Thesis for the Degree of Doctor

Proposal, evaluation and application of two new methodologies for genetic profiling in association studies.

Víctor Urrea Gales

Supervisor:

Dr. M.Luz Calle Rosingana

UVic - UCC

Vic, March 2015

Ph.D. in Systems Biology

Systems Biology Department

Universitat de Vic - Universitat Central de Catalunya

This tesis has been supported by a pre-doctoral FPU fellowship
from the Spanish Ministry of Education, Culture and Sport
(ref. AP2009-3044)

To Aran, Martí and Marta.

Acknowledgements

This thesis would never have gotten off the ground without the effort and drive of my supervisor, Dr. Malu Calle. To her I owe a dedication far beyond what would be expected.

Thanks to the members of the BEM group and the Systems Biology department at UVic for treating me so well all these years and for your consideration and support.

Thanks to the GRASS team for letting me join the group and let me benefit from their knowledge and experience.

Thanks to all who have allowed me to somehow work with them, especially to Dr. Núria Malalts.

Abstract

The aim of genetic epidemiology is to understand the effects of genes and environmental factors on human health. In recent years there have been many studies of association with the goal of studying the role of the genetic component in the risk for common diseases such as diabetes, cardiovascular disease, cancer, Alzheimer's, etc. These studies are usually based on a selection of genetic markers which are genotyped for samples of diseased (cases) and healthy (controls) individuals in order to study the level of association with disease.

This thesis is focused on "Genomic Profiling" which refers to the identification of genetic variants at multiple loci for prediction of disease risk. This involves variable selection and model building. A common strategy for genomic profiling consists in building a disease prediction model that only contains those genetic markers with a marginal significant effect on the disease. This marginal approach has an important limitation, it ignores epistasis. From a statistical point of view, an epistatic effect or a genetic interaction is present when the combined effect of several genetic markers on the observed phenotype is not explained by their marginal effects. The main difficulty in the study of genetic interactions is that usually hundreds or thousands of genetic markers are analyzed, which makes the analysis of all possible interactions unfeasible, from a computational point of view. As a consequence of this, most available methods for epistasis analysis only allow the study of low order interactions,

second or third order interactions.

In this thesis we present two new methodological approaches to address genomic profiling in presence of epistasis which allow higher order interactions to be explored with a reasonable computational cost.

The first method, called AUC-RF, is based on a classification methodology widely used in the context of automated learning techniques called Random Forest (RF). The AUC-RF algorithm implements a backward variable selection based on optimizing the ROC curves of the RF. This process provides the set of genetic markers with the highest predictive value of the disease risk.

The RF provides a ranking of the predictor variables based on some measures of importance implemented in the method. The two most commonly used importance measures are the mean decrease accuracy (MDA) and the mean decrease Gini (MDG). In this thesis we present a study in which we explored the stability and robustness of these two measures of importance. The results of this study show that MDA is very unstable and MDG provides more appropriate rankings for the predictor variables that are actually associated with the response variable. This result justifies the use of the MDG importance measure in the AUC-RF methodology.

The second proposed method, called Optimal AUC, is based on the concept of optimal ROC curves. The objective is again obtaining sets of genetic variables with the highest predictive capability, now through the optimum combination of variables based on likelihood ratio measures. In this case the proposed algorithm performs a forward selection of variables that can identify genetic interactions of higher order, i.e. a large number of variables, and is computationally feasible.

Another prominent issue addressed in this thesis is the simulation of data with similar properties to real genetic association studies. Simulation studies are very important in the development and evaluation of new methods. Power and effectiveness

can be gauged using datasets from which known results are obtained. Comparison of results obtained using different techniques is possible as well as an evaluation of the effectiveness of the methodology by varying simulation parameters. With this purpose we developed a set of R functions to simulate genotyping data and associated response variable based on risk models.

Resumen

El objetivo de la epidemiología genética es comprender los efectos de los genes y los factores ambientales en la salud humana. En los últimos años se han realizado muchos estudios de asociación con el propósito de estudiar el papel del componente genético en el riesgo de enfermedades comunes como la diabetes, enfermedades cardiovasculares, cáncer, Alzheimer, etc. Estos estudios se basan por lo general en una selección de marcadores genéticos que se genotipan para muestras de individuos enfermos (casos) y sanos (controles) con el fin de estudiar el nivel de asociación con la enfermedad.

Esta tesis se centra en lo que se denomina "Genomic Profiling", que se refiere a la identificación de múltiples variantes genéticas para predecir el riesgo a la enfermedad. Esto implica una selección de variables y la construcción de un modelo. Una estrategia común consiste en la construcción de un modelo de predicción que contenga únicamente los marcadores genéticos con un efecto marginal significativo sobre la enfermedad. Este enfoque marginal tiene una limitación importante, ignora la epistasis. Desde un punto de vista estadístico, un efecto epistático o una interacción genética aparece cuando el efecto conjunto de varios marcadores genéticos sobre el fenotipo observado no queda explicado por sus efectos marginales. La principal dificultad en el estudio de las interacciones genéticas es que generalmente se analizan cientos o miles de marcadores genéticos, lo que hace inviable el análisis de todas las

posibles interacciones, desde el punto de vista computacional. Como consecuencia de ello, la mayoría de los métodos disponibles para el análisis de epistasis sólo permiten el estudio de las interacciones de orden inferior, interacciones de segundo o tercer orden.

En esta tesis presentamos dos aproximaciones metodológicas nuevas para la identificación de perfiles genéticos en presencia de epistasis que permiten explorar interacciones de alto orden con un coste computacional razonable.

La primera de las metodologías presentadas, denominada AUC-RF, se basa en un método de clasificación ampliamente utilizado en el contexto de las técnicas de aprendizaje automatizado denominado Random Forest (RF). El algoritmo AUC-RF implementa una selección de variables hacia atrás basada en la optimización de las curvas ROC del RF que permite identificar conjuntos de marcadores genéticos con el mayor valor predictivo sobre el riesgo a la enfermedad.

Un RF proporciona una ordenación de las variables predictoras en base a algunas de las medidas de importancia que el método tiene implementado. Las dos medidas más utilizadas son el mean decrease accuracy (MDA) y el mean decrease Gini (MDG). En esta tesis presentamos un trabajo en el que hemos explorado la estabilidad y la robustez de estas dos medidas de importancia. Los resultados de este estudio ponen de manifiesto que MDA es muy inestable y que MDG proporciona unos rankings más adecuados para las variables predictoras que realmente están asociadas con la variable respuesta. Este resultado justifica que en la metodología AUC-RF utilizamos la medida de importancia MDG.

La segunda metodología propuesta, denominada Optimal AUC, se basa en el concepto de curvas ROC óptimas. El objetivo vuelve a ser la obtención de conjuntos de variables genéticas con la máxima capacidad predictiva, en este caso basándonos en la combinación óptima de variables en base a medidas de razón de verosimilitud. En este caso el algoritmo que proponemos realiza una selección de variables hacia

adelante que permite identificar interacciones genéticas de alto orden, es decir, con un número elevado de variables, y abordable desde el punto de vista computacional.

Otro aspecto importante que abordamos en esta tesis es la simulación de datos que cumplan ciertas propiedades similares a datos reales. Los estudios de simulación tienen mucha importancia en el desarrollo y evaluación de nuevas metodologías porque permiten analizar su potencia y efectividad en conjuntos de datos para los que se conoce cuál debería ser el resultado correcto a obtener, permiten comparar los resultados obtenidos con otras técnicas y permiten analizar cómo se ve afectada la efectividad de la metodología al variar los parámetros de simulación. Con este propósito, hemos desarrollado un conjunto de funciones de R para simular los datos de genotipos y la variable respuesta asociada en base a modelos de riesgo.

Resum

L'objectiu de l'epidemiologia genètica és comprendre els efectes dels gens i els factors ambientals en la salut humana. En els últims anys s'han realitzat molts estudis d'associació amb el propòsit d'estudiar el paper de la component genètica en el risc de malalties comunes com la diabetis, malalties cardiovasculars, càncer, Alzheimer, etc. Aquests estudis es basen en general en una selecció de marcadors genètics que es genotipen per a mostres d'individus malalts (casos) i sans (controls) amb la finalitat d'estudiar el nivell d'associació amb la malaltia.

Aquesta tesi es centra en el que es denomina "Genomic Profiling", que es refereix a la identificació de múltiples variants genètiques per predir el risc a la malaltia. Això implica una selecció de variables i la construcció d'un model. Una estratègia comuna consisteix en la construcció d'un model de predicció que contingui únicament els marcadors genètics amb un efecte marginal significatiu sobre la malaltia. Aquest enfocament marginal té una limitació important, ignora l'epistasi. Des d'un punt de vista estadístic, un efecte epistàtic o una interacció genètica apareix quan l'efecte conjunt de diversos marcadors genètics sobre el fenotip observat no queda explicat pels seus efectes marginals. La principal dificultat en l'estudi de les interaccions genètiques és que generalment s'analitzen centenars o milers de marcadors genètics, i això fa inviable l'anàlisi de totes les possibles interaccions, des del punt de vista computacional. Com a conseqüència d'això, la majoria dels mètodes disponibles

per a l'anàlisi d'epistasi només permeten l'estudi de les interaccions d'ordre inferior, interaccions de segon o tercer ordre.

En aquesta tesi presentem dues aproximacions metodològiques per a la identificació de perfils genètics en presència d'epistasi que permeten explorar interaccions d'ordre alt amb un cost computacional raonable.

La primera de les metodologies presentades, denominada AUC-RF, es basa en un mètode de classificació àmpliament utilitzat en el context de les tècniques d'aprenentatge automatitzat denominat Random Forest (RF). L'algorisme AUC-RF implementa una selecció de variables cap a enrere basada en l'optimització de les corbes ROC del RF que permet identificar conjunts de marcadors genètics amb el major valor predictiu sobre el risc a la malaltia.

Un RF proporciona una ordenació de les variables predictores sobre la base d'algunes de les mesures d'importància que el mètode té implementades. Les dues mesures més utilitzades són el mean decrease accuracy (MDA) i el mean decrease Gini (MDG). En aquesta tesi presentem un treball en el que hem explorat l'estabilitat i la robustesa d'aquestes dues mesures d'importància. Els resultats d'aquest estudi posen de manifest que el MDA és molt inestable i que el MDG proporciona uns rànquings més adequats de les variables predictores que realment estan associades amb la variable resposta. Aquest resultat justifica que a la metodologia AUC-RF utilitzem la mesura d'importància MDG.

La segona metodologia proposada, denominada Optimal AUC, es basa en el concepte de corbes ROC òptimes. L'objectiu torna a ser l'obtenció de conjunts de variables genètiques amb la màxima capacitat predictiva, en aquest cas basant-nos en la combinació òptima de variables sobre la base de mesures de raó de versemblança. En aquest cas l'algorisme que proposem realitza una selecció de variables cap a endavant que permet identificar interaccions genètiques d'ordre alt, és a dir, amb un nombre elevat de variables, i possible des del punt de vista computacional.

Un altre aspecte important que abordem en aquesta tesi és la simulació de dades que compleixin certes propietats similars a dades reals. Els estudis de simulació tenen molta importància en el desenvolupament i avaluació de noves metodologies perquè permeten analitzar la seva potència i efectivitat en conjunts de dades pels quals es coneix quin hauria de ser el resultat correcte a obtenir, permeten comparar els resultats obtinguts amb altres tècniques i permeten analitzar com es veu afectada l'efectivitat de la metodologia en variar els paràmetres de simulació. Amb aquest propòsit, hem desenvolupat un conjunt de funcions d'R per simular les dades de genotips i la variable resposta associada en base a models de risc.

Contents

Introduction **1**

1 Bladder Cancer and Alzheimer's Disease Studies **7**

1.1 The Spanish Bladder Cancer/EPICURO Study 8

1.2 Alzheimer Disease Study 11

2 Simulation of genetic risk profiles for binary, continuous and time-to-event phenotypes **15**

2.1 Introduction 16

2.2 Simulation of genotypes 18

2.2.1 Simulation of independent SNPs 18

2.2.2 Simulation of SNPs in LD blocks 18

2.3 Multi-locus models of disease risk for a binary phenotype 20

2.3.1 Multiplicative risk model 21

2.3.2 Multiplicative odds of risk model 23

2.4 Simulation of disease status from a genetic profile 25

2.4.1 Simulation of disease status from independent SNPs 26

2.4.2 Simulation of high-order genetic interactions associated with disease risk 26

2.5 From binary to continuous or survival data 28

2.6 R functions and Examples 29

2.6.1 Function SNPgenerate 29

2.6.2 Function LDgenerate 30

2.6.3 Function RiskGenerate 30

2.6.4 Function SNPInteract 31

2.6.5 Functions Bin2Cont and Bin2Surv 31

3 Stability of Random Forest Importance Measures 33

3.1 Introduction 34

3.2 The Random Forest methodology 35

3.3 Stability of MDA and MDG rankings 36

4 Robust feature selection with Random Forest 43

4.1 Introduction 44

4.2 Methods 45

4.2.1 AUC-RF for feature selection 45

4.2.2 Random Forest parameters 48

4.2.3 Random Forest importance measures 49

4.2.4 Random Forest prediction and AUC computation 49

4.3 Application 51

4.3.1 The Spanish Bladder Cancer/EPICURO Study 51

4.3.2 Alzheimer's disease Study 55

4.3.3 Simulation Study 55

4.4 Discussion 67

5 Identification of high-order genetic interactions using the likelihood-ratio score 69

5.1 Introduction 70

5.2 Optimal prediction and optimal ROC curve 71

5.3 Searching for higher-order interactions:
Optimal AUC algorithm 78

5.3.1 Forward selection process 80

5.3.2 Best model and inference 83

5.3.3 Pruning 85

5.3.4 Missing data 85

5.4 Results 87

5.4.1 Simulation results 87

5.4.2 Alzheimer results 90

Bibliography **92**

A Functions for the simulation of genetic risk profiles **101**

B AUCRF (R language package) **111**

AUCRF manual 113

AUCRF font code 121

C Functions for the identification of high-order interactions using the likelihood-ratio score **131**

List of Figures

3.1 MDG rank in the original dataset against MDG rank in the perturbed datasets (10% left out). 37

3.2 MDA rank in the original dataset against MDA rank in the perturbed datasets (10% left out). 38

3.3 Average percentage of overlap between the original ranking and the ranking in the perturbed datasets (10% left out). MDG (solid line) and MDA (dashed line) 39

4.1 varSelRF backward elimination procedure. 52

4.2 AUC-RF backward elimination procedure. 53

4.3 AUC-RF backward elimination procedure for AD data set. 56

4.4 AUC-RF backward elimination procedure for ApoE-ϵ4 carriers in AD data set. 57

4.5 AUC-RF backward elimination procedure for ApoE-ϵ4 noncarriers AD data set. 58

4.6 Proportion of selected causal SNPs for $k = 50$ and balanced data sets. 66

4.7 AUC of the selected SNPs for $k = 50$ and balanced data sets. 66

5.1 Example of ROC curves. 73

5.2 Optimal ROC curve and ROC curves based on logistic regression for data in Table(5.1). 77

5.3 a) Forward selection curve with $\delta = 0$ (no overfitting correction) b) Forward selection curve with $\delta = 0.001$ (with overfitting correction). . 82

5.4 Forward selection curves for the selection process starting with SNP1 (circles) and for the selection process starting with SNP474 (triangles). 84

List of Tables

1 Genotype codification of a SNP. 2

1.1 SBCS. Distribution of individuals with respect to gender, region, age and exposure to smoking. 10

1.2 SCBS. Distribution of cases and controls by exposure to smoking. . . 11

1.3 SBCS. Distribution of SNPs in the different genes. 12

1.4 AD. Distribution of SNPs in the diferent genes. 14

2.1 Joint probability distribution of genotypes, G_i and the disease indicator Y where p denotes the frequency of the variant allele A. 24

3.1 MDG ranking results for the 10 associated SNPs. 40

3.2 MDA ranking results for the 10 associated SNPs. 41

3.3 MDG and MDA rankings. Probabilities of being in the top-10, -20 or -50 list. 42

4.1 SCBS. Most important SNPs, MDG index and probability of selection (P). 54

4.2 Percentage of selected causal SNPs (Pc) for $RR1 = 1.5$. 60

4.3 Percentage of selected causal SNPs (Pc) for $RR1 = 1.3$. 61

4.4 Percentage of selected causal SNPs (Pc) for $RR1 = 1.1$. 62

4.5 test-AUC of the selected SNPs and score-AUC (in parenthesis) of the causal SNPs for $RR1 = 1.5$. 63

4.6 test-AUC of the selected SNPs and score-AUC (in parenthesis) of the causal SNPs for $RR1 = 1.3$. 64

4.7 test-AUC of the selected SNPs and score-AUC (in parenthesis) of the causal SNPs for $RR1 = 1.1$. 65

5.1 Optimal ROC partition example with two genetic markers. 76

5.2 Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting two interacting causal SNPs. . 89

5.3 Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting three interacting causal SNPs. . 89

5.4 Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting four interacting causal SNPs. . 90

5.5 Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting five interacting causal SNPs. . 90

5.6 Significant sets of SNPs obtained by Optimal AUC algorithm from Alzheimer's disease data set. 92

Introduction

Common diseases such as cancer, diabetes, Alzheimer's or cardiovascular disease are caused by the combination of multiple genetic and environmental factors. Understanding the effects of genes and environmental factors on the development of these complex diseases is a major aim of genetic epidemiology.

In the last few years a large number of association studies have been carried out with the goal of studying the inherited genetic basis of common diseases. A common study design for exploring the genetic basis of human diseases is a case-control study where single nucleotide polymorphisms (SNPs) are genotyped and differences in genotype frequencies between cases and controls are analyzed. An SNP is a polymorphic single nucleotide locus in the genome DNA where different variants (alleles) are observed among the individuals in a population. SNPs are the most simple and common form of genetic variation among individuals. Throughout the human genome there are more than 10 million SNPs and the variation in these polymorphic loci would explain an important part of our individual susceptibility to disease or the different individual responses to treatments. Most SNPs have two possible alleles denoted by an uppercase letter (for example "A" or "B") and a lowercase letter ("a" or "b"). Usually, the uppercase denotes the ancestral allele or just the most frequent allele in the population. Since the human genome is diploid, that is, the DNA is duplicated in each cell of an individual, this yields to three possible genotypes per SNP: "AA"

Table 1: *Genotype codification of a SNP.*

Genotype	X	genotype frequencies
AA	0	p_{AA}
Aa	1	p_{Aa}
aa	2	p_{aa}

for the common homozygous subjects, "Aa" for the heterozygous subjects and "aa" for the variant homozygous subjects. From a statistical point of view an SNP can be thought of as a categorical variable X with three different categories that can be recoded numerically as the number of minor alleles ("a"), that is, zero for "AA", one for "Aa" and two for "aa" (Table 1).

The number of genotyped SNPs for each individual varies from hundreds in candidate gene studies to above 1 million in genome-wide association studies (GWAS). In candidate gene studies SNPs are genotyped in a set of genes that are thought to have some relationship with the disease. Instead, GWAS are designed to cover most of the human genetic variation by genotyping SNPs across the whole genome without any prior hypothesis of causality. GWAS are indirect association studies where the genotyped SNPs act as markers for the nearby region; it is assumed that an associated SNP will be either a disease-causing variant or will be in linkage disequilibrium (LD) with an unmeasured disease-causing variant. Linkage disequilibrium (LD) is a population genetics concept that reflects the non-random association of alleles in two loci. In other words, LD appears when a particular allele at one locus is found together with a specific allele at a second locus more often than expected if the two loci were segregating independently in the population.

So far, many genes and genetic variants associated with disease risk have been identified. However, these well established variants only explain a small proportion of the inferred genetic contribution of disease and discovering the rest of genetic variants

remains a major challenge. The success of genetic association studies depends on both biological and statistical factors: small size effects of individual variants, non-additive multi-factor effects and linkage disequilibrium (LD), among others, may affect the power of the specific study design and the statistical approach chosen for analysis. The genetic architecture of the disease, controlled by high-penetrant mutations, rare disease-causing variants, common susceptibility alleles or a combination of these situations, may also play an important role in the success of association studies.

Though complex diseases are known to be caused by the joint effect of multiple genetic and environmental factors, the usual practice in genetic association studies is a marginal analysis where each genetic marker is tested separately for association with the disease. This marginal strategy is not very powerful since each genetic variant is expected to have a very small marginal effect, only detectable with very large sample sizes. Indeed, most associated variants detected up to now in GWAS have small effect sizes with odds-ratios of disease for the risk allele typically around 1.2 (Wray et al., 2007). Furthermore, this marginal analysis strategy does not take into account possible interactions between the genetic factors, also known as epistasis. Genes do not act in isolation but instead their function depends on many other genes in a network or pathway that interact in a complex way. Thus, ignoring genetic interactions is an important limitation of the marginal analysis strategy which can only be viewed as a preliminary step of the gene identification process (Cantor et al., 2010).

This thesis is the result of my participation in several projects in the Bioinformatics and Medical Statistic group at the University of Vic - Central University of Catalonia. In collaboration with Dr. Núria Malats (CNIO) we have been involved in the Spanish Bladder Cancer/EPICURO study, one of the largest studies on bladder cancer aiming at identifying genetic and environmental factors related to the etiology and progression of bladder cancer. On the other hand, in collaboration with Dr.

