **23**(10), 1294–1296.

Bach, F., Jenatton, R., Mairal, J., Obozinski, G. et al. (2012), 'Structured sparsity through convex optimization', *Statistical Science* **27**(4), 450–468.

Barrett, J. C., Fry, B., Maller, J. & Daly, M. J. (2005), 'Haploview: analysis and visualization of ld and haplotype maps', *Bioinformatics* **21**(2), 263–265.

Benjamini, Y. & Hochberg, Y. (1995), 'Controlling the false discovery rate: a practical and powerful approach to multiple testing', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 289–300.

Benjamini, Y. & Yekutieli, D. (2001), 'The control of the false discovery rate in multiple testing under dependency', *Annals of statistics* pp. 1165–1188.

Berkhin, P. (2004), 'Survey of clustering data mining techniques, 2002', *Accrue Software: San Jose, CA* .

Bonferroni, C. E. (1936), *Teoria statistica delle classi e calcolo delle probabilita*, Libreria internazionale Seeber.

Brass, P. (2008), *Advanced data structures*, Cambridge University Press Cambridge.

Bühlmann, P. (2013), 'Statistical significance in high-dimensional linear models', *Bernoulli* **19**, 1212–1242.

Burton, P. R., Clayton, D. G., Cardon, L. R., Craddock, N., Deloukas, P., Duncanson, A., Kwiatkowski, D. P., McCarthy, M. I., Ouwehand, W. H., Samani, N. J. et al. (2007), 'Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls', *Nature* **447**(7145), 661–678.

Caliński, T. & Harabasz, J. (1974), 'A dendrite method for cluster analysis', *Communications in Statistics-theory and Methods* **3**(1), 1–27.

Cardon, L. R. & Abecasis, G. R. (2003), 'Using haplotype blocks to map human complex trait loci', *TRENDS in Genetics* **19**(3), 135–140.

Carlson, C. S., Eberle, M. A., Rieder, M. J., Yi, Q., Kruglyak, L. & Nickerson, D. A. (2004), 'Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium', *The American Journal of Human Genetics* **74**(1), 106–120.

Charrad, M., Ghazzali, N., Boiteau, V. & Niknafs, A. (2014), 'NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set', *Journal of Statistical Software* **61**(6), 1–36.

Chatterjee, A. & Lahiri, S. N. (2011), 'Bootstrapping Lasso Estimators', *Journal of the American Statistical Association* **106**(494), 608–625.

Check Hayden, E. (2009), 'Genome sequencing: the third generation', *Nature News* .

Clayton, D. (2012), 'snpStats: SnpMatrix and XSnpMatrix classes and methods', *R package* .

Clayton, D. (2013), *snpStats: SnpMatrix and XSnpMatrix classes and methods*. R package version 1.12.0.

Clayton, D. & Leung, H.-T. (2007), 'An R package for analysis of whole-genome association studies', *Human heredity* **64**(1), 45–51.

Cochran, W. G. (1954), 'Some methods for strengthening the common χ 2 tests', *Biometrics* **10**(4), 417–451.

Consortium, . G. P. et al. (2010), 'A map of human genome variation from population-scale sequencing', *Nature* **467**(7319), 1061–1073.

Consortium, I. H. et al. (2005), 'A haplotype map of the human genome', *Nature* **437**(7063), 1299–1320.

Consortium, U. et al. (2015), 'The UK10K project identifies rare variants in health and disease', *Nature* **526**(7571), 82–90.

Cormen, T. H. (2009), *Introduction to algorithms*, MIT press.

Crews, K. R., Hicks, J. K., Pui, C.-H., Relling, M. V. & Evans, W. E. (2012), 'Pharmacogenomics and individualized medicine: translating science into practice', *Clinical Pharmacology & Therapeutics* **92**(4), 467–475.

Dalmasso, C., Carpentier, W., Meyer, L., Rouzioux, C., Goujard, C., Chaix, M.-L., Lambotte, O., Avettand-Fenoel, V., Le Clerc, S., de Senneville, L. D. et al. (2008), 'Distinct genetic loci control plasma HIV-RNA and cellular HIV-DNA levels in HIV-1 infection: the ANRS Genome Wide Association 01 study', *PloS one* **3**(12), e3907.

Daly, M. J., Rioux, J. D., Schaffner, S. F., Hudson, T. J. & Lander, E. S. (2001), 'High-resolution haplotype structure in the human genome', *Nature genetics* **29**(2), 229–232.

Dawson, E., Abecasis, G. R., Bumpstead, S., Chen, Y., Hunt, S., Beare, D. M., Pabial, J., Dibling, T., Tinsley, E., Kirby, S. et al. (2002), 'A first-generation linkage disequilibrium map of human chromosome 22', *Nature* **418**(6897), 544–548.

de Maturana, E., Ibáñez-Escriche, N., González-Recio, Ó. M., G. Mehrban, H. Chanock, S., Goddard, M. & Malats, N. (2014), 'Next generation modeling in GWAS: comparing different genetic architectures', *Human genetics* **133**(10), 1235–1253.

Dehman, A., Ambroise, C. & Neuvial, P. (2015), 'Performance of a blockwise approach in variable selection using linkage disequilibrium information', *BMC bioinformatics* **16**(1), 148.

Delaneau, O., Coulonges, C. & Zagury, J.-F. (2008), 'Shape-IT: new rapid and accurate algorithm for haplotype inference', *BMC bioinformatics* **9**(1), 540.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the royal statistical society. Series B (methodological)* pp. 1–38.

Devlin, B. & Risch, N. (1995), 'A comparison of linkage disequilibrium measures for fine-scale mapping', *Genomics* **29**(2), 311–322.

Diday, E. (1973), 'The dynamic clusters method in nonhierarchical clustering', *International Journal of Computer & Information Sciences* **2**(1), 61–88.

Ding, K. & Kullo, I. J. (2007), 'Methods for the selection of tagging snps: a comparison of tagging efficiency and performance', *European Journal of Human Genetics* **15**(2), 228–236.

Dudoit, S., Shaffer, J. P. & Boldrick, J. C. (2003), 'Multiple hypothesis testing in microarray experiments', *Statistical Science* pp. 71–103.

Dunning, A. M., Durocher, F., Healey, C. S., Teare, M. D., McBride, S. E., Carlomagno, F., Xu, C.-F., Dawson, E., Rhodes, S., Ueda, S. et al. (2000), 'The extent of linkage disequilibrium in four populations with distinct demographic histories', *The American Journal of Human Genetics* **67**(6), 1544–1554.

Durrant, C., Zondervan, K. T., Cardon, L. R., Hunt, S., Deloukas, P. & Morris, A. P. (2004), 'Linkage disequilibrium mapping via cladistic analysis of single-nucleotide polymorphism haplotypes', *The American Journal of Human Genetics* **75**(1), 35–43.

Excoffier, L. & Slatkin, M. (1995), 'Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population', *Molecular biology and evolution* **12**(5), 921–927.

Frazer, K. A., Ballinger, D. G., Cox, D. R., Hinds, D. A., Stuve, L. L., Gibbs, R. A., Belmont, J. W., Boudreau, A., Hardenbol, P., Leal, S. M. et al. (2007), 'A second generation human haplotype map of over 3.1 million SNPs', *Nature* **449**(7164), 851–861.

Gabriel, S. B., Schaffner, S. F., Nguyen, H., Moore, J. M., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M. et al. (2002), 'The structure of haplotype blocks in the human genome', *Science* **296**(5576), 2225–2229.

Gaunt, T. R., Rodríguez, S. & Day, I. N. (2007), 'Cubic exact solutions for the estimation of pairwise haplotype frequencies: implications for linkage disequilibrium analyses and a web tool'CubeX'', *BMC bioinformatics* **8**(1), 428.

González, J. R., Armengol, L., Solé, X., Guinó, E., Mercader, J. M., Estivill, X. & Moreno, V. (2007), 'SNPassoc: an R package to perform whole genome association studies', *Bioinformatics* **23**(5), 654–655.

Gordon, A. (1999), 'Classification', *Monographs on Statistics and Applied Probability* **82**.

Graham, R. L. & Hell, P. (1985), 'On the history of the minimum spanning tree problem', *Annals of the History of Computing* **7**(1), 43–57.