Enric Bofill and Pere Roura (Consorci Hospitalary de Vic) we have participated in an Alzheimer's disease (AD) study focused on the identification of genetic variants in the reelin pathway associated with a modification of AD risk. In chapter 1 we introduce the data sets from these two studies, on bladder cancer and Alzheimer disease, which have motivated different parts of the present thesis.

From a methodological point of view, the goal of this thesis is the development of new powerful methods for the identification of genetic interactions (epistasis analysis) or the improvement of existing methodologies. In this work we use the term genetic interactions or epistasis in a rather wide sense referring to nonlinear joint effects that are not captured by a model that only considers additive marginal effects. Several statistical methods have been proposed for epistasis analysis. From exhaustive searches using regression models including interactions (Marchini et al., 2005) to data-mining methods such as the Multi-Dimensional Reduction Method (Ritchie et al., 2001), the Model-Based Multi-Dimensional Reduction method (Calle et al., 2010, 2008), or approaches based on Random Forest (Bureau et al., 2005) and Support Vector Machine (Wei et al., 2009). An overview on this topic is given by Van Steen (2012). The methods proposed in this thesis are related to the Random Forest (chapters 3 and 4) and to the optimal ROC curve (chapter 5).

Before these developments are presented, we describe in chapter 2 some concepts and strategies for simulating data sets from a case-control genetic study with epistasis. We review two genetic disease risk models for independent genetic markers and a binary outcome (disease status) and describe how to use these models for simulating high-order genetic interactions. We also describe how to transform the binary outcome into a continuous or a time-to-event phenotype. We provide R functions for the proposed simulation strategies. These simulation studies are useful for exploring the accuracy, power and improvement of the new proposed methods for epistasis analysis.

In chapters 3 and 4 we centre on the Random Forest (RF) methodology, a classification algorithm developed by Leo Breiman (Breiman, 2001) consisting of the aggregation of multiple classification trees generated from bootstrap samples. Random Forest can be used for ranking variables according to a measure of importance and this ranking can be used for feature selection. In chapter 3 we explore the stability and robustness of two importance measures provided by RF, mean decrease accuracy (MDA) and mean decrease Gini (MDG). In this work we show that MDA is very unstable and that MDG provides better rankings for causal variants. This part of the thesis is published in Calle and Urrea (2011). In chapter 4 we propose a new strategy for variable selection using Random Forest. The proposed algorithm, namely AUC-RF, computes the Receiver Operating Curve (ROC) associated to the Random Forest, uses the area-under-the ROC curve (AUC) as the predictive accuracy of the Random Forest and implements a backward elimination process for selecting the set of variables with the highest AUC value. The goal of this chapter is two-fold: establish AUC as a preferable accuracy measure for Random Forests compared to the usual classification error rate and to provide a new selection algorithm based on the AUC. In particular, we show that the use of the classification error is especially inappropriate when dealing with unbalanced data sets. The new method is published in the Human Heredity journal (Calle et al., 2011) and the algorithm is publicly available as an R package, named AUCRF, at `http://cran.r-project.org/web/packages/AUCRF` and the package documentation can be found in appendix B.

Most statistical methods for epistasis analysis are able to scan for second or third order interactions but, since they are very computationally demanding, they become unfeasible for exploring higher order interactions when the number of variables to explore is large, as is usually the case. In chapter 5 we propose a new strategy for exploring higher order interactions. The method is based on the concept of the likelihood ratio score and optimal ROC curves and follows a process of forward

selection to obtain the subset of factors with the highest joint predictive accuracy. The algorithm, referred to as Optimal AUC, is computationally feasible in a large variable context.

In summary, the major contributions presented in this thesis are two new methodological strategies, the AUC-RF and the Optimal AUC algorithms, for the identification of genetic variants in association studies in the presence of epistasis. The development of these methodologies and the need to assess their accuracy has led to two also remarkable contributions, an in depth study of the stability of the importance measures of the Random Forest methodology and a new strategy for simulating epistasis.

CHAPTER 1

Bladder Cancer and Alzheimer's Disease Studies

This chapter presents a description of the data sets that have been used and which have served as motivation for the different sections of this thesis. The first data set corresponds to the Spanish Bladder Cancer/EPICURO study and the second one to the Alzheimer's Disease study.

1.1 The Spanish Bladder Cancer/EPICURO Study

Bladder cancer is the fifth most commonly diagnostic cancer type in Europe. In Spain about 8.000 new cases are diagnosed each year, and is one of the cancers with a higher prevalence. It is a chronic disease with a survival rate at five years of 70%, which requires strict medical lifelong checks and has a considerable impact on the quality of life of patients. It is one of the cancers with higher health care costs per patient.

The main causes of risk for bladder cancer are tobacco smoking and certain occupational exposure to carcinogens. It is known that men have a higher risk of developing the disease than women and that this risk increases with age.

Several studies also suggest genetic causes that may have some influence on the susceptibility to bladder cancer, as it has detected an increased risk in patients with family history of bladder cancer.

The study of genetic factors associated with susceptibility to this cancer has mainly focused on genes encoding enzymes involved in xenobiotic metabolism (deactivation and removal mechanism of a type of potentially harmful substances), but other types of processes are also of interest as the xenobiotic transport, apoptosis (cell death genetically regulated), cell cycle control, angiogenesis (physiological process of formation of new blood vessels from preexisting vessels), tumor progression or inflammation process.

The Spanish Bladder Cancer/EPICURO Study (SBCS) was initiated in 1997 with the purpose of advancing knowledge of this cancer for improving prevention, prognosis and treatment. Its main objectives are to evaluate the risk of bladder cancer in relation to environmental and occupational exposures, to evaluate the role of lifestyle factors in their etiology and to evaluate the effects of genetic susceptibility markers

on the risk of bladder cancer and their interaction with environmental exposures.

The SBCS is a case-control study conducted in 18 hospitals from 5 areas in Spain (Asturias, Barcelona metropolitan area, Vallès/Bages, Alicante, and Tenerife). Eligible cases were aged 21-80 years and had newly diagnosed, histologically confirmed carcinoma of the urinary bladder during 1998-2001. Patients who had a previous diagnosis of cancer of the lower urinary tract (i.e. bladder,renal pelvis, ureters, or urethra) were not eligible for the study, as were patients with bladder tumors that were secondary to other malignancies. Controls were selected from patients admitted to participating hospitals with diagnoses thought to be unrelated to the exposures of interest, such as tobacco use. Controls were individually matched to the cases for age at interview within 5-year categories, sex, ethnic origin, and region. After having accepted to participate, subjects gave information on their past exposure to several environmental risk factors and provided blood/saliva as a source of genomic DNA.

Participants were classified as never smokers if they had smoked fewer than 100 cigarettes in their lifetime and ever smokers otherwise. Ever smokers were further classified as regular smokers if they had smoked at least 1 cigarette per day for 6 months or longer and occasional smokers otherwise. Current smokers were defined as those regular smokers who had smoked within a year of the reference date; individuals who had smoked regularly but who had stopped smoking more than 1 year before the reference date were defined as former smokers.

In this thesis we use a data set from SBCS consisting of a total of 2299 individuals (1150 controls and 1149 cases). Four environmental variables (gender, region, age and smoking status) and 267 genetic variables. Table 1.1 describes the distribution of individuals with respect to the variables gender, region, age that were used for matching cases and controls.

Table 1.2 shows the distribution of cases and controls by exposure to smoking, which

Table 1.1: *SBCS. Distribution of individuals with respect to gender, region, age and exposure to smoking.*

Variable	Category	Cases	Controls	Total
Gender	Male	1004 (87.3%)	1002 (87.2%)	2006 (87.3%)
	Female	146 (12.7%)	147 (12.8%)	293 (12.7%)
Age	< 55	171 (14.9%)	194 (16.9%)	365 (15.9%)
	55-64	240 (20.9%)	280 (24.4%)	520 (22.6%)
	65-69	254 (22.1%)	263 (22.9%)	517 (22.5%)
	70-74	254 (22.1%)	226 (19.7%)	480 (20.9%)
	≥ 75	231 (20.0%)	186 (16.1%)	417 (18.1%)
Region	Barcelona	209 (18.2%)	232 (20.2%)	441 (19.2%)
	Vallès Occidental/Bages	180 (15.7%)	182 (15.8%)	362 (15.7%)
	Alicante	85 (7.4%)	82 (7.1%)	167 (7.3%)
	Tenerife	211 (18.3%)	191 (16.6%)	402 (17.5%)
	Asturias	465 (40.4%)	462 (40.3%)	927 (40.3%)

Table 1.2: *SCBS. Distribution of cases and controls by exposure to smoking.*

Category	Cases	Controls
Non-smokers	159 (32.0%)	338 (68.0%)
Occasional smokers	50 (36.2%)	88 (63.8%)
Former smokers	447 (51.1%)	428 (48.9%)
Current smokers	494 (62.6%)	295 (37.4%)

is the main risk factor associated with bladder cancer. We observe a ratio of twice as many cases than controls among smokers and this ratio is reversed among non-smokers and former smokers.

The genetic variables corresponds to 267 SNPs genotyped in a total of 106 genes in the inflammatory pathway. Table 1.3 describes the distribution of SNPs in the different genes.

1.2 Alzheimer Disease Study

Alzheimer's disease (AD) is one of the most common neurodegenerative diseases, its prevalence ranges between 20% and 40% in developed countries. The hallmarks of Alzheimer's disease is the extracellular accumulation of β-amyloid plaques, hard and insoluble plaques of protein fragments, and the intracellular accumulation of neurofibrillary tangles, insoluble twisted fibers found inside the brain's cells.

We were involved in the AD project "Genotypes associated with synaptic Neuroplasticity in Alzheimer's disease" in collaboration with Dr. Enric Bofill from the Neurology Service and Pere Roura (Consorci Hospitalari de Vic). The project,

Table 1.3: *SBCS. Distribution of SNPs in the different genes.*

Gene Name	SNPs	Gene Name	SNPs	Gene Name	SNPs
ABCA1	5	EPHX2	1	IRF3	2
ABCA7	2	EXO1	2	JAK3	3
ABCC4	2	FAS	3	LEPR	4
AKR1C3	10	FASLG	1	LITAF	2
AKT1	1	FCGR2A	1	LTA	2
ALOX12	1	FOS	3	MASP1	8
ALOX15	1	GATA3	6	MBL2	8
ALOX5	5	GDF15	2	MPO	2
APOA2	3	GSK3B	7	MSH2	7
ARHGDIB	2	HFE	3	MX1	5
BCL6	4	ICAM1	4	NBS1	2
BIRC2	1	IFNAR2	1	NCF2	2
BPI	1	IFNG	1	NFKB1	3
CARD15	3	IFNGR1	2	NINJ1	2
CASP3	4	IFNGR2	2	NOS2A	3
CASP8	2	IL10	5	OPRD1	2
CBR1	2	IL10RA	2	PARP4	3
CCL5	2	IL12A	1	PTGS1	1
CCND3	1	IL12B	1	PTGS2	4
CCR2	2	IL13	2	RHOA	1
CCR3	2	IL15	3	SCARB1	5
CCR5	2	IL15RA	4	SELE	1
CD14	1	IL1A	2	SFTPD	2
CD4	1	IL1B	3	SLAMF1	3
CD40	1	IL1RN	2	TFF1	1
CD80	3	IL2	2	TFF3	1
CD81	1	IL3	1	TGFB1	2
CD86	2	IL4	2	TGFBR1	2
CFH	5	IL4R	5	TLR2	3
CRP	2	IL6	1	TLR4	1
CSF1R	2	IL6R	1	TNF	5
CSF2	1	IL7R	2	TNFRSF10A	2
CSF3	2	IL8	1	TNFRSF1A	1
CTLA4	6	IL8RA	1	TNIP1	1
CX3CR1	2	IRF1	1	VCAM1	1
				XBP1	2

funded by the UNNIM - Obra Social (UNNIM grant in health sciences 2011), was aimed to explore the role of the genetic variability in the Reelin signaling pathway in Alzheimer's disease. The Reelin signaling pathway contributes to the formation of synaptic circuits in the central nervous system and interacts with ApoE protein, whose ApoE-ϵ4, allele is the best established genetic risk factor for late-onset AD (Rice and Curran, 2001; Seshadri et al., 1995). The specific goals of the study were to analyze any association between the genes involved in the Reelin signaling pathway with Alzheimer's disease, identifying genetic risk profiles of SNPs in genes of the Reelin pathway and to analyze potential interactions between ApoE genotypes and those SNPs.

With this purpose we used the data from a publicly available GWAS (Genome Wide Association Study) conducted by Reiman et al. (2007). The dataset corresponds to a case-control study with 1411 subjects (861 cases and 550 controls) and $502,627$ SNPs genotyped. Since we focus in the Reelin pathway, we extracted 682 SNPs which lie within 32 genes of the Reelin signaling pathway. Table 1.4 describes the distribution of SNPs in the different genes. The dataset also provides an indicator variable of whether the individual is carrying the apolipoprotein variant ApoE-ϵ4.

The dataset was available for download from `http://public.tgen.org/tgen.org/supplementarydata/neurogenomics/supplementarydata/GAB2`.

Table 1.4: *AD. Distribution of SNPs in the diferent genes.*

Gene	Chm	SNPs in gene	SNPs in promotor	Gene	Chm	SNPs in gene	SNPs in promotor
ABL1	9	27	0	LRP2	2	43	1
ABL2	1	6	2	PIK3R1	5	8	2
APOE	19			CDK5R1	17		
APOER2	1	11	0	CDK5R2	2		
APP	21	49	1	TP73	1	4	0
BDNF	11	5	1	AKT1	14	2	1
CAMK2A	5	10	1	PLK2	5	1	0
CASK	X			PSEN1	14	4	0
CDC42	1	10	2	PSEN2	1	5	1
CDK5	7	4	0	RAC1	7	5	0
CNR1	6	6	2	RELN	7	80	2
DAB1	1	251	1	RHO	3	1	1
EMX2	10	0	1	RHOA	3	0	4
EPHA1	7	1	2	SHIP	2	18	0
FYN	6	36	1	SRC	20	4	0
GSK3B	3	7	0	TAU	17	31	0
ITGA3	17	4	6	TBR1	2	2	4
LDLR	19	4	0	VLDLR	9	5	2

CHAPTER 2

Simulation of genetic risk profiles for binary, continuous and time-to-event phenotypes

Simulation studies in genetic epidemiology research are essential for evaluating the accuracy, power and potential improvement of new methodologies for exploring the genetic component of a disease. In this chapter we review two genetic disease risk models for independent genetic markers and binary outcome (disease status) and describe how to use these models for simulating high-order genetic interactions. We also describe how to transform the binary outcome in a continuous or time-to-event phenotype. We provide R functions for the proposed methods.

2.1 Introduction

Simulation studies play an important role in genetic epidemiology research. They are essential for evaluating the accuracy, power and potential improvement of new methodologies for exploring the genetic component of a disease. The simulation of such datasets involves two steps, the simulation of the genetic markers and the simulation of the phenotype. Here we center on the case where the genetic markers are single nucleotide polymorphisms (SNPs). The simulation of genotypes requires the specification of allelic and genotype frequencies and possible correlations between the genetic variables (LD). The simulation of the phenotype, which can be binary, continuous or a time-to-event variable, requires the specification of a disease risk model that relates the genotypes with the phenotype.

Though epistasis analysis is one important research topic, most disease risk models considered in simulation studies assume independence among causal SNPs or, at best, incorporate second-order interactions by adding the corresponding interaction terms in the model. Modeling higher order interactions in this parametric way is complicated. In this chapter, we illustrate these concepts, review two genetic disease risk models for independent genetic markers and binary outcome (disease status) and describe how to use these models for simulating high-order genetic interactions. We also describe how to transform the binary outcome into a continuous or time-to-event phenotype. In appendix A we provide the R language code of the functions for the proposed methods.

There exist several sophisticated methods and programs for simulating genotypes that reproduce the intrinsic architecture of the human genome in a very realistic manner (Li and Li, 2008; Wright et al., 2007). They provide simulated datasets with realistic patterns of LD and allele frequencies by resampling from human genome variation databases, as Hapmap (I.H.C., 2003). This strategy consists on select-

ing a set of SNPs in an LD block at the Hapmap database and, based on their genotype information, estimate the haplotypes frequencies and then generate the individual genotypes by matching randomly pairs of simulated haplotypes. More sophisticated models and software are available for generating simulated data that closely resemble real data, such as HAPGEN2 software that simulates haplotypes by conditioning on a reference set of population haplotypes and an estimate of the fine-scale recombination rate across the region, so that the simulated data has the same LD patterns as the reference data, or the simulation program COSI which implements a coalescent model (Schaffner et al., 2005).

Although these methods are very useful for evaluating the performance of a new approach in a realistic situation, in initial phases of a new methodological development it can be convenient to explore the performance of the new method in a simpler and more controlled framework. With this aim, in section 2.2 we describe simple methods for simulating genotypes where the user can control the allele frequencies and the degree of linkage disequilibrium (LD) between variables. Sections 2.3 to 2.5 are dedicated to the simulation of phenotypes. In section 2.3 we describe two alternative models of genetic disease risk that have been proposed in the literature, the multiplicative model and the multiplicative odds of risk model (Janssens et al., 2006; Wray and Goddard, 2010; Wray et al., 2007). In section 2.4 we propose a strategy for simulating a binary phenotype (disease status) given a specific genetic profile and a risk model. In these simulations the causal genetic variants may either act independently on the risk of disease or have a joint nonlinear interacting effect. In section 2.5 we show how to transform a simulated binary phenotype into a continuous or a time-to-event variable. Finally, in section 2.6 we provide some examples to illustrate the use of a set of R functions that we have developed for this purpose.

2.2 Simulation of genotypes

We consider diallelic loci, for instance, Single Nucleotide Polymorphisms (SNPs). As mentioned in the introduction, we denote by A the wild-type or major allele and by a the variant or minor allele. Let p denote the frequency of the variant allele a. We assume that the genotypes are in Hardy-Weinberg equilibrium (HWE), that is, the genotype frequencies for the three genotypes $(AA,\ Aa,\ aa)$ are $(1-p)^2$, $2p(1-p)$ and p^2, respectively.

2.2.1 Simulation of independent SNPs

The simulation of independent SNPs can be easily implemented. From a statistical point of view a SNP is a categorical variable with three different categories that can be recoded numerically as the number of minor alleles in the locus, that is, zero for AA, one for Aa and two for aa. The individual genotypes for a SNP in a sample are generated as i.i.d. realizations of a random variable following a binomial distribution with size equal to 2 and probability of success equal to p, the frequency of the minor allele.

2.2.2 Simulation of SNPs in LD blocks

Linkage disequilibrium (LD) is a population genetics concept that reflects the non-random association of alleles in two loci. In other words, LD appears when a particular allele at one locus is found together with a specific allele at a second locus more often than expected if the two loci were segregating independently in the population.

There are different measures to characterize the strength of LD between two loci. For two diallelic loci with minor alleles a and b and minor allele frequencies p_a and

p_b, respectively, the most simple and intuitive measure of LD is D, the difference between the observed haplotype frequencies and the expected frequencies assuming independence ($D = p_{AB} - p_A \cdot p_B = p_{ab} - p_a \cdot p_b$). A standardized version of D is given by the correlation coefficient defined as $r = D/\sqrt{p_A \cdot p_a \cdot p_B \cdot p_b}$.

As mentioned before, the goal of the following proposal for simulating SNPs in LD is not to obtain a realistic genetic dataset that mimics the LD structure in the human genome. Instead, our goal is to simulate genetic data sets of SNPs including correlations between them in a controlled manner, where the allele frequencies and the degree of LD between SNPS are specified by the user.

Given a first SNP (SNP1), we can simulate a second SNP (SNP2) with a specified LD correlation r with SNP1 by sampling from the conditional distributions of the genotypes of SNP2 given the genotypes of SNP1. These conditional distributions can be derived from the definition of D assuming HWE, and are given by:

$$
\begin{aligned}
P(BB \mid AA) &= (p_A \cdot p_B + D)^2 \,/\, {p_A}^2, \\
P(Bb \mid AA) &= 2(p_A \cdot p_B + D)(p_A \cdot p_b - D) \,/\, {p_A}^2, \\
P(bb \mid AA) &= (p_A \cdot p_B - D)^2 \,/\, {p_A}^2,
\end{aligned}
$$

$$
\begin{aligned}
P(BB \mid Aa) &= (p_A + p_B + D)(p_a \cdot p_b - D) \,/\, (p_A \cdot p_a), \\
P(Bb \mid Aa) &= [(p_A + p_B + D)(p_a \cdot p_b + D) + (p_A + p_B - D)(p_a \cdot p_b - D)] \,/\, (p_A \cdot p_a), \\
P(bb \mid Aa) &= (p_A + p_B - D)(p_a \cdot p_b + D) \,/\, (p_A \cdot p_a),
\end{aligned}
$$

$$
\begin{aligned}
P(BB \mid aa) &= (p_a \cdot p_B - D)^2 \,/\, {p_a}^2, \\
P(Bb \mid aa) &= 2(p_a + p_B - D)(p_a \cdot p_b + D) \,/\, {p_a}^2, \\
P(bb \mid aa) &= (p_a \cdot p_b + D)^2 \,/\, {p_a}^2.
\end{aligned}
$$

Since $D = r\sqrt{p_A \cdot p_a \cdot p_B \cdot p_b}$, the conditional distributions of the genotypes of SNP2

given the genotypes of SNP1 are determined by the correlation coefficient r and the allele frequencies.

Thus, a simple method for simulating a set of SNPs that form a haplotype block of SNPs in LD is to start by simulating a first SNP that is used as a reference and then generate consecutively the other SNPs in the block according to a specified LD correlation with the reference SNP and using the above conditional distributions. This has the limitation that we can only control the correlation of each SNP in the block with a reference SNP, but not the correlation among the rest of SNPs. In this sense, it does not perfectly mimic genotype data, but it is a way to introduce structures of LD in the data that are controlled with just one parameter.

2.3 Multi-locus models of disease risk for a binary phenotype

Once the genotypes of M SNPs have been generated for a sample of individuals, we will simulate the phenotype, in this case, a binary phenotype Y denoting diseased ($Y = 1$) and non-diseased ($Y = 0$). We assume that a subset of the total available SNPs are associated with disease (for simplicity we will call them causal or risk SNPs) and the rest are not associated with Y (null SNPs). We denote by m the number of causal SNPs with $m \leq M$.

The simulation of Y requires the specification of a genetic disease risk model, a model that describes the individual genetic risk of an individual on the basis of his/her genotypes at the m genetic causal loci. We describe below two alternative multi-locus disease models, both multiplicative risk models in different scales. The first model, referred simply as multiplicative risk model, is multiplicative in the risk scale while the second, referred as odds of risk model, is multiplicative in the odds

scale.

For each causal loci $i = 1, \ldots, m$ we denote by a_i and A_i the variant and wild-type alleles, respectively, and p_i the frequency of the variant allele. We assume HWE. For simplicity of explanation we will describe the models under an additive mode of inheritance for each locus where r_i is the relative risk of the heterozygous group $A_i a_i$ with respect to the reference group $A_i A_i$ and r_i^2 is the corresponding relative risk for the minor homozygous genotypes $a_i a_i$, that is:

$$r_i = P(Y = 1 \mid A_i a_i)/P(Y = 1 \mid A_i A_i),$$
$$r_i^2 = P(Y = 1 \mid a_i a_i)/P(Y = 1 \mid A_i A_i).$$

2.3.1 Multiplicative risk model

The multiplicative risk model for independent loci under an additive mode of inheritance assumes that each risk allele increases multiplicatively the baseline genetic risk of disease. Given a genomic profile $(G_1, \ldots, G_m)$ and Y, the indicator of disease, the multiplicative model can be expressed as:

$$P(Y = 1 \mid G_1, \ldots, G_m) = g_0 \prod_{i=1}^{m} r_i^{X_i} \tag{2.1}$$

where X_i is the number of risk alleles, 0, 1, or 2, at the ith locus and $g_0 = P(Y = 1 \mid G_1 = A_1 A_1, \ldots, G_m = A_m A_m) = P(Y = 1 \mid X_1 = \cdots = X_m = 0)$ is the baseline disease risk, that is, the disease risk of those individuals with no minor allele in the m loci.

If we assume that the relative risk at each loci is the same, say r, then the multiplicative model reduces to

$$P(Y = 1 \mid G_1, \ldots, G_m) = g_0 \cdot r^X \tag{2.2}$$

where $X = X_1 + \cdots + X_m$ is the total number of minor alleles across the m loci (Wray et al., 2007).

Under model (2.2) and assuming equal allele frequency p across loci, the prevalence of the disease (K) and the broad sense heritability (h^2) can be explicitly expressed as a function of the number of loci, m, the minor allele frequency, p, the relative risk, r, and the baseline risk, g_0:

$$K = P(Y = 1) = g_0(1 + p(r - 1))^{2m} \tag{2.3}$$

and

$$h^2 = \frac{K}{(1 - K)} \cdot \frac{(1 + p(r^2 - 1))^{2m}}{(1 + p(r - 1))^{4m}} - 1 \tag{2.4}$$

where the heritability, h^2, indicates the fraction of the phenotype variability that can be attributed to the genetic variation or the proportion of disease cases attributable to genetic effects (Wray et al., 2007).

Expressions (2.3) and (2.4) can also be used to derive the number of risk loci underlying a complex disease, that is, the number of loci that explains the estimated prevalence and heritability of a given disease for different values of the risk allele frequency and the relative risk:

$$n = \frac{1}{2} \cdot \frac{\ln[h^2 + (1 - h^2)K] - \ln(K)}{\ln[1 + p(r^2 - 1)] - 2\ln[1 + p(r - 1)]}. \tag{2.5}$$

This expression, together with the low relative risk observed in empirical studies, suggests that most common diseases might be affected by a large number of loci (Wray et al., 2007).

The multiplicative risk model is intuitive and attractive for its simplicity; however, it has a major drawback: equations (2.1), (2.2) or (2.3) are not restricted to be in the plausible range of values of a probability and under some combinations of model

parameters the probability of disease can take values larger than 1 (Wray et al., 2010). For this reason, alternative models such as the odds of risk model have been proposed (Janssens et al., 2006; Wray et al., 2010).

2.3.2 Multiplicative odds of risk model

The multiplicative odds of risk model for independent loci (Janssens et al., 2006; Wray and Goddard, 2010) assumes a multiplicative model for the likelihood ratios (LR). In other words, the model assumes that the LR of a genetic profile can be obtained by multiplying the LRs of the individual genotypes:

$$LR(G_1, \ldots, G_m) = \prod_{i=1}^{m} LR_i \tag{2.6}$$

where

$$LR(G_1, \ldots, G_m) = \frac{P(G_1 = g_1, \ldots, G_m = g_m \mid Y = 1)}{P(G_1 = g_1, \ldots, G_m = g_m \mid Y = 0)}$$

and

$$LR_i = \frac{P(G_i = g_i \mid Y = 1)}{P(G_i = g_i \mid Y = 0)}, \quad i = 1, \ldots, m.$$

The odds of disease, $P(Y = 1 \mid G_1, \ldots, G_m) \,/\, [1 - P(Y = 1 \mid G_1, \ldots, G_m)]$, is obtained by multiplying the prior odds by the likelihood ratio (LR). The prior odds can be calculated from the prevalence of disease as $K/(1-K)$. Thus, the odds of disease is:

$$\text{odds} = \text{prior odds} \cdot LR(G_1, \ldots, G_m) = \frac{K}{(1-K)} \cdot LR(G_1, \ldots, G_m).$$

Under the multiplicative odds risk model, the odds of disease is given by

$$\text{odds} = \frac{K}{(1-K)} \cdot \prod_{i=1}^{m} LR_i. \tag{2.7}$$

On the other hand, the individual risk of disease given a genomic profile can be expressed in terms of the odds of disease:

$$P(Y = 1 \mid G_1, \ldots, G_m) = \frac{\text{odds}}{1 + \text{odds}}. \tag{2.8}$$

Unlike the multiplicative model, this expression guarantees that under the odds of risk model the probability of disease takes values in $[0,\ 1]$.

The multiplicative odds of risk model is described in Janssens et al. (2006) and expression (2.6) is further developed in terms of odds-ratios. Here we describe the model in terms of relative risks.

Table 2.1: *Joint probability distribution of genotypes, G_i and the disease indicator Y where p denotes the frequency of the variant allele A.*

G	X	$Y = 1$	$Y = 0$	$P(G)$
AA	0	p_{01}	p_{00}	$f_0 = (1-p)^2$
Aa	1	p_{11}	p_{10}	$f_1 = 2p(1-p)$
aa	2	p_{21}	p_{20}	$f_2 = p^2$
(Total)		K	$1-K$	1

The LR at each single loci, $LR_i = P(G_i = g_i \mid Y = 1) \,/\, P(G_i = g_i \mid Y = 0)$, can be obtained from the joint probability distribution of genotypes, G_i, and the disease indicator Y, shown in Table 2.1, where the values in the table can be derived from the disease model parameters, prevalence, K, risk allele frequency, p, relative risk,

r, and are given by the following expressions:

$$\begin{aligned} p_{01} &= K \cdot f_0 \,/\, (f_0 + r \cdot f_1 + r^2 \cdot f_2), \\ p_{11} &= p_{01} \cdot r \cdot f_1 \,/\, f_0, \\ p_{21} &= p_{01} \cdot r^2 \cdot f_2 \,/\, f_0, \\ p_{00} &= f_0 - p_{01}, \\ p_{10} &= f_1 - p_{11}, \\ p_{20} &= f_2 - p_{21}. \end{aligned}$$

Then, the likelihood ratios at each locus for the three possible genotypes are:

$$\begin{aligned} LR(X=0) &= \frac{P(G=AA \mid Y=1)}{P(G=AA \mid Y=0)} = \frac{p_{01} \cdot (1-K)}{p_{00} \cdot K}, \\ LR(X=1) &= \frac{P(G=Aa \mid Y=1)}{P(G=Aa \mid Y=0)} = \frac{p_{11} \cdot (1-K)}{p_{10} \cdot K}, \\ LR(X=2) &= \frac{P(G=aa \mid Y=1)}{P(G=aa \mid Y=0)} = \frac{p_{21} \cdot (1-K)}{p_{20} \cdot K}. \end{aligned}$$

2.4 Simulation of disease status from a genetic profile

Once the genotypes have been generated we proceed with the simulation of the response variable Y, the indicator of the presence or absence of disease. In subsection 2.4.1 we describe how to simulate disease status when the risk SNPs are independent. Instead, in subsection 2.4.2 we propose a new strategy for simulating the response variable from a set of interacting SNPs.

2.4.1 Simulation of disease status from independent SNPs

For each combination of genotypes of the m risk loci, $(G_1, \ldots, G_m)$, we obtain the individual risk of disease $P(Y = 1 \mid G_1, \ldots, G_m)$ from equation (2.1), if we assume the multiplicative risk model, or from equation (2.8), if we consider the multiplicative odds of risk model. For these two models to be valid it is necessary that the m risk loci are mutually independent. We assume that each risk locus is in a different LD block but they may be in LD with other non-susceptible loci. Then, as in Janssens et al. (2006), we generate a value u from a Uniform (0, 1) distribution and if the individual risk of disease is larger than u we set $Y = 1$ for this individual and $Y = 0$, otherwise. That is, $Y = 1$ if $P(Y = 1|G_1, \ldots, G_m) > u$ and $Y = 0$, otherwise.

2.4.2 Simulation of high-order genetic interactions associated with disease risk

One of the challenges of genetic association studies is the identification of interactions between genetic variants that could explain part of the genetic risk that remains unknown. Many methods have been recently proposed for epistasis analysis and the performance of these methods should be explored and compared through simulation studies. Here we propose a simple strategy for simulating high-order interactions. Note that we use the term interaction in a very broad sense, meaning that the joint effect of a set of SNPs does not correspond with the aggregation (multiplicatively or additively) of the individual effects of each SNP.

Given m loci, an interaction of order m associated with disease risk can be simulated by considering a latent variable L that assigns a value 0, 1 or 2 to each multi-locus genotype $(G_1, \ldots, G_m)$ and this value corresponds to the disease risk of the multi-locus genotype: 0 for low risk, 1 for intermediate risk and 2 for high risk of disease. We denote by p_0, p_1 and p_2 the proportion of multi-locus genotypes in each risk group

(below is described how we set these proportions). The values of L, 0, 1 and 2, can be randomly assigned to the different multi-locus genotypes in order that the joint multilocus effect does not correspond with the aggregation of the individual effects. This procedure might be useful if we want to explore the performance of a method for identifying genetic interactions in the absence of main effects. Alternatively, the risk groups can be assigned according to some function of the genotypes, for instance, the score function $S = \sum_{i=1}^{m} X_i$ that counts the total number of variant alleles for this genotype. In this case the joint effect will correspond to the accumulation of individual effects.

The newly generated latent variable L, that takes values 0, 1 and 2, is then treated as if it was a SNP for generating the phenotype Y following the strategy described in subsection 4.1. That is, we obtain the individual risk of disease given the latent variable, $P(Y = 1 \mid L)$, from equation (2.1) or (2.8), depending on the assumed disease model, and from this probability we generate Y as described above. This requires the specification of three model parameters: disease prevalence, relative risk, r, and frequency, p. Here r is the relative risk of individuals in the intermediate risk group with respect to the low risk group. In this case the frequency p is not the risk allele frequency but a parameter that we use to specify the frequency of the 3 risk groups: Given p we define the frequency of the low risk as $p_0 = (1-p)^2$, the frequency of the intermediate risk group as $p_1 = 2p(1-p)$ and the frequency of the high risk group as $p_2 = p^2$ (as if they were the genotype frequencies of a SNP in HWE).

The disease risk phenotype can also be simulated from a set of J independent interacting blocks of SNPs. For each block of SNPs $j = 1, \ldots, J$, we create a latent variable L_j, as described previously, and simulate the phenotype given the latent variables, $P(Y = 1 \mid L_1, \ldots, L_J)$, from equation (2.1) or (2.8) according to the assumed disease model.

2.5 From binary to continuous or survival data

Following (Gui et al., 2011), we can exploit the procedures and models described in the previous sections for simulating continuous and time-to-event phenotypes.

Given a set of genetic variables we first generate a binary outcome Y based on a specified disease risk model. We interpret the generated binary variable not as the disease status indicator but as an indicator of disease risk: $Y = 1$ corresponds to higher risk of disease and $Y = 0$ to lower risk. We specify two different distributions for the phenotype of these two groups of risk and generate the phenotype value for each individual according to its Y risk indicator.

For a continuous phenotype X we will typically assume two normal distributions $X_1 = N(\mu_1, \sigma_1)$ when $Y = 1$ and $X_0 = N(\mu_0, \sigma_0)$ with $\mu_1 > \mu_0$ if larger values of X increase the risk of disease. Of course, alternative distributions are possible.

For a time-to-event phenotype T we will assume the usual distributions for survival data, for instance, the Weibull distribution. In this case, $T_1 \sim Weib(\alpha_1, \beta_1)$ if $Y = 1$ and $T_0 \sim Weib(\alpha_0, \beta_0)$ if $Y = 0$ with the parameterization $Weib(a, b)$ corresponding to a hazard function $h(t) = \frac{b}{a^b} t^{b-1}$ where a is the scale parameter and b the shape parameter. Assuming the same shape parameter beta for T_1 and T_2, corresponds to assuming proportional hazards. In this case, the hazard ratio between the two groups of individuals is equal to $HR = (\alpha_1 \,/\, \alpha_2)^{\beta}$. This expression is useful for specifying the parameters of T_1 and T_2 given the hazard ratio.

In addition to the survival times we also need to sample censoring times for a censoring variable C that indicates the follow-up time of each individual. We can assume any positive distribution for C. Typical choices are the uniform distribution, the exponential distribution and the Weibull distribution.

Then, the observed survival data consists of the pair (T^*, δ) where $T^* = \min(T, C)$

and $\delta = 1\{T \leq C\}$ indicates if the data is observed, $\delta = 1$, or right censored, $\delta = 0$. The proportion of censoring is given by $P(T > C)$ and a pre-specified censoring proportion can be used to determine the parameters of the distribution of C.

2.6 R functions and Examples

In this section we provide some examples that illustrate the new developed R functions for the different sections of this paper.

2.6.1 Function SNPgenerate

This function generates the genotypes of n individuals for a set of independent SNPs according to the specified minor allele frequencies (MAF) as described in section 2.2.1. For example, the syntax for simulating the genotypes of a SNP with MAF equal to 0.3 for 100 individuals is:

```
SNPgenerate(n=100, maf=0.3)
```

The `maf` argument can be a vector specifying the MAF of several SNP. The length of maf defines the number of SNPs to be simulated. The following example produces a data set with four SNPs with specified MAFs:

```
SNPgenerate(n=100, maf=c(0.2, 0.3, 0.3, 0.4))
```

The following example generates a data set with 30 SNPs with random MAFs:

```
SNPgenerate(n=100, maf=runif(30, min=0.1, max=0.5))
```

2.6.2 Function LDgenerate

Given a first SNP, this function generates new SNPs with specified MAFs and with specified LD correlation with the initial SNP as described in section 2.2.2. For example, the code for simulating a SNP with MAF equal to 0.2 and LD with `snp1` equal to 0.9 is:

```
LDgenerate(x=snp1, r=0.9, maf=0.2)
```

As in the previous functions, `r` and `maf` arguments can be vectors of equal length, and the length of these vectors determines the number of SNPs that are simulated. Next example generates a bloc of 11 SNPs in LD with `snp1`, according to a decreasing LD correlation sequence and random MAFs between 0.2 and 0.4:

```
LDgenerate(x=snp1, r=seq(from=0.9, to=0.7, length.out=11),
           maf=runif(n=11, min=0.2, max=0.4))
```

2.6.3 Function RiskGenerate

Given a dataset of genotypes for n individuals, this function generates the individual genetic risk and the disease status as detailed in section 2.4.1 and assuming the multiplicative odds of risk model. The following sentence generates the individual disease status from the genotypes in `SNPdata` with disease prevalence equal to 0.1 and assuming that all the SNPs in `SNPdata` are causal and have the same relative risk equal to 2:

```
status <- RiskGenerate(data=SNPdata, RR=2, p=0.1)
```

We can specify a subset of SNPs to be causal using the argument `data`. For example, we can simulate a dataset where `SNP3`, `SNP10` and `SNP14` are causal SNPs:

```
status <- RiskGenerate(data=SNPdata[,c("SNP3","SNP10","SNP14")], RR=2,
   p=0.1)
```

In this case, different values of RR may also be specified for each SNP, by specifying a vector for RR of length equal to the number of SNPs:

```
status <- RiskGenerate(data=SNPdata[,c("SNP3","SNP10","SNP14")],
                       RR=c(2,1.5,1.3), p=0.1)
```

2.6.4 Function SNPInteract

This function simulates disease status from a set of interacting SNPs as described in section 2.4.2. It requires the specification of the set of interacting SNPs, the relative risk (`RR`), the prevalence of the disease (`p`) and the proportion of individuals in the high risk group (`hrp`). The following R instruction will simulate the disease status associated to an interaction of the SNPs in a data set named `SNPdata`, with a relative risk equal to 1.5, a prevalence equal to 0.1, and a proportion of individuals in the high risk group equal to 0.3:

```
status <- SNPInteract(data=SNPdata, RR=1.5, p=0.1, hrp=0.3)
```

If the interacting SNPs are only a subset of all available SNPs, the value in data argument must be restricted. The following example illustrates the disease status generated from an interaction among the first four SNPs:

```
status <- SNPInteract(data=SNPdata[,1:4], RR=1.5, p=0.1, hrp=0.3)
```

2.6.5 Functions Bin2Cont and Bin2Surv

As described in section 2.5, it is possible to generate a continuous or survival time variable from a binary response Y. We have implemented two functions for doing this. For the continuous case, function `Bin2Cont` transforms the binary variable Y to a continuous variable using two normal distributions, one for individuals with Y=0 and the other for individuals with Y=1. The mean and standard deviation of

the two normal distributions must be specified. In the following example, the means are equal to 40 and 60 and the standard deviations are equal to 10 in both groups:

```
Z <- Bin2Cont(Y, mean0=40, sd0=10, mean1=60, sd1=10)
```

The `Bin2Surv` function is similar. It transforms the binary variable Y to a time-to-event variable T assuming two Weibull distributions, one for individuals with $Y = 0$ and the other for individuals with $Y = 1$. The parameters of the Weibull distribution must be specified. It is required and additional argument, `tmax`, that specifies the maximum follow-up time. Censoring times C are simulated according to a uniform distribution in the interval [0, `tmax`]. The output of this function is a data frame with two variables, the observed survival time, $T^* = \min(T, C)$, and the observed event indicator, $1\{T \leq C\}$. Below, there is an example where the scale parameter of the two groups is equal to 40 (low risk group) and equal to 30 (high risk group) while the shape parameter is equal for the two groups (this last condition implies proportional hazards between the two groups):

```
S <- Bin2Surv(Y, tmax=30, shape0=1.3, scale0=40, shape1=1.3, scale1=30)
```

CHAPTER 3

Stability of Random Forest Importance Measures

Genetic variants are usually ranked and selected according to their $p - value$ obtained from a single-SNP analysis. This approach ignores possible epistasis and only captures marginal effects. An alternative is to rank the variables according to Random Forest importance measures, these ranks would incorporate possible non-linear effects among the variables. Random Forest provides two possible importance measures, mean decrease accuracy (MDA) and mean decrease Gini (MDG). In this chapter we explore the stability of these two importance measures. We illustrate with a real and a simulated example that ranks based on the MDA are unstable to small perturbations of the dataset while ranks based on the MDG provide more robust results. The content of this chapter has been published as a letter to the editor in *Briefings in Bioinformatics* (Calle and Urrea, 2011).

3.1 Introduction

Random Forest (Breiman, 2001) is a popular and widely used supervised learning method consisting a large set of tree-based models (regression or classification trees). Predictions with Random Forest are obtained by averaging the predictions of the different trees. In addition, it provides different measures of importance for each variable. The R package `randomForest` (Liaw and Wiener, 2002), available at `http://cran.r-project.org`, implements this method and provides two different importance measures, mean decrease accuracy (MDA) and mean decrease Gini (MDG), that can be used for ranking variables and for variable selection.