Gramfort, A. & Kowalski, M. (2009), Improving m/eeg source localizationwith an inter-condition sparse prior, *in* 'Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on', IEEE, pp. 141–144.

Greenspan, G. & Geiger, D. (2004), 'Model-based inference of haplotype block variation', *Journal of computational biology* **11**(2-3), 493–504.

Guo, S. W. & Thompson, E. A. (1992), 'Performing the exact test of hardy-weinberg proportion for multiple alleles', *Biometrics* pp. 361–372.

Halldorsson, B. V., Bafna, V., Lippert, R., Schwartz, R., Francisco, M., Clark, A. G. & Istrail, S. (2004), 'Optimal haplotype block-free selection of tagging snps for genome-wide association studies', *Genome research* **14**(8), 1633–1640.

Hardy, G. H. (1908), 'Mendelian proportions in a mixed population', *Science* pp. 49–50.

Hartigan, J. (1975), 'Clustering algorithms (probability & mathematical statistics)'.

Hebiri, M. (2008), 'Regularization with the smooth-lasso procedure', *arXiv preprint arXiv:0803.0668* .

Hill, W. & Robertson, A. (1968), 'Linkage disequilibrium in finite populations', *Theoretical and Applied Genetics* **38**(6), 226–231.

Hochberg, Y. & Tamhane, A. C. (2009), *Multiple comparison procedures*, Wiley.

Huang, J., Howie, B., McCarthy, S., Memari, Y., Walter, K., Min, J. L., Danecek, P., Malerba, G., Trabetti, E., Zheng, H.-F. et al. (2015), 'Improved imputation of low-frequency and rare variants using the UK10K haplotype reference panel', *Nature communications* **6**.

Huang, J., Zhang, T. & Metaxas, D. (2011), 'Learning with structured sparsity', *The Journal of Machine Learning Research* **12**, 3371–3412.

Hunter, D. J. (2005), 'Gene–environment interactions in human diseases', *Nature Reviews Genetics* **6**(4), 287–298.

Jacob, L., Obozinski, G. & Vert, J.-P. (2009), Group lasso with overlap and graph lasso, *in* 'Proceedings of the 26th annual international conference on machine learning', ACM, pp. 433–440.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), 'Data clustering: a review', *ACM computing surveys (CSUR)* **31**(3), 264–323.

Jeffreys, A. J., Kauppi, L. & Neumann, R. (2001), 'Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex', *Nature genetics* **29**(2), 217–222.

Jenatton, R., Audibert, J.-Y. & Bach, F. (2011), 'Structured variable selection with sparsity-inducing norms', *The Journal of Machine Learning Research* **12**, 2777–2824.

Jenatton, R., Obozinski, G. & Bach, F. (2009), 'Structured sparse principal component analysis', *arXiv preprint arXiv:0909.1440* .

Johnson, G. C., Esposito, L., Barratt, B. J., Smith, A. N., Heward, J., Di Genova, G., Ueda, H., Cordell, H. J., Eaves, I. A., Dudbridge, F. et al. (2001), 'Haplotype tagging for the identification of common disease genes', *Nature genetics* **29**(2), 233–237.

Johnson, R. & Wichern, D. (2002), 'Applied Multivariate Analysis'.

Jorde, L. (2000), 'Linkage disequilibrium and the search for complex disease genes', *Genome research* **10**(10), 1435–1444.

Koivisto, M., Perola, M., Varilo, T., Hennah, W., Ekelund, J., Lukk, M., Peltonen, L., Ukkonen, E. & Mannila, H. (2003), An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries, *in* 'Pacific Symposium on Biocomputing', Vol. 8, World Scientific, pp. 502–513.

Kruglyak, L. (1999), 'Prospects for whole-genome linkage disequilibrium mapping of common disease genes', *Nature genetics* **22**(2), 139–144.

Kruglyak, L. & Nickerson, D. A. (2001), 'Variation is the spice of life', *Nature genetics* **27**(3), 234–235.

Krzanowski, W. J. & Lai, Y. (1988), 'A criterion for determining the number of groups in a data set using sum-of-squares clustering', *Biometrics* pp. 23–34.

Kwee, L., Liu, D., Lin, X., Ghosh, D. & Epstein, M. (2008), 'A powerful and flexible multilocus association test for quantitative traits', *The American Journal of Human Genetics* **82**(2), 386–397.

Lance, G. N. & Williams, W. T. (1967), 'A general theory of classificatory sorting strategies 1. hierarchical systems', *The computer journal* **9**(4), 373–380.

Land, S. & Friedman, J. (1996), 'Variable fusion: a new method of adaptive signal regression', *Manuscript* .

Lander, E. S. (2011), 'Initial impact of the sequencing of the human genome', *Nature* **470**(7333), 187–197.

Last, J. M., Association, I. E. et al. (2001), *A dictionary of epidemiology*, Vol. 4, Oxford Univ Press.

Lévy-Leduc, C., Delattre, M., Mary-Huard, T. & Robin, S. (2014), 'Two-dimensional segmentation for analyzing Hi-C data', *Bioinformatics* **30**(17), i386–i392.

Lewontin, R. (1964), 'The interaction of selection and linkage. I. General considerations; heterotic models', *Genetics* **49**(1), 49.

Lewontin, R. (1988), 'On measures of gametic disequilibrium.', *Genetics* **120**(3), 849–852.

Li, M.-X., Gui, H.-S., Kwan, J. S. & Sham, P. C. (2011), 'GATES: a rapid and powerful gene-based association test using extended Simes procedure', *The American Journal of Human Genetics* **88**(3), 283–293.

Lieberman-Aiden, E., van Berkum, N. L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., Amit, I., Lajoie, B. R., Sabo, P. J., Dorschner, M. O. et al. (2009), 'Comprehensive mapping of long-range interactions reveals folding principles of the human genome', *science* **326**(5950), 289–293.

Liu, J., Huang, J., Ma, S. & Wang, K. (2013), 'Incorporating group correlations in genome-wide association studies using smoothed group Lasso', *Biostatistics* **14**(2), 205–219.

Lohninger, H. (2010), 'Fundamentals of statistics', *Retrieved December* **5**, 2010.

Lou, X.-Y., Casella, G., Littell, R. C., Yang, M. C., Johnson, J. A. & Wu, R. (2003), 'A haplotype-based algorithm for multilocus linkage disequilibrium mapping of quantitative trait loci with epistasis', *Genetics* **163**(4), 1533–1548.

MacQueen, J. et al. (1967), Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of the fifth Berkeley symposium on mathematical statistics and probability', Vol. 1, Oakland, CA, USA., pp. 281–297.

Mandozzi, J. & Bühlmann, P. (2015), 'A sequential rejection testing method for high-dimensional regression with correlated variables', *arXiv preprint arXiv:1502.03300* .

Mannila, H., Koivisto, M., Perola, M., Varilo, T., Hennah, W., Ekelund, J., Lukk, M., Peltonen, L. & Ukkonen, E. (2003), 'Minimum description length block finder, a method to identify haplotype blocks and to compare the strength of block boundaries', *The American Journal of Human Genetics* **73**(1), 86–94.

Manolio, T. A., Collins, F. S., Cox, N. J., Goldstein, D. B., Hindorff, L. A., Hunter, D. J., McCarthy, M. I., Ramos, E. M., Cardon, L. R., Chakravarti, A. et al. (2009), 'Finding the missing heritability of complex diseases', *Nature* **461**(7265), 747–753.

McCullagh, P., Nelder, J. A. & McCullagh, P. (1989), *Generalized linear models*, Vol. 2, Chapman and Hall London.

Meinshausen, N. (2008), 'Hierarchical testing of variable importance', *Biometrika* **95**(2), 265–278.

Milligan, G. W. & Cooper, M. C. (1985), 'An examination of procedures for determining the number of clusters in a data set', *Psychometrika* **50**(2), 159–179.

Mohajer, M., Englmeier, K.-H. & Schmid, V. J. (2011), 'A comparison of Gap statistic definitions with and without logarithm function', *arXiv preprint arXiv:1103.4767* .

Molitor, J., Marjoram, P. & Thomas, D. (2003), 'Fine-scale mapping of disease genes with multiple mutations via spatial clustering techniques', *The American Journal of Human Genetics* **73**(6), 1368–1384.