We were exploring the ability and limitations of Random Forest for identifying the genetic component of susceptibility and prognosis of bladder cancer in the EPICURO/Spanish Bladder Cancer Study (García-Closas et al., 2005; Guey et al., 2010), when the *Briefings in Bioinformatics* journal published the paper *Stability and aggregation of ranked gene lists* by Boulesteix and Slawski (2009), and we decided to explore the stability of both MDA and MDG rankings in our dataset using different graphical and numerical descriptive measures proposed in the paper.

The goal of this chapter is to emphasize the importance of exploring ranking stability when using the importance measures, MDA and MDG, provided by Random Forest. We show that ranks based on the MDA are unstable to small perturbations of the dataset and ranks based on the MDG provides more robust results. We illustrate this with the real data of the Spanish Bladder Cancer Study introduced in Chapter 1 and with a simulated example.

3.2 The Random Forest methodology

Random Forest performs a bagging (bootstrap aggregation) of tree-based models (regression and classification trees) and additional modifications that improve its performance. Bagging is an ensemble technique for reducing the variance of an estimated prediction function and avoiding overfitting. Random Forest introduces an additional layer of randomness to bagging by changing the construction of the tree models. In standard classification and regression trees (CARTs), each node is split using the best split among all variables and a post-pruning process is made after tree construction. In Random Forest each tree is fully grown, without pruning, and at each step of the tree growing process, the best split for each node is chosen from a random subset of variables. The following scheme describes the Random Forest algorithm:

- Draw B bootstrap samples from original data.
- For each bootstrap sample grow an unpruned classification or regression tree T_b by the following steps:
  - Select a subsample of m variables from the whole set of variables available.
  - Choose the best split from among the m selected variables.
  - Split the node into two nodes.
  - For the new nodes, recursively repeat the previous steps until the terminal nodes containing a single element or a predefined minimum node size is reached.
- Output the ensemble of the T_b trees, $b = 1, \ldots, B$

The `randomForest` implementation provides two measures of the importance of the predictor variables, the mean decrease accuracy (MDA) and mean decrease Gini

(MDG). MDA quantifies the importance of a variable by measuring the change in prediction accuracy when the values of the variable are randomly permuted compared to the original observations. MDG is the sum of all decreases in Gini impurity due to a given variable (when this variable is used to form a split in the Random Forest), normalized by the number of trees. Most of the papers applying the Random Forest methodology that we have consulted use the MDA ranking, usually without any justification of this choice.

3.3 Stability of MDA and MDG rankings

Our data consists of 723 SNPs in the inflammatory pathway, acting as independent variables in the Random Forest, and a binary dependent variable indicating the recurrence of the tumour in the first five years after diagnosis. Our surprise was the different behaviour in stability displayed by MDA and MDG rankings. Indeed, while MDG was robust to small perturbations of the data, MDA rankings behaved completely unstable. In Figure 3.1 and Figure 3.2 we show the scatter-plot among the original ranking, based on the original dataset, (x-axis) and 100 Jackknife rankings (y-axis) where, for each Jackknife sample, 10% of the observations were randomly selected and removed from the dataset. In Figure 3.1, it is clear that MDG perturbed and original rankings are correlated and that the stability is more important in the tail of the original ranking. Instead, in Figure 3.2, one can realize that, after a small perturbation of the data, any variable, irrespective of its original MDA ranking, can virtually have any MDA ranking in the perturbed sample.

In addition to the scatter plots, we also explored the average percentage of overlap in the top-k list between the original rankings and the rankings of the perturbed datasets as a function of k, for both ranking methods (Figure 3.3). MDG reaches a stable overlap around 75% for k larger than 25 while MDA maximum coverage is

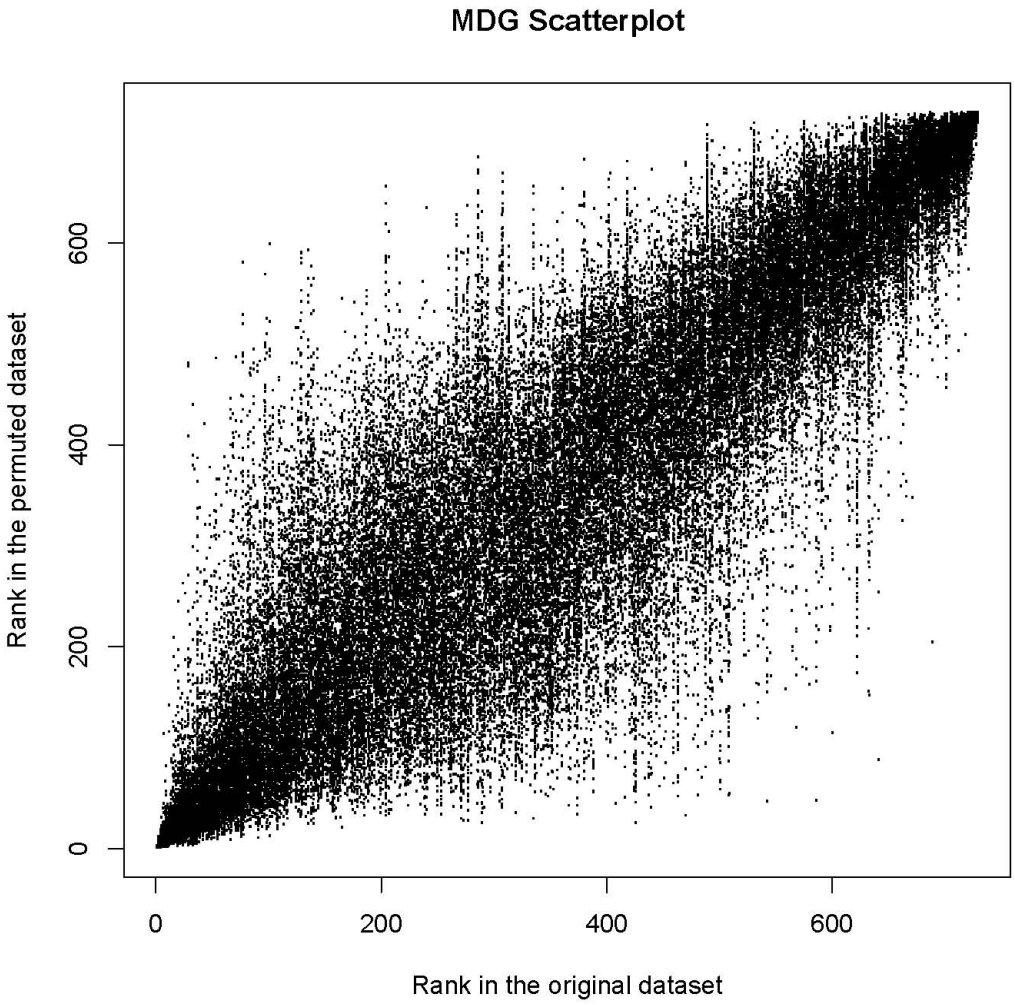

Figure 3.1: *MDG rank in the original dataset against MDG rank in the perturbed datasets (*10% *left out).*

around 60% for k approximately equal to 13, but decreasing quickly to only a 50% as k increases.

Similar results were obtained when we applied the Random Forest algorithm to the analysis of susceptibility of bladder cancer (the same set of SNPs and a case-control dependent variable). Again, MDG rankings performed much better in terms of stability than MDA rankings (data not shown).

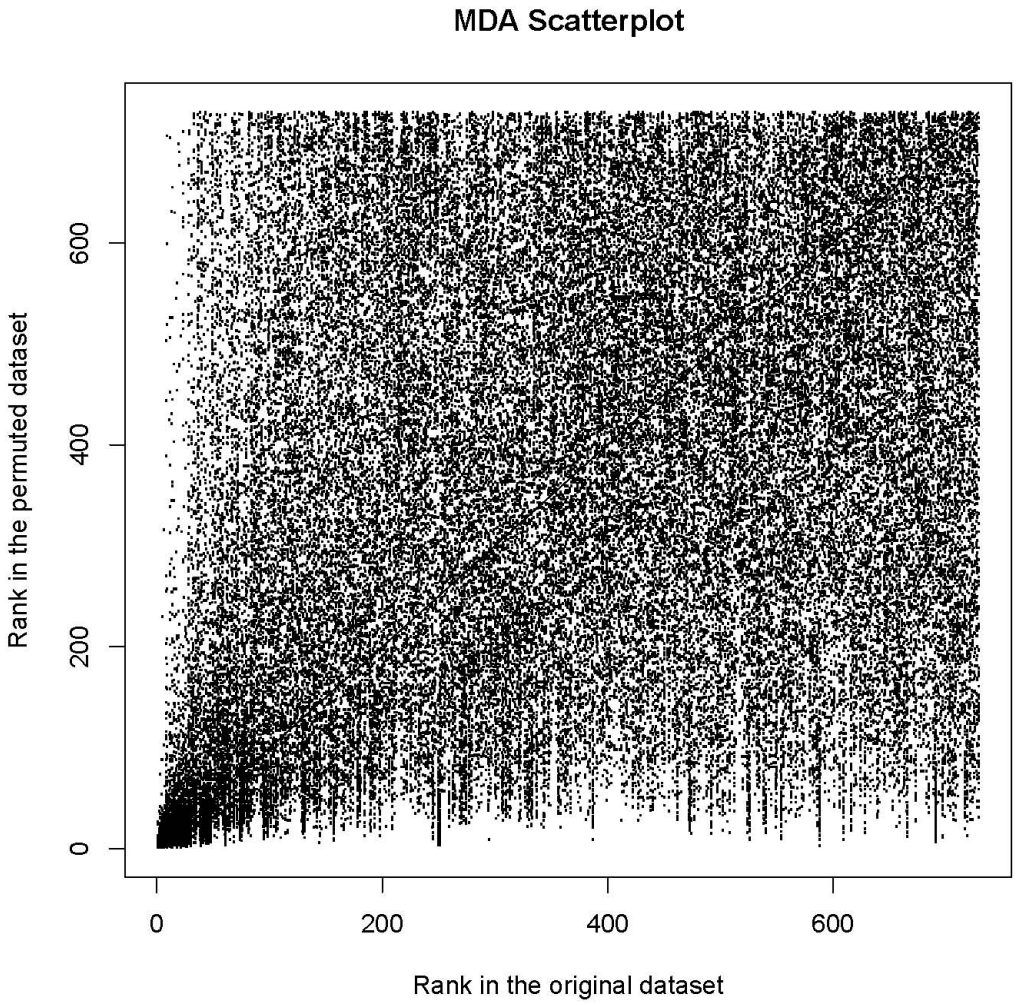

Figure 3.2: *MDA rank in the original dataset against MDA rank in the perturbed datasets (*10% *left out).*

However, as it is also discussed in Boulesteix's paper, though stability is a necessary property of a good ranking procedure, stability alone does not ensure a good behaviour of the ranking in the sense that it may not identify the correct variables. To explore the ability of both ranking measures to capture real known associations, we performed a small simulation. We simulated a dataset, similar to our bladder cancer data set, with 1,000 SNPs and a binary dependent variable. Ten SNPs were associated with the response, following a recessive model with a specified odds-ratio,

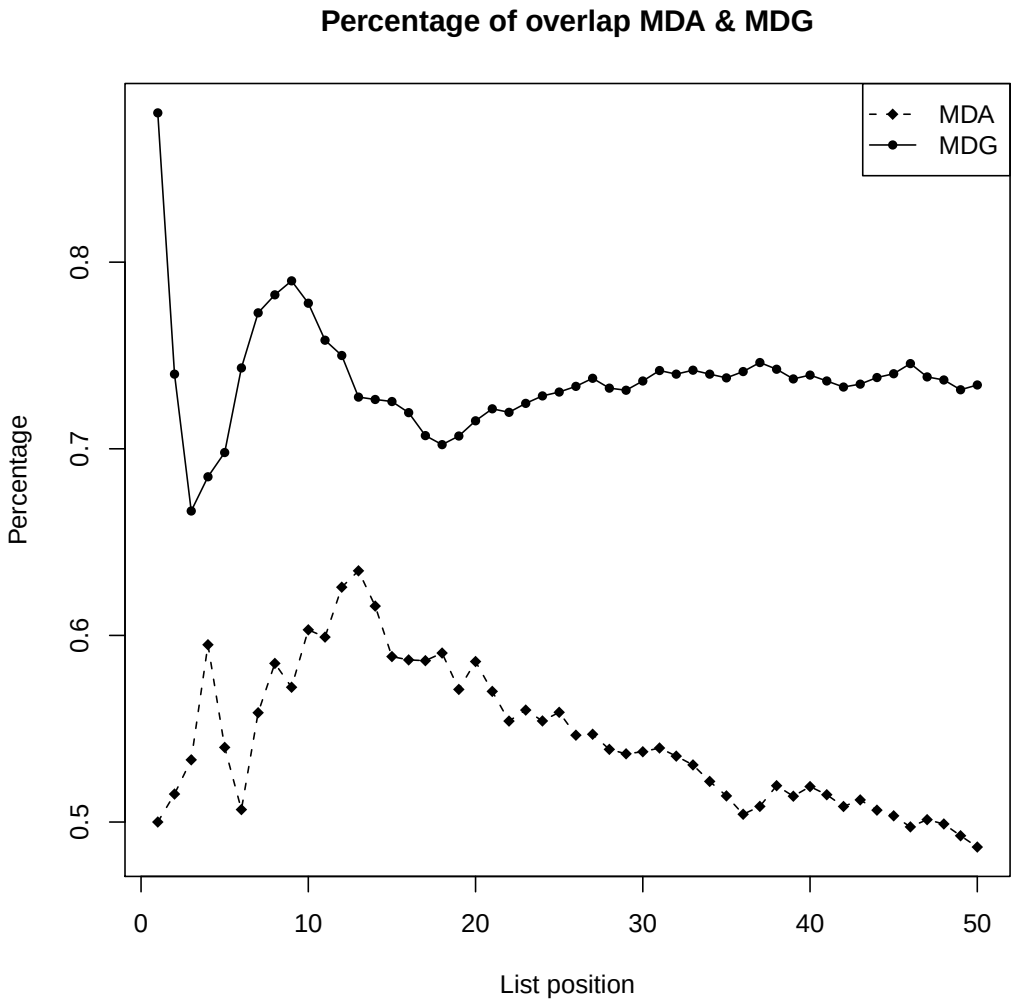

Figure 3.3: *Average percentage of overlap between the original ranking and the ranking in the perturbed datasets (*10% *left out). MDG (solid line) and MDA (dashed line)*

and the rest were generated at random with different minor allele frequencies.

Table 3.1 and Table 3.2 show, for each of the ten associated SNP, its OR, p-value, original ranking, and a summary of the perturbed rankings (min, max and quartiles) for MDG and MDA, respectively. As before, also in this simulated dataset, we observe much more variability of the MDA ranking than the MDG ranking. Also,

Table 3.1: *MDG ranking results for the* 10 *associated SNPs.*

			MDG original	MDG Jackknife rankings summary				
SNP	OR	P-value	ranking	Min	p25	p50	p75	Max
SNP1	3.06	1.06E-05	1	1	1	1	1.25	11
SNP2	2.69	4.25E-05	2	1	2	3	5	25
SNP3	2.54	6.12E-05	3	1	2	3	5	11
SNP4	2.32	0.000115	4	1	3	5	7	29
SNP5	2.16	0.000195	5	1	3	5	8	31
SNP6	1.94	0.000439	6	1	7	10	18	44
SNP7	1.72	0.000570	7	2	4.75	7	10	37
SNP8	1.53	0.001565	15	3	10	14.5	25	90
SNP9	1.31	0.005942	55	9	30	43.5	62.25	263
SNP10	1.22	0.033273	98	17	126.5	223.5	375.8	858

we observe a nice behaviour of the MDG original ranking, capturing the decreasing association order of the simulated SNPs. Instead, MDA ranking was not able to capture the order in the association effect.

Finally, Table 3.3 shows, for each SNP, its probability of being in the top-k list, with $k = 10$, 20 and 50. MDG picks-up the first 4 SNPS in the top-10 list and the first 7 SNPs in the top-20 list in approximated 90% of cases. MDA is only able to select SNP1 with this high probability in the top-10 list and the first 3 SNPs in the top-20 list. MDG puts almost always the first 8 SNPs in the top-50 list while MDA only picks-up the first 4 SNPs in the top-50 list with probability near 1. Neither MDG nor MDA are able to pick-up SNP9 and SNP10.

In addition to stability problems, there are other aspects that can affect the good

Table 3.2: *MDA ranking results for the* 10 *associated SNPs.*

			MDA original	MDA Jackknife rankings summary				
SNP	OR	P-value	ranking	Min	p25	p50	p75	Max
SNP1	3.06	1.06E-05	1	1	1	1	2	26
SNP2	2.69	4.25E-05	5	1	2	5	10	49
SNP3	2.54	6.12E-05	2	1	3	5	10	40
SNP4	2.32	0.000115	16	1	4	8	14	112
SNP5	2.16	0.000195	8	1	7.75	12	22	774
SNP6	1.94	0.000439	3	2	13.75	22.5	62.25	910
SNP7	1.72	0.000570	12	2	10.75	21.5	64	975
SNP8	1.53	0.001565	649	5	28.5	122.5	507.2	988
SNP9	1.31	0.005942	49	11	88.75	270.5	662.5	998
SNP10	1.22	0.033273	738	23	236	445.5	786.8	996

performance of a ranking procedure. Strobl et al. (2007) discuss different aspects, such as correlation between the predictor variables and the scale or the number of categories of these predictors, that can induce bias in the Random Forest importance measures. In absence of these sources of bias, the two specific examples, although not pretending to be representative of the wide range of possible situations, clearly illustrate that the stability of rankings is an important issue that should be routinely explored and that the ranks based on the Mean Decrease Gini provide more robust results.

Table 3.3: *MDG and MDA rankings. Probabilities of being in the top-*10*,* -20 *or* -50 *list.*

SNP	MDG.P10	MDG.P20	MDG.P50	MDA.P10	MDA.P20	MDA.P50
SNP1	0.99	1	1	0.98	0.99	1
SNP2	0.95	0.99	1	0.77	0.94	1
SNP3	0.97	1	1	0.79	0.97	1
SNP4	0.9	0.99	1	0.65	0.83	0.96
SNP5	0.87	0.95	1	0.43	0.72	0.84
SNP6	0.53	0.85	1	0.18	0.42	0.72
SNP7	0.76	0.94	1	0.25	0.49	0.72
SNP8	0.3	0.65	0.96	0.05	0.16	0.37
SNP9	0.01	0.09	0.65	0	0.05	0.17
SNP10	0	0.01	0.06	0	0	0.06

CHAPTER 4

Robust feature selection with Random Forest

In this chapter we propose a new strategy for variable selection using Random Forest. The proposed algorithm, namely AUC-RF, computes the Receiver Operating Curve (ROC) associated to the Random Forest, uses the area-under-the ROC curve (AUC) as the predictive accuracy of the Random Forest and implements a backward elimination process for selecting the set of variables with the highest AUC value. The goal of this chapter is two-fold: establish AUC as a preferable accuracy measure for Random Forests in front of the usual classification error rate and to provide a new selection algorithm based on the AUC. In particular, we show that the use of the classification error is especially inappropriate when dealing with unbalanced data sets. The new AUC-RF procedure for variable selection is illustrated with data from the Spanish Bladder Cancer/EPICURO Study (described in Chapter 1) and also with simulated data. The content of this chapter has been published in the Human Heredity journal (Calle et al., 2011) and the algorithm is publicly available as an R package, named AUCRF, at `http://cran.r-project.org/web/packages/AUCRF` and the package documentation can be found in appendix B.

4.1 Introduction

Genomic profiling, the use of genetic variants at multiple loci simultaneously for prediction of disease risk, requires the selection of the set of genetic variants that best predicts disease status. The goal of this chapter is to provide a new selection algorithm for genomic profiling.

We propose a new algorithm for genomic profiling based on optimizing the area-under-the ROC curve (AUC) of the Random Forest. The proposed strategy implements a backward elimination process based on the initial ranking of variables. We demonstrate the advantage of using the AUC instead of the classification error as a measure of predictive accuracy of the Random Forest. In particular, we show that the use of the classification error is especially inappropriate when dealing with unbalanced datasets. The new procedure for variable selection and prediction, namely AUC-RF, is illustrated with data from a bladder cancer study and also with simulated data.

Díaz-Uriarte and Alvarez de Andrés (2006) proposed a backward elimination method for obtaining the optimal subset of variables providing the lowest overall classification error, implemented in the R package `varSelRF` (Diaz-Uriarte, 2007). This method is based on two default strategies of the Random Forest, the most voted class for prediction and the out-of-bag classification error rate (OOB-ER) for accuracy. As mentioned before, for unbalanced data sets, the most voted class strategy tends to classify almost all individuals in the largest class and the classification error rate does not distinguish between false positives (FP) and false negatives (FN) which can give a false impression of accuracy (for instance, in a data set with 80% of controls and 20% of cases, a method that correctly classifies all controls but classifies cases randomly will have a classification error rate of only 10%). We prove that the proposed new algorithm overcomes these limitations.

In the context of a genetic study, the AUC-RF algorithm can be used for genomic profiling, that is, for identifying the set of variants with the highest combined predictive value of individual risk of disease. Our goal was to use this methodology for exploring the contribution of the inflammation pathway on bladder carcinogenesis through the Spanish Bladder Cancer Study data. Stratified analysis by tobacco smoking risk group was also of interest. While the whole sample was approximately balanced, the stratified samples were strongly unbalanced and, in this case, the use of Random Forest for variable selection based on the classification error rate was not satisfactory. This was the motivation of the new proposed strategy for variable selection using Random Forest. A simulation analysis has been conducted to evaluate the effectiveness of the AUC-RF method for selecting causative SNPs. Balanced and unbalanced scenarios have been simulated considering different numbers of causal SNPs, relative risks, minor allele frequencies (MAF) and disease prevalence. Though RF can be used for both discrete and continuous dependent variables we will explain the approach for a binary dependent variable representing disease status in the context of a case-control association study.

4.2 Methods

4.2.1 AUC-RF for feature selection

The AUC-RF algorithm iteratively fits Random Forests and eliminates a proportion of variables. Denote by $D = (Y; X)$ the $n \times (k + 1)$ matrix corresponding to the data set of interest where n is the number of individuals, Y is the binary dependent variable and $X = (X_1, \ldots, X_k)$ is the $n \times k$ matrix containing the predictor variables. The AUC-RF algorithm is described below:

1. Iterative process:

   First step:

   - Denote by X^1 the initial set of predictor variables ($X^1 = X$) and by $D^1 = (Y;\, X^1) = D$ the initial data set.
   - Build a Random Forest RF^1 on D^1, that is, using all predictor variables and the response (see subsection Random Forest parameters below)
   - Obtain the ranking of the predictor variables using the chosen measure of importance (see subsection Random Forest importance measures for details). Denote by $r = (r_1, \ldots, r_k)$ the ranking vector of variables $(X_1, \ldots, X_k)$.
   - Compute the out-of-bag AUC (OOB-AUC) of the Random Forest RF^1, namely OOB-AUC^1 (see subsection Random Forest prediction and AUC computation for details).

   Subsequent steps. Step j, $j > 1$:

   - Based on the initial ranking r, remove a fraction (by default 20%) of the less important variables from X^{j-1} and denote the resulting matrix of predictors as X^j.
   - Denote by D^j the reduced data set: $D^j = (Y;\, X^j)$
   - Build a Random Forest on D^j, namely RF^j.
   - Compute the OOB-AUC of the Random Forest RF^j , namely OOB-AUC^j.

   Repeat step j until

   - the number of remaining variables is less than $k0$ (by default $k0 = 1$).

2. Elimination process representation:

   The elimination process is visualized with a curve describing the OOB-AUC value, OOB-AUC^j, of Random Forest RF^j $(y - axis)$ as a function of the number of predictor variables $(x - axis)$.

3. Optimal set of predictors:

   The optimal set of predictor variables is the one giving rise to the Random Forest with the highest OOB-AUC, denoted by OOB-AUC$_{opt}$. The number of selected predictors is denoted by K_{opt}.

4. Predictive accuracy and probability of selection:

   The obtained OOB-AUC$_{opt}$ value cannot be considered as the genuine predictive accuracy of the selected variables on a new data set. It is inflated by the fact that it is measured on the same training data set that has been used for the selection process. A correction of this overoptimism is required. Also of special concern is the robustness of the rankings and, consequently, of the selected variables (Boulesteix and Slawski, 2009). AUC-RF deals with these two important issues by performing a repeated cross-validation analysis. The results of this analysis provide a corrected estimation of the predictive accuracy of the selected variables and an estimate of the probability of selection for each variable (Pepe et al., 2003). For ease of notation we illustrate this process in the case of a 5-fold-cross-validation process that is repeated 20 times.

   - For $m = 1, \ldots, M = 20$ repeat a 5-fold-cv process consisting of the following steps:
     (a) Divide the original data set into 5 subsets: $D_1^m, \ldots, D_5^m$,
     (b) For $j = 1, \ldots, J = 5$
         – Perform the AUC-RF feature selection on the learning data set, $L^{m,j} = D - D^{m,j}$.

– Let $RF_{opt}^{m,j}$ denote the optimal Random Forest (after feature elimination) and $S_{opt}^{m,j}$ the set of selected variables.

– Use $RF_{opt}^{m,j}$ to predict individuals in the test data set $D^{m,j}$ (See Subsection Random Forest prediction and AUC computation). This provides a vector of probabilities, $Pred^{m,j}$, corresponding to the proportion of trees yielding $Y = 1$.

(c) Join the predictions of the 5 cv subsets, $(Pred^{m,1}, \ldots, Pred^{m,5})$, and compute the AUC of these predictions, denoted by cv-AUC^m.

- Compute the mean cv-AUC $= \frac{1}{M} \sum_{m=1}^{M=20}$ cv-AUC^m .

- For each variable X_i, $i = 1, \ldots, k$, we compute its probability of selection as the proportion of times that it has been selected by the AUC-RF method:

$$P(X_i) = \frac{1}{M * J} \sum_{j=1}^{j=5} \sum_{M=20}^{m=1} 1(X_i \in S_{opt}^{m,j})$$

4.2.2 Random Forest parameters

AUC-RF uses Random Forest with the default parameters of the R-package randomForest available at `http://cran.r-project.org/web/packages/randomForest`. The most relevant specifications are `ntree=500` (the number of trees in a forest is 500), `mtry=`$\sqrt{n}$ (the number of selected candidate variables in each node is the squared of the total number of variables) and `replace=TRUE`, `nodesize=1`, `maxnodes=NULL`, `importance=FALSE`, `norm.votes=TRUE` (see the randomForest documentation for details). These default values can be modified when randomForest function is called.

4.2.3 Random Forest importance measures

Random Forest, as implemented in the R-package randomForest, provides two different importance measures, mean decrease accuracy (MDA) and mean decrease Gini (MDG). MDA quantifies the importance of a variable by measuring the change in prediction accuracy when the values of the variable are randomly permuted compared to the original observations. MDG is the sum of all decreases in Gini impurity due to a given variable (when this variable is used to form a split in the Random Forest), normalized by the number of trees. Strobl et al. (Strobl et al., 2008, 2007) studied different mechanisms that can induce bias in the Random Forest importance measures and Calle et al. (Calle et al., 2010) explored stability of these measures, as described in chapter 3. On the one hand, both the MDG and MDA importance measures may be biased in the case of variables with different scales or in the case of categorical variables with different numbers of categories (Strobl et al., 2007), but in the context of SNP data analysis (almost) all variables are three-categorical. On the other hand, in terms of robustness, the ranks based on the MDA provide very unstable results (Calle et al., 2010). More research is needed to elucidate the respective advantages and inconveniences of MDG and MDA in general. However, in the considered context, our preliminary study has clearly shown that MDA performs consistently and substantially worse than MDG, probably because of its high instability. We will thus use MDG it in this chapter for both the bladder cancer analysis and the simulation study.

4.2.4 Random Forest prediction and AUC computation

The individual class prediction using Random Forest is based on what are called the votes. The principle is that each tree *votes* for a class and that the predicted class of an individual is finally the class with more votes. However, the voting

procedure differs depending whether one wants to compute the so-called Out-Of-Bag Error Rate (OOB-ER) or rather make predictions for new individuals from a test data set. If the goal is to compute the OOB-ER, those trees for which an individual was out-of-bag (i.e. was not used to build the tree) contribute with a vote to the predictive class for this individual. For dichotomous class prediction ($Y = 0,\ Y = 1$), the votes are two variables ($v0,\ v1$), where $v0$ is the number of votes for class $Y = 0$ and $v1$ is the number of votes for class $Y = 1$. The total number $v0 + v1$ is the number of trees for which the individual was out-of-bag: approximately a third of the total number of trees when `replace=TRUE` is used. The OOB-ER is then defined as the proportion of individuals with predicted class different from the true class. If the goal is to predict new individuals from a test data set the procedure is similar but in this case all trees in the RF contribute with a vote ($v0 + v1 =$ `ntree`), since the individual was never used to build the trees. The default prediction procedure is to predict the most voted class and to provide the OOB-ER. Alternatively, AUC-RF explores the predictive accuracy of the Random Forest through its ROC curve and the corresponding AUC (area under the ROC curve) (Pepe, 2003). The AUC-RF procedure computes the AUC based on out-of-bag predictions, similarly to the OOB-ER, hence the notation OOB-AUC. Each individual is characterized by the numbers $v0$ or $v1$ of trees predicting $Y = 0$ and $Y = 1$, respectively. The ROC curve plots sensitivity against 1-specificity and can be obtained by varying the cutoff, c, in the prediction procedure based on the votes. The randomForest package allows to specify the cutoff as a vector $(1 - c,\ c)$ and then predicts an individual as $\hat{Y} = 1$ if $v_0 \cdot c < v_1 \cdot (1 - c)$. The most voted class strategy corresponds to $c = 0.5$. The OOB-AUC can be calculated directly from the mean rank of the cases, denoted by $\overline{r}_1$, as $\text{AUC} = \frac{1}{n_0}(\overline{r}_1 - \frac{n_1}{2} - \frac{1}{2})$ where n_1 and n_0 are the number of cases and controls, respectively, and the ranks are based on the proportion of trees yielding $Y = 1$, that is, $v1/(v0 + v1)$ (Wray et al., 2010).

4.3 Application

4.3.1 The Spanish Bladder Cancer/EPICURO Study

Here we center our attention on the analysis of the joint effect of multiple genes in the inflammation pathway on bladder carcinogenesis for which information on 282 SNPs genotyped in a total of 108 genes in this pathway is available. After excluding patients with more than 20% missing genotypes, the available sample for analysis consists of 1150 cases and 1149 controls. The remaining missing genotypes were imputed using function `rfimpute` provided in `randomForest` library. Smoking is the most important risk factor for bladder cancer and gene×smoking interactions has been reported (García-Closas et al., 2005; Samanic et al., 2006). For this reason, we were interested in performing a stratified analysis by tobacco smoking risk group (current smokers, former smokers and never smokers).

We use the never smokers group, an unbalanced data set consisting of 426 controls and 209 cases, for illustration of the proposed methodology, AUC-RF, and for comparison with `varSelRF` by Diaz-Uriarte. The backward elimination process performed by `varSelRF` algorithm is depicted in Figure 4.1. As it was anticipated, the use of the most voted classification strategy and the OOB-ER provides unsatisfactory results in the unbalanced non-smokers data set. The first RF, considering all variables, results in an OOB-ER = 0.32. Though, in some contexts, a predictive error of 32% could be acceptable, in this case, this value only reflects the proportion of cases in the sample, which are almost all incorrectly classified as controls. Indeed, all 426 controls are predicted as controls (classification error = 0) but only 12 out of the total 209 cases are classified as cases (classification error =0.94). A similar behaviour is observed for the subsequent RF built in the backward elimination process, providing always an OOB-ER around 0.3 and, consequently, an OOB-ER

curve almost flat which is not useful for identifying the optimal subset of predictors. In this case, varSelRF feature selection algorithm selects only 3 variables providing an OOB-ER = 0.31.

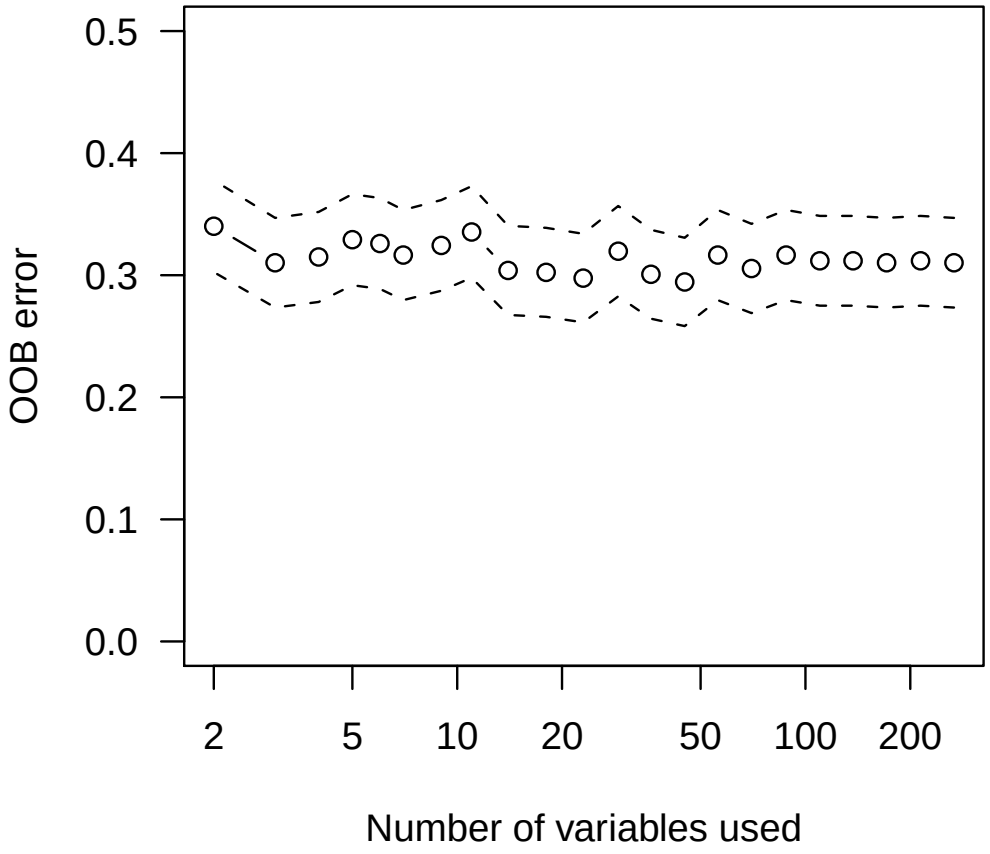

Figure 4.1: *varSelRF backward elimination procedure.*

The backward elimination process performed by the AUC-RF algorithm using the MDG importance measure can be visualized in Figure 4.2. The points in the curve correspond to the OOB-AUC of consecutive RF obtained with the remaining variables, after the less important variables were removed. The plot is built from the right (all variables) to the left (one variable). The optimal OOB-AUC is provided

by the top 43 more important variables, giving an OOB-AUC$_{opt}$ equal to 0.721. Correction for overfitting was performed with a 5-cv analysis and a cv-AUC = 0.56 was obtained.

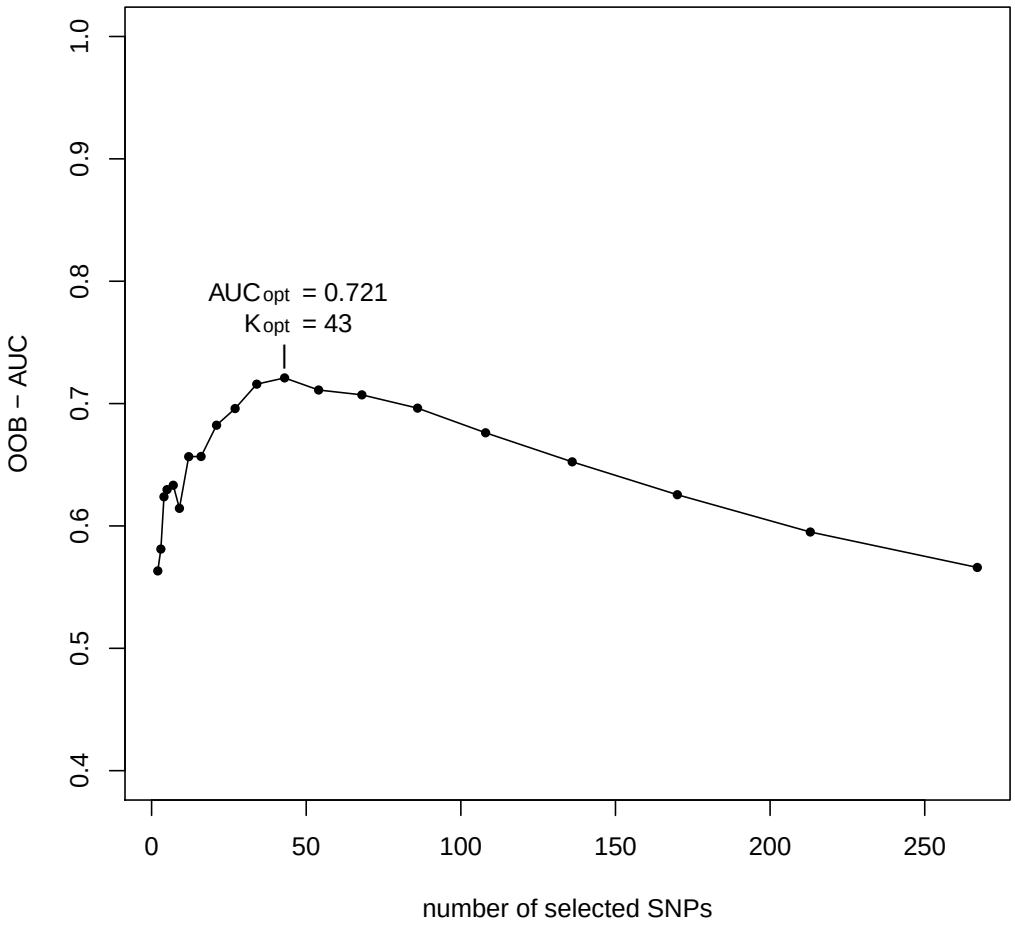

Figure 4.2: *AUC-RF backward elimination procedure.*

An important concern of selection methods, especially when they are based on rankings, is the robustness of the rankings and, consequently, of the selected set of SNPs. It is possible that different sets of variables provide practically the same predictive accuracy. For this reason, it is very important to provide the list of selected variables together with a measure of robustness of this selection. The AUC-RF algorithm im-

plements a repeated cross-validation process that provides the percentage of times that each variable has been selected. In this data set we repeated 20 times a 5-fold cv process. Table 4.1 provides the list of the most important SNPs that were selected by AUC-RF at least 70% of the times. We can see that the selection of this set of 18 SNPs is very robust, with the top two being selected almost every time.

Table 4.1: *SCBS. Most important SNPs, MDG index and probability of selection (P).*

NCBI gene reference sequence (SNP Id)	MDG	P
abca1.04	3.7387	1
masp1.53	2.6158	0.99
ephx2.04	2.6030	0.96
il10.17	2.1866	0.89
lta.04	2.0868	0.89
fcgr2a.01	2.3368	0.88
ptgs2.05	2.0644	0.85
ccr2.02	1.8030	0.80
csf1r.05	2.0657	0.79
mbl2.12	1.9348	0.78
gdf15.02	1.8225	0.77
alox5.10	1.6049	0.77
tlr2.04	1.7858	0.74
il4r.10	1.6247	0.74
cd86.02	1.6072	0.71
alox5.28	1.7345	0.70

4.3.2 Alzheimer's disease Study

The Alzheimer's disease study was introduced in chapter 1. A total of 682 SNPs in 32 genes of the reelin pathway are analyzed on a total of 1411 individuals (861 cases and 550 controls). Figures 4.3 to 4.5 show the AUC-RF output for the Alzheimer's disease data set including all the individuals and stratifying by APOE-ϵ4 carriers and noncarriers. For the case with no stratification, the optimal OOB-AUC is provided by the top 43 more important variables, giving an OOB-AUC$_{opt}$ equal to 0.629. For APOE-ϵ4 carriers strata, we obtain 34 variables and a value of 0.703 for OOB-AUC$_{opt}$, and also 34 variables and OOB-AUC$_{opt}$ = 0.6528 for APOE-ϵ4 noncarriers strata. However, when cross-validation analysis is performed the obtained cv-AUC values are close to 0.5 in the three cases which means that the SNPs have no predictive capacity on AD status.