Morris, A. P. & Zeggini, E. (2010), 'An evaluation of statistical approaches to rare variant analysis in genetic association studies', *Genetic epidemiology* **34**(2), 188.

Morton, N. E. (1982), *Outline of genetic epidemiology*, Karger Basel.

Mourad, R., Sinoquet, C. & Leray, P. (2011), 'A hierarchical bayesian network approach for linkage disequilibrium modeling and data-dimensionality reduction prior to genome-wide association studies', *BMC bioinformatics* **12**(1), 16.

Niu, T., Qin, Z. S., Xu, X. & Liu, J. S. (2002), 'Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms', *The American Journal of Human Genetics* **70**(1), 157–169.

Patil, N., Berno, A. J., Hinds, D. A., Barrett, W. A., Doshi, J. M., Hacker, C. R., Kautzer, C. R., Lee, D. H., Marjoribanks, C., McDonough, D. P. et al. (2001), 'Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21', *Science* **294**(5547), 1719–1723.

Pattaro, C., Ruczinski, I., Fallin, D. M. & Parmigiani, G. (2008), 'Haplotype block partitioning as a tool for dimensionality reduction in SNP association studies', *BMC genomics* **9**(1), 405.

Phillips, M., Lawrence, R., Sachidanandam, R., Morris, A., Balding, D., Donaldson, M., Studebaker, J., Ankener, W., Alfisi, S., Kuo, F.-S. et al. (2003), 'Chromosome-wide distribution of haplotype blocks and the role of recombination hot spots', *nature genetics* **33**(3), 382–387.

Pritchard, J. K. & Przeworski, M. (2001), 'Linkage disequilibrium in humans: models and data', *The American Journal of Human Genetics* **69**(1), 1–14.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J. et al. (2007), 'PLINK: a tool set for whole-genome association and population-based linkage analyses', *The American Journal of Human Genetics* **81**(3), 559–575.

Qin, Z. S., Niu, T. & Liu, J. S. (2002), 'Partition-ligation–expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms', *American journal of human genetics* **71**(5), 1242.

Rakyan, V. K., Down, T. A., Balding, D. J. & Beck, S. (2011), 'Epigenome-wide association studies for common human diseases', *Nature Reviews Genetics* **12**(8), 529–541.

Rao, M. (1971), 'Cluster analysis and mathematical programming', *Journal of the American statistical association* **66**(335), 622–626.

Rapaport, F., Barillot, E. & Vert, J.-P. (2008), 'Classification of arrayCGH data using fused SVM', *Bioinformatics* **24**(13), i375–i382.

Reich, D. E., Cargill, M., Bolk, S., Ireland, J., Sabeti, P. C., Richter, D. J., Lavery, T., Kouyoumjian, R., Farhadian, S. F., Ward, R. et al. (2001), 'Linkage disequilibrium in the human genome', *Nature* **411**(6834), 199–204.

Rinaldo, A., Bacanu, S.-A., Devlin, B., Sonpar, V., Wasserman, L. & Roeder, K. (2005), 'Characterization of multilocus linkage disequilibrium', *Genetic epidemiology* **28**(3), 193–206.

Rinaldo, A. et al. (2009), 'Properties and refinements of the fused lasso', *The Annals of Statistics* **37**(5B), 2922–2952.

Risch, N. J. (2000), 'Searching for genetic determinants in the new millennium', *Nature* **405**(6788), 847–856.

Rousseau, F. & Laflamme, N. (2003), 'Génétique moléculaire humaine: des maladies monogéniques aux maladies complexes', *M/S: médecine sciences* **19**(10), 950–954.

Sasieni, P. D. (1997), 'From genotypes to genes: doubling the sample size', *Biometrics* pp. 1253–1261.

Schaid, D. J., Rowland, C. M., Tines, D. E., Jacobson, R. M. & Poland, G. A. (2002), 'Score tests for association between traits and haplotypes when linkage phase is ambiguous', *The American Journal of Human Genetics* **70**(2), 425–434.

Schölkopf, B. & Smola, A. J. (2002), *Learning with kernels: Support vector machines, regularization, optimization, and beyond*, MIT press.

Sedgewick, R. (1988), 'Algorithms, 1988'.