4.3.3 Simulation Study

We performed a simulation study with the goal of investigating the performance of the proposed AUC-RF method for selecting variables with predictive capacity. We generated a set of k causal SNPs and $1000 - k$ non-causal SNPs. We followed the strategy described in chapter 3 for simulating the causal SNPs assuming independence and a multiplicative odds of risk model. All causal SNPs were assumed to be in HWE, to have the same effect size on the response and the same genotype frequencies. For each causal SNP we fixed the heterozygous relative risk (RR1) and assumed that the minor homozygous relative risk is $RR2 = (RR1)^2$. We investigated the role of disease prevalence ($p = 0.01,\ 0.1,\ 0.2,\ 0.3$), effect size ($RR1 = 1.1,\ 1.3,\ 1.5$), Minor Allele Frequencies ($MAF = 0.1,\ 0.2,\ 0.3$) on balanced ($N0$ = number of controls = $N1$ = number of cases = 2000) and unbalanced ($N0 = 3000$ and $N1 = 1000$) data sets. The number of causal SNPs was $k = 10,\ 50$

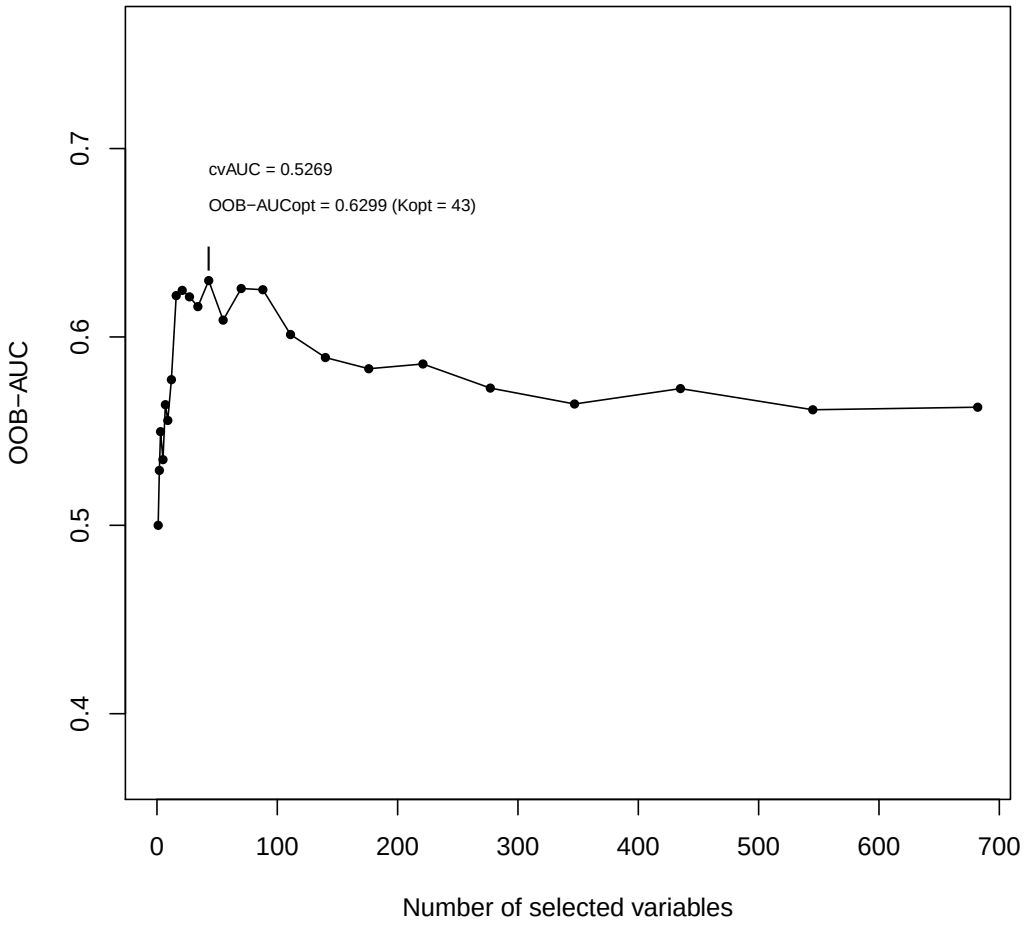

Figure 4.3: *AUC-RF backward elimination procedure for AD data set.*

and, for $RR1 = 1.1$, we further explored $k = 100$. This yield a total of 192 scenarios. For each scenario we generated two data sets, a learning data set (LD) and a test data set (TD) that was used for validation of the predictive accuracy of the selected set of SNPs. We performed the AUC-RF feature selection algorithm and kept the percentage (Pc) of causal SNPs that AUC-RF picks-up and the predictive accuracy of the selected set of SNPs on the test data set, denoted by test-AUC. This predictive accuracy depends on the ability of the algorithm to identify the causal SNPs but also on the predictive capacity of the causal SNPs. Thus, we also computed the

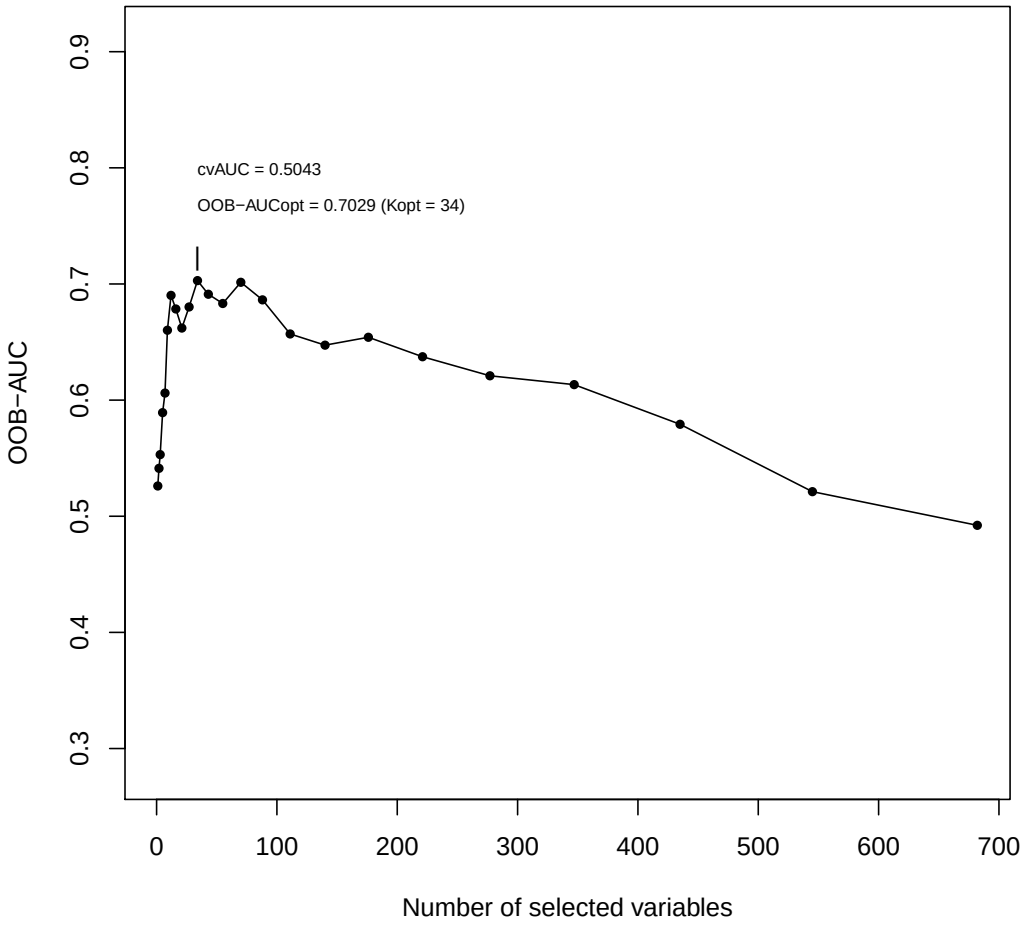

Figure 4.4: *AUC-RF backward elimination procedure for ApoE-ϵ4 carriers in AD data set.*

predictive ability of the causal SNPs as follows: Each individual is assigned a risk score given by

$$\sum_{j=1}^{k}(1\{G_j = 1\} \cdot \log OR1_j + 1\{G_j = 2\} \cdot \log OR2_j)$$

where $OR1$ is the odds-ratio of heterozygous vs major homozygous and $OR2$ is the odds-ratio of minor homozygous vs major homozygous. We computed the AUC of predictions based on the above risk score, denoted by score-AUC. The score-AUC

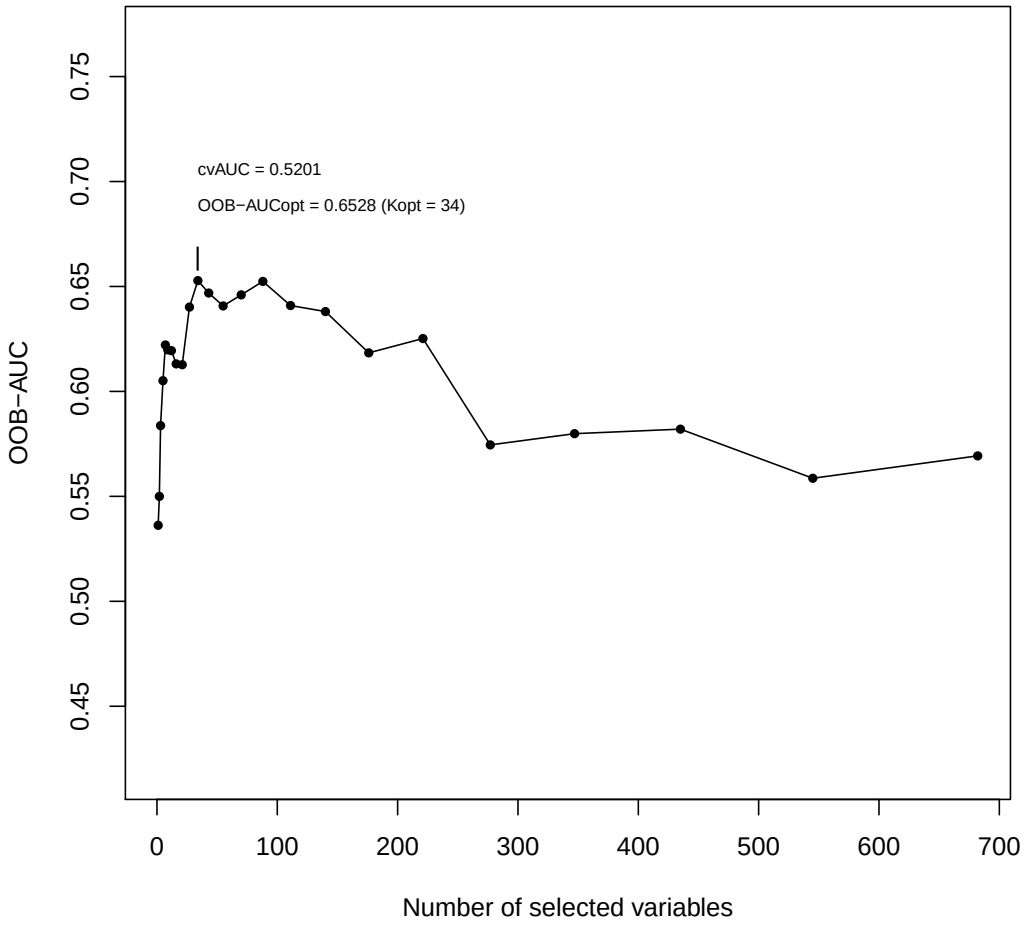

Figure 4.5: *AUC-RF backward elimination procedure for ApoE-ϵ4 noncarriers AD data set.*

can be seen as the best empirical predictive accuracy provided by the causal SNPs, if they were known, and will be used as a reference for interpreting the observed predictive accuracy of the AUC-RF method. We repeated this process 100 times for each scenario and averaged the results over the 100 replications.

We summarize the results of the simulation study in terms of the percentage of selected causal SNPs, denoted by Pc, in Tables 4.2 to 4.4 . The predictive accuracy

of the selected set of SNPs on the test data set, denoted test-AUC, is reported in Tables 4.5 to 4.7 . The score-AUC is also provided in parenthesis as a reference value of the maximum predictive accuracy of the causal SNPs. In order to visualize some of the obtained results, Figures 4.6 and 4.7 show the results for $k = 50$ and balanced data set. The percentage, Pc, of causal SNPs that AUC-RF is able to pick-up is mainly affected by the effect size, followed by the minor allele frequency and the disease prevalence (Figure 4.6 and Tables 4.2 to 4.4). For $RR1 = 1.5$, the percentage Pc is almost 100% in all cases, that is, all causal SNPs are identified. When $RR1$ reduces to 1.3 the efficacy remains for $MAF = 0.3$ and 0.2 but reduces considerably for $MAF = 0.1$. For $RR1 = 1.1$ the percentage of selected causal SNPs reduces drastically to 30-40% for $MAF = 0.3$, 10-20% for $MAF = 0.2$ and it is almost null for $MAF = 0.1$. A slight effect of the disease prevalence is observed in some situations: For $RR1 = 1.3$ and $MAF = 0.1$ and $RR1 = 1.1$ and $MAF = 0.3,\ 0.2$; the largest the prevalence, the higher the percentage Pc. A similar behaviour is observed in terms of predictive accuracy of the set of selected SNPs (Figure 4.7 and Tables 4.5 to 4.7). For $RR1 = 1.5$, the test-AUC is very high (around 0.8-0.9) which corresponds to very accurate predictions. Indeed, the obtained test-AUC after feature selection is very similar to the score-AUC provided by all causal SNPs (given in parenthesis). The effect of the genotype frequencies is observed, with $MAF = 0.3$ giving slightly better results than for $MAF = 0.2$ and better than for $MAF = 0.1$. Instead, the disease prevalence effect is not apparent in terms of test-AUC. For $RR1 = 1.3$ the predictive accuracy is around 0.7-0.8 when $MAF = 0.2,\ 0.3$ and around 0.65 when $MAF = 0.1$. The loss in predictive capacity (comparing the obtained test-AUC with the score-AUC) is more apparent for low values of MAF (around 7% when $MAF = 0.1$). In this case, the effect of disease prevalence is not apparent. For $RR1 = 1.1$ the predictive capacity of the selected set of SNPs is in general very low or non-existent. Note, however, that in this setting the score-AUC given by all causal SNPs is also very low. Only for $k = 100$ and $MAF = 0.3$ we obtain more acceptable predictive values, around 0.6. This is in

accordance with Janssens et al. (2007) who say that a genomic profile from a set of causal SNPs with such a weak marginal effect on the phenotype will require a larger number of SNPs to jointly get a useful predictive accuracy. Indeed, we can observe looking at Tables 4.5 to 4.7 that the larger the number k of causal SNPs the larger the AUC (both score-AUC and test-AUC) in all settings. Instead, this effect of the number of causal SNPs is not observed in the efficacy of the AUC-RF method for detecting causal SNPs (Tables 4.2 to 4.4). For instance, in Tables 4.5 when $MAF = 0.1$, the percentages Pc of identified causal SNPs are larger for $k = 10$ than for $k = 50$.

Table 4.2: *Percentage of selected causal SNPs (Pc) for* $RR1 = 1.5$.

		Balanced data set				Unbalanced data set			
Prevalence (p)		0.01	0.1	0.2	0.3	0.01	0.1	0.2	0.3
K=10									
	MAF=0.1	99.1	99.6	99.9	100	99.4	99.8	99.8	100
	MAF=0.2	100	100	100	100	100	100	100	100
	MAF=0.3	100	100	100	100	100	100	100	100
K=50									
	MAF=0.1	96.8	94.3	94.3	94.9	98.1	96.1	96.3	97
	MAF=0.2	100	99.8	99.4	99.3	100	99.7	99.5	99.3
	MAF=0.3	100	100	99.9	99.8	100	99.9	99.7	99.7

Table 4.3: *Percentage of selected causal SNPs (Pc) for* $RR1 = 1.3$.

		Balanced data set				Unbalanced data set			
Prevalence (p)		0.01	0.1	0.2	0.3	0.01	0.1	0.2	0.3
K=10									
	MAF=0.1	62.6	72	83.8	93.4	74.9	81.4	90.6	96
	MAF=0.2	96.8	98.5	99.9	99.9	98	98.4	99.6	99.9
	MAF=0.3	99.9	100	100	100	99.5	99.9	100	100
K=50									
	MAF=0.1	57.3	59.2	64.9	72.1	71.7	72.4	76	83.3
	MAF=0.2	94.2	92.9	92.7	94.3	96.5	94.8	94.8	96.5
	MAF=0.3	99.3	98.6	98.3	98.3	98.9	97.5	97.7	98.2

Table 4.4: *Percentage of selected causal SNPs (Pc) for* $RR1 = 1.1$.

		Balanced data set				Unbalanced data set			
Prevalence (p)		0.01	0.1	0.2	0.3	0.01	0.1	0.2	0.3
K=10									
	MAF=0.1	1.9	2.4	4.1	5.8	9.6	11.6	13.8	16.3
	MAF=0.2	14.8	18.8	21.8	27.7	31.5	31.6	40.4	50
	MAF=0.3	36.8	43.8	50.9	61.7	44.7	51.6	58.3	67.2
K=50									
	MAF=0.1	2.2	2.4	3.6	5	9.8	11.1	13	17
	MAF=0.2	13.4	16.3	20.2	27.3	30	34.1	37.7	45.3
	MAF=0.3	34.5	39.7	45.1	53.4	44	46.8	53.7	60.6
K=100									
	MAF=0.1	2	2.5	3.2	4.4	9.3	10.1	13.1	15.7
	MAF=0.2	13.6	15.5	18.5	22.4	29.3	31.9	36.5	40.7
	MAF=0.3	33.7	36.2	40.5	47	44.5	46.4	50.4	54.8

Table 4.5: *test-AUC of the selected SNPs and score-AUC (in parenthesis) of the causal SNPs for* $RR1 = 1.5$.

	Balanced data set				Unbalanced data set			
Prevalence (p)	0.01	0.1	0.2	0.3	0.01	0.1	0.2	0.3
K=10								
MAF=0.1	0.63 (0.66)	0.64 (0.67)	0.66 (0.69)	0.69 (0.71)	0.61 (0.66)	0.63 (0.67)	0.65 (0.69)	0.68 (0.71)
MAF=0.2	0.67 (0.7)	0.69 (0.72)	0.71 (0.74)	0.73 (0.76)	0.66 (0.7)	0.68 (0.72)	0.7 (0.74)	0.73 (0.76)
MAF=0.3	0.7 (0.73)	0.71 (0.74)	0.73 (0.76)	0.76 (0.78)	0.69 (0.73)	0.7 (0.74)	0.72 (0.76)	0.75 (0.78)
K=50								
MAF=0.1	0.78 (0.81)	0.77 (0.82)	0.78 (0.83)	0.79 (0.85)	0.77 (0.81)	0.76 (0.81)	0.77 (0.83)	0.79 (0.85)
MAF=0.2	0.85 (0.88)	0.84 (0.87)	0.84 (0.88)	0.85 (0.89)	0.84 (0.87)	0.83 (0.87)	0.83 (0.88)	0.84 (0.89)
MAF=0.3	0.88 (0.9)	0.86 (0.89)	0.86 (0.9)	0.87 (0.91)	0.87 (0.9)	0.85 (0.89)	0.85 (0.89)	0.86 (0.91)

Table 4.6: *test-AUC of the selected SNPs and score-AUC (in parenthesis) of the causal SNPs for* $RR1 = 1.3$.

	Balanced data set				Unbalanced data set			
Prevalence (p)	0.01	0.1	0.2	0.3	0.01	0.1	0.2	0.3
K=10								
MAF=0.1	0.54 (0.6)	0.56 (0.61)	0.58 (0.62)	0.6 (0.64)	0.54 (0.6)	0.55 (0.61)	0.57 (0.62)	0.59 (0.64)
MAF=0.2	0.59 (0.63)	0.6 (0.64)	0.62 (0.66)	0.64 (0.68)	0.58 (0.63)	0.59 (0.64)	0.61 (0.66)	0.63 (0.68)
MAF=0.3	0.61 (0.65)	0.62 (0.66)	0.64 (0.68)	0.66 (0.7)	0.6 (0.65)	0.61 (0.66)	0.63 (0.68)	0.65 (0.7)
K=50								
MAF=0.1	0.63 (0.71)	0.64 (0.72)	0.66 (0.74)	0.69 (0.76)	0.64 (0.71)	0.64 (0.72)	0.66 (0.73)	0.69 (0.76)
MAF=0.2	0.73 (0.77)	0.73 (0.78)	0.75 (0.79)	0.76 (0.81)	0.72 (0.77)	0.72 (0.77)	0.74 (0.79)	0.76 (0.81)
MAF=0.3	0.76 (0.8)	0.77 (0.8)	0.78 (0.82)	0.79 (0.84)	0.76 (0.8)	0.75 (0.8)	0.76 (0.81)	0.78 (0.83)

Table 4.7: *test-AUC of the selected SNPs and score-AUC (in parenthesis) of the causal SNPs for* $RR1 = 1.1$.

	Balanced data set				Unbalanced data set			
Prevalence (p)	0.01	0.1	0.2	0.3	0.01	0.1	0.2	0.3
K=10								
MAF=0.1	0.5 (0.53)	0.5 (0.53)	0.5 (0.54)	0.5 (0.54)	0.5 (0.52)	0.5 (0.53)	0.5 (0.53)	0.5 (0.54)
MAF=0.2	0.5 (0.54)	0.51 (0.55)	0.51 (0.55)	0.51 (0.56)	0.5 (0.54)	0.51 (0.54)	0.51 (0.55)	0.52 (0.56)
MAF=0.3	0.51 (0.55)	0.51 (0.55)	0.52 (0.56)	0.52 (0.57)	0.51 (0.55)	0.51 (0.55)	0.51 (0.56)	0.52 (0.57)
K=50								
MAF=0.1	0.5 (0.56)	0.5 (0.57)	0.5 (0.58)	0.51 (0.59)	0.51 (0.55)	0.51 (0.56)	0.51 (0.57)	0.52 (0.58)
MAF=0.2	0.52 (0.58)	0.52 (0.6)	0.53 (0.61)	0.54 (0.63)	0.52 (0.58)	0.53 (0.59)	0.54 (0.6)	0.55 (0.62)
MAF=0.3	0.53 (0.6)	0.54 (0.61)	0.55 (0.63)	0.57 (0.65)	0.54 (0.59)	0.54 (0.61)	0.56 (0.62)	0.57 (0.64)
K=100								
MAF=0.1	0.51 (0.58)	0.51 (0.59)	0.51 (0.6)	0.51 (0.62)	0.51 (0.57)	0.52 (0.58)	0.52 (0.6)	0.53 (0.61)
MAF=0.2	0.53 (0.62)	0.53 (0.63)	0.54 (0.65)	0.56 (0.67)	0.54 (0.61)	0.55 (0.62)	0.56 (0.64)	0.57 (0.66)
MAF=0.3	0.56 (0.64)	0.57 (0.65)	0.58 (0.67)	0.6 (0.69)	0.57 (0.63)	0.57 (0.64)	0.59 (0.66)	0.61 (0.68)

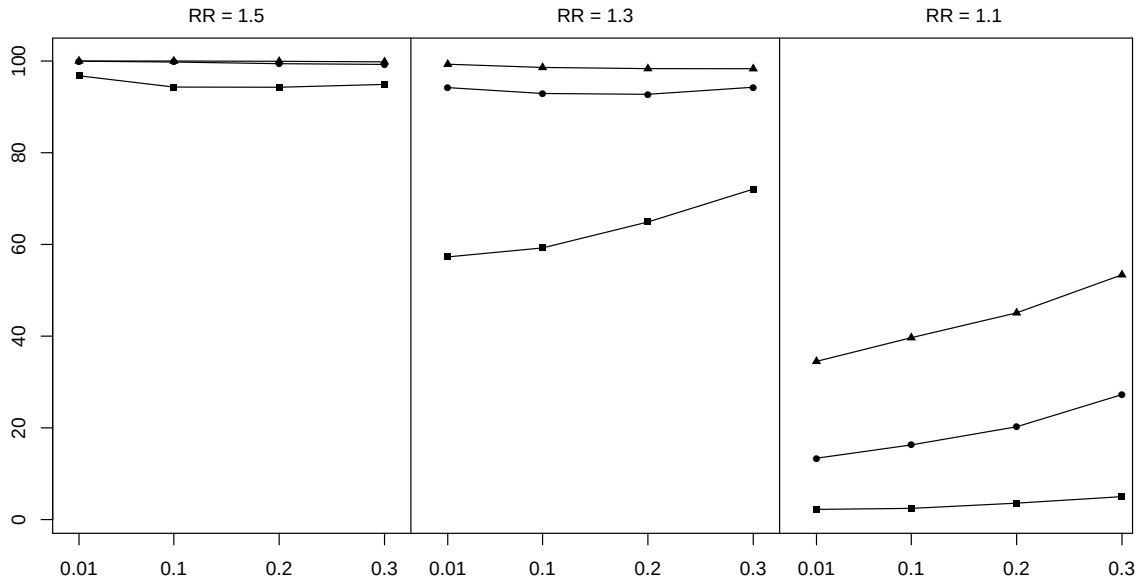

Figure 4.6: *Proportion of selected causal SNPs for* $k = 50$ *and balanced data sets.*

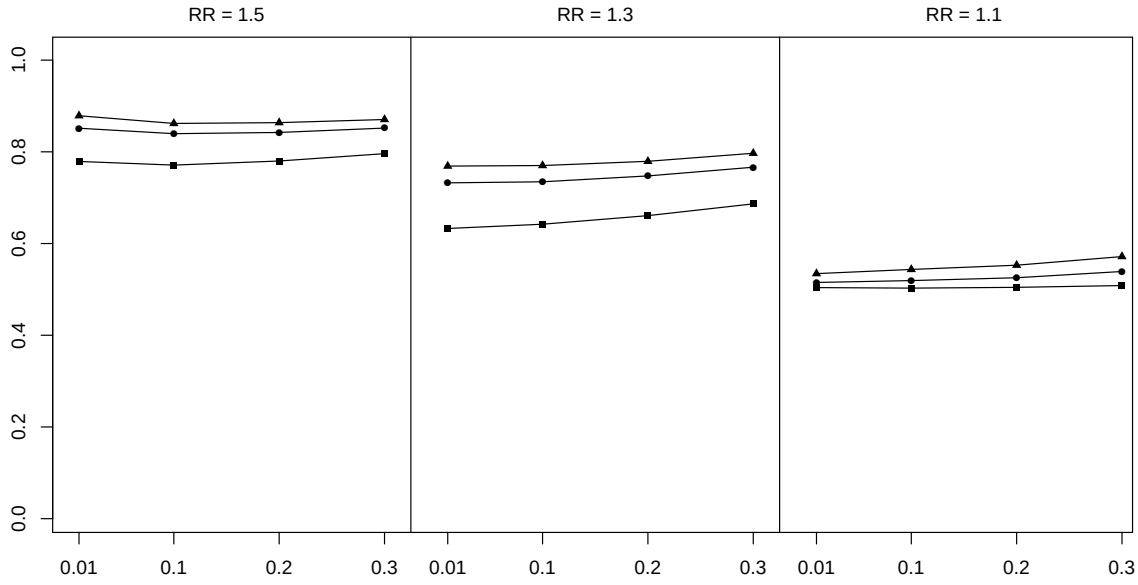

Figure 4.7: *AUC of the selected SNPs for* $k = 50$ *and balanced data sets.*

4.4 Discussion

In this work we proposed a new feature selection strategy consisting of a backward elimination process for optimization of the Random Forest AUC. The application of this method to a real bladder cancer study proves that the default RF most voted class prediction strategy together with the use of the classification error rate provides unsatisfactory results in unbalanced data sets. However, even for balanced data sets, the use of the AUC is preferable to the classification error rate because the error rate is dependent on the case/control rates in the sample which not necessarily represent the case/control rates in the population. The use of the AUC is especially appealing after the recent increasing interest on this measure in the molecular and genetic epidemiology field (Jakobsdottir et al., 2009; Janssens et al., 2006; Kraft et al., 2009; Lu and Elston, 2008; Lu et al., 2010; Moonesinghe et al., 2010; Pepe et al., 2004; Wray et al., 2010). The maximum value of the AUC of a genetic risk predictor model has been related to the heritability and prevalence of the disease.

In the proposed approach the same initial ranking is used for all iterations in the backward elimination process. Jiang et al. proposed a backward elimination strategy for variable selection using RF similar to Diaz Uriarte's method, the main difference being the recomputation of the ranking of the remaining variables at each step of the elimination process (Jiang et al., 2004). In our opinion a potential drawback of this strategy is that it might accentuate the overfitting problem already present in any elimination process, giving rise to an increase of overfitting.

CHAPTER 5

Identification of high-order genetic interactions using the likelihood-ratio score

In this chapter we consider the use of the likelihood ratio score for exploring high-order interactions. We propose a forward selection method for obtaining sets of genetic variants (SNPs) with optimal prediction accuracy. The proposed algorithm, namely Optimal AUC, is computationally feasible for exploring higher order interactions, not only second or third order interactions, in a large number of variables setting. The new procedure for epistasis identification is illustrated with data from the Alzheimer Disease Association Study (described in Chapter 1) and also with simulated data.

5.1 Introduction

Many statistical methods for epistasis analysis have been proposed, from exhaustive searches using regression models including interactions (Marchini et al., 2005) to data-mining methods such as the Model-Based Multifactor Dimensionality Reduction method (Calle et al., 2010) that is an extension of the popular MDR method (Ritchie et al., 2003) but allows adjusting for marginal effects and confounders. An overview on this topic is given by Van Steen (2012). These methods are usually able to scan for second order interactions but, since they are very computationally demanding, they become infeasible for exploring higher order interactions when the number of variables to explore is large, as it is usually the case.

In this chapter we propose a new strategy for exploring the joint predictive effect of a set of genetic or environmental factors, including their possible interactions. The proposed method, referred to as the "Optimal AUC algorithm", is based on the likelihood ratio score. Given a set of predictors, the likelihood ratio score provides an optimal prediction of a binary variable in the sense that it provides the largest discrimination accuracy among all possible risk scores derived from the set of predictors (McIntosh and Pepe, 2002).

The Optimal AUC algorithm follows a process of forward selection to obtain the subset of factors with the highest joint predictive accuracy. The algorithm is computationally feasible for exploring higher order interactions, not only second or third order interactions, in a large number of variables context.

5.2 Optimal prediction and optimal ROC curve

In this section we address the question of how to combine multiple predictors to obtain the best possible prediction of disease risk. We focus in case-control genetic association studies where the outcome of interest is a binary variable Y that informs of the presence ($Y = 1$) or absence ($Y = 0$) of a disease and the predictors are $G = (G_1, \ldots, G_m)$, a set of m genetic markers that have been genotyped on each subject. We assume that the genetic markers are single-nucleotide polymorphisms (SNP). The observed data is of the form $(Y^i, G_1^i, \ldots, G_m^i), i = 1, \ldots, n$, where n is the number of individuals, Y^i is disease status for individual i and G_k^i is the genotype (0, 1 or 2) of individual i for marker k.

Our goal is to predict $P(Y^i = 1 \mid G_1^i, \ldots, G_m^i)$, the probability of being diseased based on the genetic profiles of each individual. A relevant question is how to choose the best combination of genetic markers in order to optimize the prediction accuracy. There are several measures of prediction accuracy that can be used. Here we focus on the area under the receiver operating characteristic curve (ROC curve), that is denoted by AUC.

The prediction of a dichotomous variable given a set of variables is usually based on what is known as the risk score. In our setting, the risk score given the genetic markers is a real valued function $S = S(G_1, \ldots, G_m)$ so that larger values of S are associated to higher disease risk. Predictions based on S are of the form:

$$\hat{Y}^i = \begin{cases} 1 \; if \; S^i \geq c \\ 0 \; if \; S^i < c \end{cases}$$

where c is a threshold parameter.

For every possible value of c we can define the true positive rate $TP(c)$ as the probability of a diseased individual to be correctly predicted as diseased, $TP(c) =$

$P(S > c \mid Y = 1)$, and the false positive rate $FP(c)$ as the probability of a non diseased individuals to be incorrectly predicted as diseased, $FP(c) = P(S > c \mid Y = 0)$. The ROC curve is used for exploring the prediction accuracy of a score function S for the different possible cut-offs. It consists of a plot of the true positive rate (y axis) against the false positive rate (x axis), or, equivalently, sensitivity vs (1−specificity). More formally, the ROC curve is defined as the set of points $\{(FP(c), TP(c)),\ c \in (-\infty, \infty)\}$. The ROC curve is a monotone increasing function from $(0, 0)$ to $(1, 1)$. A risk score S with a ROC curve that is closer to $(0, 1)$ represents a very accurate prediction model.

The area under the ROC curve (AUC) provides a measure of discrimination of the risk score among diseased and non-diseased individuals. AUC takes values in the range $(0.5, 1)$, where 1 represents perfect discrimination for some threshold c while 0.5 indicates that the risk score S has no discrimination capacity. It can be shown that the AUC is equal to the probability that a diseased individual has a score larger than a non-diseased: $AUC = P\left[S(Y = 1) > S(Y = 0)\right]$. In Figure 5.1 two ROC curves have been plotted. The black curve has an AUC of 0.9293 that describes an almost perfect discrimination. The grey curve has an AUC of 0.6899 corresponding to a moderate capacity of discrimination.

The logistic regression model, defined as a linear model for the log-odds of disease, is the most popular method for combining multiple predictors when the output is a binary variable:

$$log\left(P(Y = 1)/\left(1 - P(Y = 1)\right)\right) = \beta_0 + \beta_1 G_1 + \cdots + \beta_m G_m \ .$$

In this model the risk score is the linear part of the model $S = \beta_0 + \beta_1 G_1 + \cdots + \beta_m G_m$. However, the logistic regression is not necessarily the best predictive model. Pepe (2003) proved that the optimal risk score for predicting a binary variable Y given $G = (G_1, \ldots, G_m)$ is the likelihood ratio score:

$$LR(G) = LR(G_1, \ldots, G_m) = \frac{P(G_1, \ldots, G_m \mid Y = 1)}{P(G_1, \ldots, G_m \mid Y = 0)} \ .$$

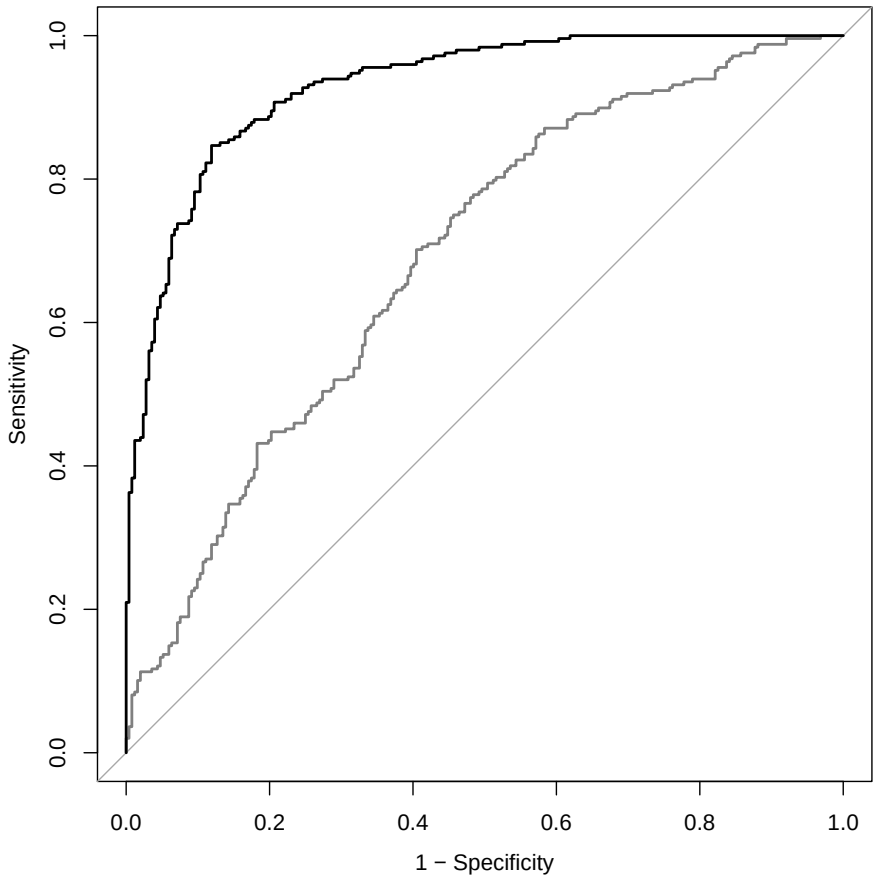

Figure 5.1: *Example of ROC curves.*

Proposition 1. *The likelihood ratio score is the best test for discrimination between cases and controls based on* $(G_1, \ldots, G_m)$*. The ROC curve for* $LR(G)$ *is uniformly above any other ROC curves based on* $(G_1, \ldots, G_m)$*. Thus,* $AUC_{LR} \geq AUC_S$ *for any risk score* S *based on* $(G_1, \ldots, G_m)$*. The ROC curve for the likelihood ratio score,* $LR(G)$*, receives the name of optimal ROC curve and* $AUC_{opt}\{G_1, \ldots, G_m\}$ *denotes the AUC of the optimal ROC curve and is referred to as the optimal AUC.*

Graphical proof

Let's sketch a graphical proof of the above result, that the ROC curve of the likelihood ratio score of a set of variables is always above any other ROC curve based on the same variables.

In our case, where the predictors are categorical variables with three categories, a risk score S based on $G = (G_1, \ldots, G_m)$ is a discrete function with $K = 3^m$ different categories corresponding to the total number of different multilocus genotypes. We denote by G^k (with an upper index) for $k = 1, \ldots, K = 3^m$ the different possible realizations of G. The ROC curve is then obtained by ranking the multilocus genotypes according to S in decreasing order, $S(G^{(1)}) \geq S(G^{(2)}) \geq \cdots \geq S(G^{(K)})$, where $G^{(k)}$ denotes the multilocus genotype with rank k for $k = 1, \ldots, K = 3^m$, and then by computing the true positive and false positive rates. For a threshold c such that $S(G^{(k)}) \geq c \geq S(G^{(k+1)})$ and $k = 1, \ldots, K = 3^m$:

$$\begin{cases} TP(c) = P(S > c \mid Y = 1) = \sum_{j=1}^{k} P(G^{(j)} \mid Y = 1) \\ FP(c) = P(S > c \mid Y = 0) = \sum_{j=1}^{k} P(G^{(j)} \mid Y = 0) \end{cases}$$

Graphically this means that the ROC curve is built by adding consecutive triangles with basis $P(G^{(j)} \mid Y = 0)$ and height $P(G^{(j)} \mid Y = 1)$ for $j = 1, \ldots, K = 3^m$. The slope of the j-th triangle is $P(G^{(j)} \mid Y = 1)/P(G^{(j)} \mid Y = 0)$ which is equal to $LR(G^{(j)})$, the likelihood ratio of $G^{(j)}$. When we use the likelihood ratio score, the multilocus genotypes are ranked according to a decreasing likelihood ratio: $LR(G^{(1)}) \geq LR(G^{(2)}) \geq \cdots \geq LR(G^{(K)})$. Thus, when we built the ROC curve of LR we start with the triangle with the largest slope, then the second triangle with the second largest slope and so on, yielding a concave curve that is always above any other possible ROC curve based on G.

Example with two genetic markers

Two genetic markers (SNPs), G_1 and G_2, define a partition of the sample space into 9 categories, $G^j,\ j = 1, \ldots, 9$. The likelihood ratio for each category j can be estimated as the proportion of cases divided by the proportion of controls in the category:

$$\hat{LR}(G^j) = \frac{\left(\sum_{i=1}^{n} 1\left\{(G_1^i, G_2^i) = G^j, Y^i = 1\right\}\right)/n_1}{\left(\sum_{i=1}^{n} 1\left\{(G_1^i, G_2^i) = G^j,\ Y^i = 0\right\}\right)/n_0}$$

where $n_1 = \sum_{i=1}^{n} 1\left\{Y^i = 1\right\}$ is the number of cases and $n_0 = \sum_{i=1}^{n} 1\left\{Y^i = 0\right\}$ is the number of controls. The different categories are ordered according to the likelihood ratio score and the ROC curve is obtained as described before.

To illustrate these concepts we consider a simulated case-control study with 1600 subjects (800 cases and 800 controls) and 1000 SNPs. The first two SNPs (SNP1 and SNP2) in this dataset are associated with the case-control variables through an interaction effect but without exhibiting marginal main effects. The remainder 998 variables are not associated with the phenotype. The dataset was downloaded from "Jason Moore Computational Genetics lab" at `http://discovery.dartmouth.edu/epistatic_data`. We will also use this dataset in the next section to illustrate the proposed "Optimal AUC algorithm".

Table 5.1 provides a summary of the first two SNPs. The first five columns of the table show each genotype category combination for the two SNPs, the number of cases and controls and their proportions in each category, the LR (cases and controls proportions ratio) for each category and their ranking based on the LR. The AUC calculated for the ROC curve based on LR is 0.865. For illustrative purposes, we also fitted a logistic lineal model with the two SNPs as predictors and their associated ROC curve. As expected, no significant terms were detected and the corresponding

ROC curve based on the regression model has an AUC of 0.49. Finally we also fitted the logistic regression model including the interaction terms for the two SNPs. In this case we obtained a significant model with the same ROC curve and AUC obtained for the model based on the LR. The three ROC curves are represented in Figure 5.2.

Table 5.1: *Optimal ROC partition example with two genetic markers.*

			Proportions		LR	Ranking
Genotype	Cases	Controls	cases ($p1$)	controls ($p0$)	($p1/p0$)	LR
$G^1 = (0,0)$	13	184	0.02	0.23	0.07	9
$G^2 = (0,1)$	197	82	0.25	0.10	2.40	4
$G^3 = (0,2)$	67	12	0.08	0.01	5.58	3
$G^4 = (1,0)$	261	25	0.33	0.03	10.44	1
$G^5 = (1,1)$	84	284	0.10	0.35	0.30	6
$G^6 = (1,2)$	52	83	0.06	0.10	0.63	5
$G^7 = (2,0)$	13	75	0.02	0.09	0.17	7
$G^8 = (2,1)$	109	16	0.14	0.02	6.81	2
$G^9 = (2,2)$	4	39	0.00	0.05	0.10	8
Total	800	800				

In this example the LR score is clearly better than the logistic regression model without interaction. However, there is no advantage of using the LR score over the logistic regression including an interaction since both models provide the same discrimination accuracy. This is the case not only in this example, in general, the LR score and the logistic regression with an interaction will provide the same classification. The advantage of using the LR score over the logistic regression will be

manifest when dealing with higher-order interactions. In logistic regression the interactions are specified parametrically. A second order interaction is specified through an additional parameter corresponding to the product of the two interacting variables. Third order interactions could also be specified parametrically in a logistic regression model, but this is not feasible for higher order interactions.