Seltman, H., Roeder, K. & Devlin, B. (2003), 'Evolutionary-based association analysis using haplotype data', *Genetic epidemiology* **25**(1), 48–58.

Shaffer, J. P. (1995), 'Multiple hypothesis testing', *Annual review of psychology* **46**(1), 561–584.

Sham, P. C. & Purcell, S. M. (2014), 'Statistical power and significance testing in large-scale genetic studies', *Nature Reviews Genetics* **15**(5), 335–346.

Shifman, S., Kuypers, J., Kokoris, M., Yakir, B. & Darvasi, A. (2003), 'Linkage disequilibrium patterns of the human genome across populations', *Human molecular genetics* **12**(7), 771–776.

Shin, J.-H., Blay, S., McNeney, B. & Graham, J. (2006), 'LDheatmap: an R function for graphical display of pairwise linkage disequilibria between single nucleotide polymorphisms', *Journal of Statistical Software* **16**(3), 1–10.

Šidák, Z. (1967), 'Rectangular confidence regions for the means of multivariate normal distributions', *Journal of the American Statistical Association* **62**(318), 626–633.

Simon, N., Friedman, J., Hastie, T. & Tibshirani, R. (2013), 'A sparse-group lasso', *Journal of Computational and Graphical Statistics* **22**(2), 231–245.

Skiena, S. S. (1998), *The algorithm design manual: Text*, Vol. 1, Springer Science & Business Media.

Sokal, R. R., Sneath, P. H. et al. (1963), 'Principles of numerical taxonomy', *Principles of numerical taxonomy.* .

Stephens, M. & Donnelly, P. (2003), 'A comparison of bayesian methods for haplotype reconstruction from population genotype data', *The American Journal of Human Genetics* **73**(5), 1162–1169.

Stojnic, M., Parvaresh, F. & Hassibi, B. (2009), 'On the reconstruction of block-sparse signals with an optimal number of measurements', *Signal Processing, IEEE Transactions on* **57**(8), 3075–3085.

Sugar, C. A. (1998), Techniques for clustering and classification with applications to medical problems, PhD thesis, Stanford University.

Sugar, C. A., Lenert, L. A. & Olshen, R. A. (1999), 'An application of cluster analysis to health services research: Empirically de ned health states for depression from the SF-12', *Citeseer* .

Tabor, H. K., Risch, N. J. & Myers, R. M. (2002), 'Candidate-gene approaches for studying complex genetic traits: practical considerations', *Nature Reviews Genetics* **3**(5), 391–397.

Taillon-Miller, P., Bauer-Sardiña, I., Saccone, N. L., Putzel, J., Laitinen, T., Cao, A., Kere, J., Pilia, G., Rice, J. P. & Kwok, P.-Y. (2000), 'Juxtaposed regions of extensive and minimal linkage disequilibrium in human Xq25 and Xq28', *Nature genetics* **25**(3), 324–328.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005), 'Sparsity and smoothness via the fused lasso', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(1), 91–108.

Tibshirani, R., Walther, G. & Hastie, T. (2001), 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(2), 411–423.

Tomita, M., Hatsumichi, M. & Kurihara, K. (2008), 'Identify LD blocks based on hierarchical spatial data', *Computational Statistics & Data Analysis* **52**(4), 1806–1820.

Twells, R. C., Mein, C. A., Phillips, M. S., Hess, J. F., Veijola, R., Gilbey, M., Bright, M., Metzker, M., Lie, B. A., Kingsnorth, A. et al. (2003), 'Haplotype structure, LD blocks, and uneven recombination within the LRP5 gene', *Genome research* **13**(5), 845–855.

Tzeng, J.-Y. (2005), 'Evolutionary-based grouping of haplotypes in association analysis', *Genetic epidemiology* **28**(3), 220–231.

Tzeng, J.-Y., Wang, C.-H., Kao, J.-T. & Hsiao, C. K. (2006), 'Regression-based association analysis with clustered haplotypes through use of genotypes', *The American Journal of Human Genetics* **78**(2), 231–242.

Waldmann, P., Mészáros, G., Gredler, B., Fuerst, C. & Sölkner, J. (2013), 'Evaluation of the lasso and the elastic net in genome-wide association studies', *Frontiers in genetics* **4**.

Wall, J. D. & Pritchard, J. K. (2003), 'Haplotype blocks and linkage disequilibrium in the human genome', *Nature Reviews Genetics* **4**(8), 587–597.

Wang, N., Akey, J. M., Zhang, K., Chakraborty, R. & Jin, L. (2002), 'Distribution of recombination crossovers and the origin of haplotype blocks: the interplay of population history, recombination, and mutation', *The American Journal of Human Genetics* **71**(5), 1227–1234.

Ward Jr, J. H. (1963), 'Hierarchical grouping to optimize an objective function', *Journal of the American statistical association* **58**(301), 236–244.

Weale, M. E., Depondt, C., Macdonald, S. J., Smith, A., San Lai, P., Shorvon, S. D., Wood, N. W. & Goldstein, D. B. (2003), 'Selection and evaluation of tagging SNPs in the neuronal-sodium-channel gene SCN1A: implications for linkage-disequilibrium gene mapping', *The American Journal of Human Genetics* **73**(3), 551–565.

Weinberg, W. (1908), 'Über vererbungsgesetze beim menschen', *Molecular and General Genetics MGG* **1**(1), 440–460.

Weir, B. S. & Cockerham, C. (1996), 'Genetic data analysis II: Methods for discrete population genetic data. Sinauer Assoc', *Inc., Sunderland, MA, USA* .

Welter, D., MacArthur, J., Morales, J., Burdett, T., Hall, P., Junkins, H., Klemm, A., Flicek, P., Manolio, T., Hindorff, L. et al. (2014), 'The NHGRI GWAS Catalog, a curated resource of SNP-trait associations', *Nucleic acids research* **42**(D1), D1001–D1006.

Williams, J. W. J. (1964), 'Algorithm-232-heapsort'.

Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E. & Lange, K. (2009), 'Genome-wide association analysis by lasso penalized logistic regression', *Bioinformatics* **25**(6), 714–721.

Yi, H., Breheny, P., Imam, N., Liu, Y. & Hoeschele, I. (2015), 'Penalized Multimarker vs. Single-Marker Regression Methods for Genome-Wide Association Studies of Quantitative Traits', *Genetics* **199**(1), 205–222.

Yuan, A., Chen, G., Zhou, Y., Bentley, A. & Rotimi, C. (2012), 'A novel approach for the simultaneous analysis of common and rare variants in complex traits', *Bioinformatics and biology insights* **6**, 1.

Yuan, M. & Lin, Y. (2005), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67.

Zhang, K., Deng, M., Chen, T., Waterman, M. S. & Sun, F. (2002), 'A dynamic programming algorithm for haplotype block partitioning', *Proceedings of the National Academy of Sciences* **99**(11), 7335–7339.

Zhang, K., Sun, F., Waterman, M. S. & Chen, T. (2003), 'Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data', *The American Journal of Human Genetics* **73**(1), 63–73.

Zhao, P., Rocha, G. & Yu, B. (2006), 'Grouped and hierarchical model selection through composite absolute penalties', *Department of Statistics, UC Berkeley, Tech. Rep* **703**.

Zhu, X., Yan, D., Cooper, R. S., Luke, A., Ikeda, M. A., Chang, Y.-P. C., Weder, A. & Chakravarti, A. (2003), 'Linkage disequilibrium and haplotype diversity in the genes of the renin–angiotensin system: findings from the family blood pressure program', *Genome research* **13**(2), 173–181.

Zou, H. (2006), 'The adaptive lasso and its oracle properties', *Journal of the American statistical association* **101**(476), 1418–1429.

Zou, H. & Hastie, T. (2005), 'regularization and variable selection via the elastic net', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(2), 301–320.

Zuk, O., Hechter, E., Sunyaev, S. R. & Lander, E. S. (2012), 'The mystery of missing heritability: Genetic interactions create phantom heritability', *Proceedings of the National Academy of Sciences* **109**(4), 1193–1198.

Abstract

With recent development of high-throughput genotyping technologies, the usage of Genome-Wide Association Studies (GWAS) has become widespread in genetic research. By screening large portions of the genome, these studies aim to characterize genetic factors involved in the development of complex genetic diseases.

GWAS are also based on the existence of statistical dependencies, called Linkage Disequilibrium (LD) usually observed between nearby loci on DNA. LD is defined as the non-random association of alleles at different loci on the same chromosome or on different chromosomes in a population. This biological feature is of fundamental importance in association studies as it provides a fine location of unobserved causal mutations using adjacent genetic markers. Nevertheless, the complex block structure induced by LD as well as the large volume of genetic data are key issues that have arisen with GWA studies.

The contributions presented in this manuscript are in twofold, both methodological and algorithmic. On the methodological part, we propose a three-step approach that explicitly takes advantage of the grouping structure induced by LD in order to identify common variants which may have been missed by single marker analyses. In the first step, we perform a hierarchical clustering of SNPs with an adjacency constraint using LD as a similarity measure. In the second step, we apply a model selection approach to the obtained hierarchy in order to define LD blocks. Finally, we perform Group Lasso regression on the inferred LD blocks. The efficiency of the proposed approach is investigated compared to state-of-the art regression methods on simulated, semi-simulated and real GWAS data.

On the algorithmic part, we focus on the spatially-constrained hierarchical clustering algorithm whose quadratic time complexity is not adapted to the high-dimensionality of GWAS data. We then present, in this manuscript, an efficient implementation of such an algorithm in the general context of any similarity measure. By introducing a user-parameter h and using the min-heap structure, we obtain a sub-quadratic time complexity of the adjacency-constrained hierarchical clustering algorithm, as well as a linear space complexity in the number of items to be clustered. The interest of this novel algorithm is illustrated in GWAS applications.

keywords: Genome-Wide Association Studies, Linkage Disequilibrium, Hierarchical clustering, Model selection, Gap statistic, Penalized regression, Group lasso

Résumé

Avec le développement récent des technologies de génotypage à haut débit, l'utilisation des études d'association pangénomiques (GWAS) est devenue très répandue dans la recherche génétique. Au moyen de criblage de grandes parties du génome, ces études visent à caractériser les facteurs génétiques impliqués dans le développement de maladies génétiques complexes.

Les GWAS sont également basées sur l'existence de dépendances statistiques, appelées déséquilibre de liaison (DL), habituellement observées entre des loci qui sont proches dans l'ADN. Le DL est défini comme l'association non aléatoire d'allèles à des loci différents sur le même chromosome ou sur des chromosomes différents dans une population. Cette caractéristique biologique est d'une importance fondamentale dans les études d'association car elle permet la localisation précise des mutations causales en utilisant les marqueurs génétiques adjacents. Néanmoins, la structure de blocs complexe induite par le DL ainsi que le grand volume de données génétiques constituent les principaux enjeux soulevés par les études GWAS.

Les contributions présentées dans ce manuscrit comportent un double aspect, à la fois méthodologique et algorithmique. Sur le plan méthodologie, nous proposons une approche en trois étapes qui tire profit de la structure de groupes induite par le DL afin d'identifier des variants communs qui pourraient avoir été manquées par l'analyse simple marqueur. Dans une première étape, nous effectuons une classification hiérarchique des SNPs avec une contrainte d'adjacence et en utilisant le DL comme mesure de similarité. Dans une seconde étape, nous appliquons une approche de sélection de modèle à la hiérarchie obtenue afin de définir des blocs de DL. Enfin, nous appliquons le modèle de régression Group Lasso sur les blocs de DL inférés. L'efficacité de l'approche proposée est comparée à celle des approches de régression standards sur des données simulées, semi-simulées et réelles de GWAS.

Sur le plan algorithmique, nous nous concentrons sur l'algorithme de classification hiérarchique avec contrainte spatiale dont la complexité quadratique en temps n'est pas adaptée à la grande dimension des données GWAS. Ainsi, nous présentons, dans ce manuscrit, une mise en œuvre efficace d'un tel algorithme dans le contexte général de n'importe quelle mesure de similarité. En introduisant un paramètre h défini par l'utilisateur et en utilisant la structure de tas-min, nous obtenons une complexité sous-quadratique en temps de l'algorithme de classification hiérarchie avec contrainte d'adjacence, ainsi qu'une complexité linéaire en mémoire en le nombre d'éléments à classer. L'intérêt de ce nouvel algorithme est illustré dans des applications GWAS.