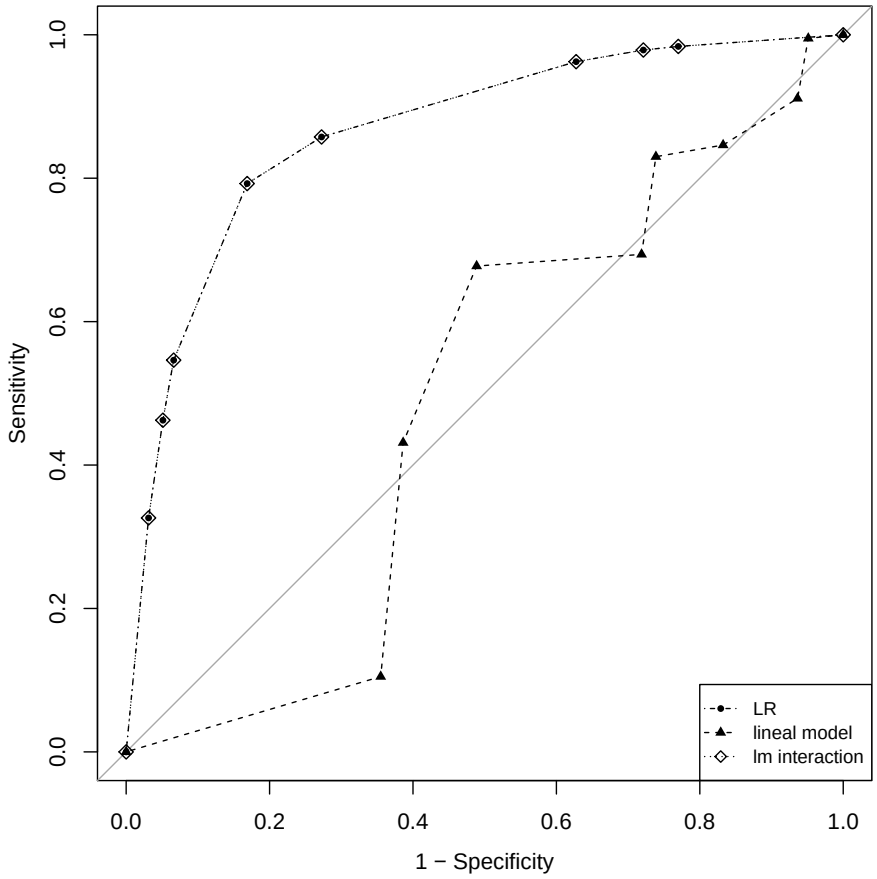

Figure 5.2: *Optimal ROC curve and ROC curves based on logistic regression for data in Table(5.1).*

5.3 Searching for higher-order interactions: Optimal AUC algorithm

In this section we propose a method, the Optimal AUC algorithm, that uses the likelihood ratio score and the optimal ROC curve for the identification of subsets of variables that best predicts disease risk.

Given a set of p genetic markers (SNPs), $G_1, \ldots, G_p$, our goal is to identify the subset of predictors with the highest discrimination accuracy between diseased and controls. We set $m \leq p$, the maximum size of the subsets to be explored.

The proposed algorithm follows a forward selection process where variables are successively included in the model according to a prespecified optimization criteria, that in this case is maximization of the optimal AUC (the AUC of the optimal ROC curve). Typically, forward selection algorithms start with no variable in the model and, in the first step, all variables are checked and the one that optimizes the selection criteria is selected as the first variable in the model. With this strategy the final model is strongly influenced by the selection of the first variable. This is not very convenient in contexts like ours where the SNPs have very small marginal effects. In order to reduce the influence of the first selected variable, we propose to perform p different forward selection processes, one for each variable $G_1, \ldots, G_p$ as the first variable in the model. At the end, we analyze the p resulting models and select the best one.

The Optimal AUC algorithm consists of the following steps that are briefly described here. Each step is explained in more detail in the following subsections:

Optimal AUC algorithm

Step 1: Forward selection process

Given the first SNP, $G_{(1)}$, we select the second SNP, $G_{(2)}$, so that

$$AUC_{opt}\{G_{(1)}, G_{(2)}\} = max_{\{j=1,\ldots,p\}} AUC_{opt}\{G_{(1)}, G_j\} \ .$$

Given a set of SNPs $G_{(1)}, \ldots, G_{(k-1)}$, we select $G_{(k)}$ so that

$$AUC_{opt}\{G_{(1)}, \ldots, G_{(k-1)}, G_{(k)}\} = max_{\{j=1,\ldots,p\}} AUC_{opt}\{G_{(1)}, \ldots, G_{(k-1)}, G_j\}$$

for $k = 2, \ldots, m$ where m is a fixed value (by default $m = 10$).

We repeat this process p times, taking each available SNP as the first variable in the forward selection process: $G_{(1)} = G_l$ for $l = 1, \ldots, p$.

We obtain p models of the form $M^l = \{G^l_{(1)} = G_l, G^l_{(2)}, \ldots, G^l_{(m)}\},\ l = 1, \ldots, p$.

Step 2: Selection of the best model and inference

In this step we select the best among the p models $M^l,\ l = 1, \ldots, p$, that we denote by $M^* = \{G^*_{(1)}, G^*_{(2)}, \ldots, G^*_{(m)}\}$, and explore its significance.

Step 3: Pruning

Since the size m of the models has been taken arbitrarily, the best model $M^* = \{G^*_{(1)}, G^*_{(2)}, \ldots, G^*_{(m)}\}$, may contain variables that are not really associated with the response variable Y and should be removed from the model. This step performs a pruning process where the last variables in the model are eliminated if they do not add significant prediction accuracy to the whole model.

Ye et al. (2011) proposed a forward selection method, called "forward ROC method", for identification of predictive genetic profiles on high-dimensional data including interactions. Both methods, the "forward ROC method" and the "Optimal AUC" proposed here were developed independently but essentially follow the same strategy: a forward selection process that maximizes the area under the optimal ROC curve. However, the implementation of both algorithms is very different and, consequently, both methods can yield to different models and conclusions. The main differences, as will be discussed next in more detail, are: the computation of the empirical optimal AUC, the missing data treatment, the selection of the best m dimensional model and the pruning process.

5.3.1 Forward selection process

As explained before, the forward selection process performs iteratively the following variable selection:

Given a set of SNPs $G_{(1)}, \ldots, G_{(k-1)}$, we select $G_{(k)}$ so that

$$AUC_{opt}\{G_{(1)}, \ldots, G_{(k-1)}, G_{(k)}\} = max_{\{j=1,\ldots,p\}} AUC_{opt}\{G_{(1)}, \ldots, G_{(k-1)}, G_j\}$$

for $k = 2, \ldots, m$

This process requires the computation of the optimal AUC that is defined as the area under the optimal ROC curve. The optimal ROC curve for k variables $G_1, \ldots, G_k$ is given by the likelihood ratio score:

$$LR(G_1, \ldots, G_k) = \frac{P(G_1, \ldots, G_k \mid Y = 1)}{P(G_1, \ldots, G_k \mid Y = 0)} \ .$$

In general the multivariate distribution of $G_1, \ldots, G_k$ for cases and controls is unknown but can be estimated nonparametrically with the empirical frequency. However, in order to reduce overfitting, we propose the following estimation:

$$\hat{P}\left(G_1 = g_1, \ldots, G_k = g_k | Y = 1\right) = \frac{\sum_{i=1}^{n} 1\left\{G_1^i = g_1, \ldots, G_k^i = g_k,\ Y^i = 1\right\}}{\sum_{i=1}^{n} 1\left\{Y^i = 1\right\}} + \delta$$

and

$$\hat{P}\left(G_1 = g_1, \ldots, G_k = g_k | Y = 0\right) = \frac{\sum_{i=1}^{n} 1\left\{G_1^i = g_1, \ldots, G_k^i = g_k,\ Y^i = 0\right\}}{\sum_{i=1}^{n} 1\left\{Y^i = 0\right\}} + \delta$$

where δ is a small value taken by default equal to 0.001. We discuss below the overfitting problem.

The process stops after the prespecified number m of SNPs have been selected.

We can represent the forward selection process with a plot, that we refer as the "forward selection curve", that represents the optimal AUC at each step versus the number of selected SNPs. Each curve has m dots with coordinates $(k,\ AUC_{opt}^k)$, $k = 1, \ldots, m$ where AUC_{opt}^k denotes the optimal AUC of the first k selected SNPs $(G_{(1)}, \ldots, G_{(k)})$.

To illustrate the forward selection process described above we consider again the example introduced in the previous section: a simulated case-control dataset with 800 cases and 800 controls and 1000 SNPs where the first two SNPs (SNP1 and SNP2) are associated with the outcome through an epistasis effect but without marginal main effects. The challenge is how to detect this second order interaction among all possible second or higher order interactions of the 1000 SNPs.

In Figure 5.3 we provide two plots. Both plots represent the selection process taking SNP1 as the first SNP. The plot on the left is obtained without overfitting correction ($\delta = 0$) while for the plot on the right we used the default overfitting correction ($\delta = 0.001$). The different dots of the selection curves provide the Optimal AUC in each step of the selection algorithm. In the first step, both plots provide an optimal AUC around 0.5 which reflects the null discrimination accuracy of SNP1 individually. In the second step of the selection process the algorithm considers all

the SNPs in combination of SNP1 and selects the one giving the largest prediction accuracy. In this case, the selected SNP was SNP2 which, in combination with SNP1, provides an optimal AUC of 0.86. The difference of correcting for overfitting or not is manifest after this second step. In the plot on the left, where no overfitting correction was applied, the Optimal AUC increases when a new SNP is added to the pair (SNP1, SNP2). This increase is only apparent since only SNP1 in combination with SNP2 are associated with the phenotype. When overfitting correction is applied, the Optimal AUC decreases when new SNPs are added and the two interacting SNPs can be identified. This example illustrates the need of correcting for overfitting in the selection process.

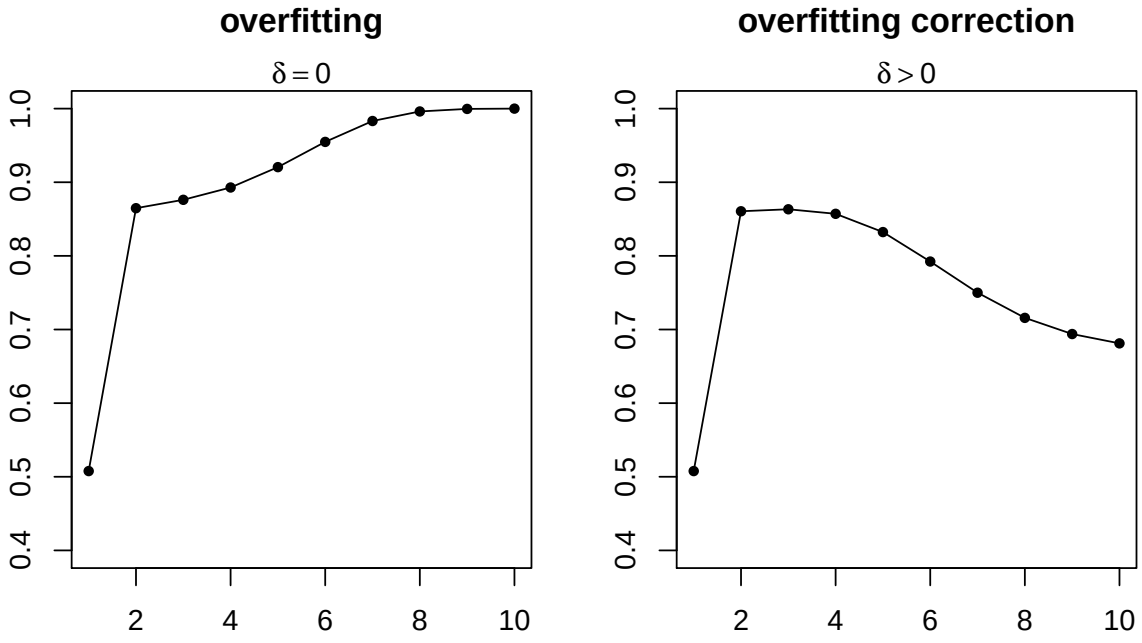

Figure 5.3: *a) Forward selection curve with $\delta = 0$ (no overfitting correction) b) Forward selection curve with $\delta = 0.001$ (with overfitting correction).*

5.3.2 Best model and inference

The selection process described before provides a total of p models (sets of m SNPs), one for each initial SNP. The selection of the best model will be based on the forward selection curves. In Figure 5.4 we see two forward selection curves corresponding to 2 different models. The upper curve (dots are circles) represents the selection process starting with SNP1 and the other curve (dots are triangles) represents the selection process for SNP474. It is clear that the preferable model is the one that achieves larger values. We consider the area under the forward selection curve as the numerical summary that indicates wether a model achieves large values of optimal AUC or not. The area under the forward selection curve is obtained as $W_j = \sum_{i=2}^{m} \frac{1}{2}(AUC_{opt}^{i} + AUC_{opt}^{i-1})$ for $j = 1, \ldots, p$.

We define the best model as the one with the largest area $W_{max} = max\{W_1, \ldots, W_p\}$. This criterion is inspired in the optimal AUC interpretation. The curves with largest areas will be the curves that reach larger values, in this case AUC_{opt} values, and grow fastest.

For the study of the statistical significance of results we need to obtain the empirical distribution of W_{max} under the null hypothesis of no association. In the absence of an analytical expression we could obtain the null distribution through a permutational approach but this would be computational unfeasible for large datasets. Instead, since W_{max} is the largest of a set of p areas W_i we can use the extreme value distribution. We assume that the areas, W_i, are independent and normal distributed under the null, $W_i \sim N(\mu, \sigma)$, $i = 1, \ldots, p$. In this setting the extreme value distribution is given by:

$$P(W_{max} \leq w) = \prod_{i=1}^{p} P(W_i \leq w) = \left[\Phi\left(\frac{w-\mu}{\sigma}\right)\right]^p$$

where Φ indicates the standard normal distribution. Then, for a significance level

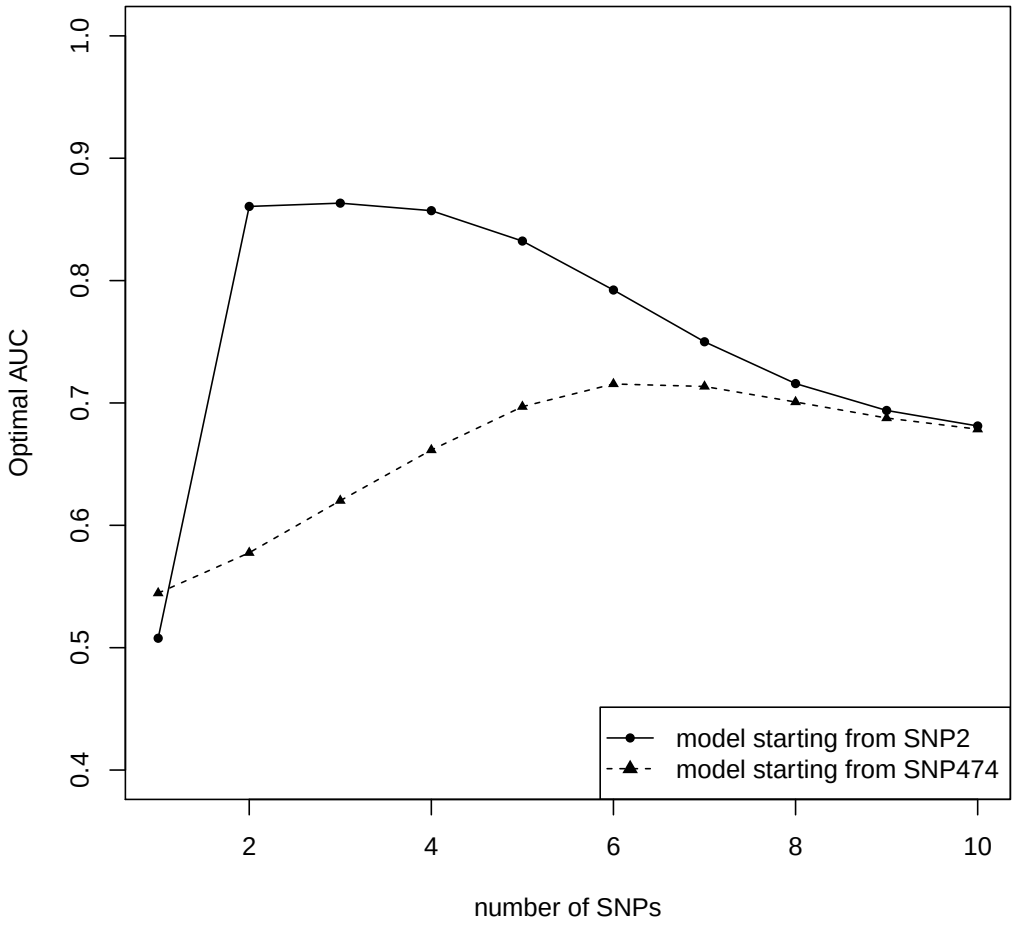

Figure 5.4: *Forward selection curves for the selection process starting with SNP1 (circles) and for the selection process starting with SNP474 (triangles).*

α, the critical value w_α is obtained by solving $P\left(W_{max} \leq w\right) = 1 - \alpha$ and gives

$$w_\alpha = \mu + \sigma\Phi^{-1}[(1-\alpha)^{1/p}] \ .$$

Thus, the best model is significant at a level α when $W_{max} > w_\alpha$.

Parameters μ and α are estimated empirically through a permutational approach.

In the example with two interaction SNPs, the best model is the one starting with

SNP1 or the one starting with SNP2. Both models achieve the maximum area under the forward selection curve among all the models, equal to $W_{max} = 0.7731$. The critical value in this example is $w_\alpha = 0.6635$ for $\alpha = 0.05$. Thus, the best model is significant.

5.3.3 Pruning

When the best model is significant, a further step is performed in order to refine the selected subset of SNPs. This is similar to the pruning process in classification and regression trees. The goal of this pruning process is to remove those SNPs that do not increment significantly the W area. To do this we implement a backward elimination process, we evaluate the SNPs in the reverse order that they were selected and we remove them if their corresponding area increase is not significant. The pruning process stops when a first significant SNP is found.

We start testing wether the last SNP, $G_{(m)}$ is significant or not. $G_{(m)}$ was selected as the SNP providing the maximum increment in the optimal AUC, thus, we test the significance of this increment by using the extreme value distribution. We assume that the increments follow a normal distribution and its parameters μ and σ are obtained through a permutational approach. In the example of Figure 5.4, the optimal set after pruning is reduced to the pair of interacting SNPs: $SNP1,\ SNP2$.

5.3.4 Missing data

An important issue when dealing with multivariate genetic models is how to handle missing genotypes. At each step of the forward stepwise process, a new SNP is added to the selected set and a new partition of the dataset is made. After that, the LR for each strata of the partition are computed. The problem arises with the classification of the subjects that present a missing value for the added SNP. In this

situation, we calculate the probability of the subject to belong to the new strata based on their previous strata and the genotypes frequencies of the new SNP in that strata. Finally we add the subject in all strata with a weight equal to the probability of belonging to stratum. The new formulas for LR estimations considering missing data are

$$\hat{P}\left(g_1, \ldots, g_k | Y = 1\right) = \frac{\sum_{i=1}^{n} 1\left\{Y^i = 1\right\} \prod_{j=1}^{k} w(G_j^i = g_j)}{\sum_{i=1}^{n} 1\left\{Y^i = 1\right\}} + \delta$$

$$\hat{P}\left(g_1, \ldots, g_k | Y = 0\right) = \frac{\sum_{i=1}^{n} 1\left\{Y^i = 0\right\} \prod_{j=1}^{k} w(G_j^i = g_j)}{\sum_{i=1}^{n} 1\left\{Y^i = 0\right\}} + \delta$$

where

$$w(G_j^i = g_j) = \begin{cases} 1 \text{ if } G_j^i = g_j \\ 0 \text{ if } G_j^i \neq g_j \\ \hat{P}\left(G_j^i = g_j\right) \text{ if } G_j^i \text{ is missing} \end{cases}$$

and $\hat{P}\left(G_j^i = g_j\right)$ is the genotype frequency of g_j for the observed individuals.

For instance, given a partition with three SNPs, an individual who has the genetic profile $(0, \mathrm{NA}, 1)$, that is, with a missing value in the second SNP, will contribute to the genetic profile $(0, 0, 1)$ with a weight of $w(G_2^i = 0)$ equal to the frequency of genotype "0" for SNP2, to the $(0, 1, 1)$ profile with a weight of $w(G_2^i = 1)$ equal to the frequency of genotype "1" for SNP2 and to the genetic profile $(0, 2, 1)$ with a weight of $w(G_2^i = 2)$ equal to the frequency of genotype "2" for SNP2.

5.4 Results

5.4.1 Simulation results

We conducted a simulation study to determine the power of the optimal AUC algorithm to identify the SNPs involved in high-order interactions. We compared the results with a marginal approach, the univariate logistic regression model.

We simulated case-control datasets consisting of a binary response variable (disease status) and 100 SNPs using the R functions described in chapter 2. First, using function `SNPgenerate` we obtained the genotypes for 100 independent SNPs. Second, using function `SNPinteact` we generated the response variable assuming a multiplicative odds of risk model (Section 2.3.2) and a K-th order interaction or epistatic effect of the K first SNPs for $K \in \{2, 3, 4, 5\}$. As described in Subsection 2.4.2, this is achieved by randomly assigning the multi-locus genotypes $(G_1, \ldots, G_K)$ of the causal interacting SNPs to a latent variable $L \in \{0, 1, 2\}$. We specify the relative risk of the three categories in L as $RR = P(Y = 1|L = 1)/P(Y = 1|L = 0)$ and $RR^2 = P(Y = 1|L = 2)/P(Y = 1|L = 0)$. We considered two values for the relative risk, $RR = 2$ (high risk) and $RR = 1.5$ (low risk) and two different values of disease prevalence, 0.1 and 0.02. In total we simulated 16 different scenarios resulting from all combinations of these parameters. We maintained fixed the minor allele frequency of every SNP at 0.2 and we generated 100 different datasets of size 4000 (2000 cases and 2000 controls) for each scenario.

For each simulated data set we applied the complete Optimal AUC process to build the optimal subset of SNPs and analyzed their significance. In these simulations we did not perform the pruning process, instead, we considered subsets of a fixed number of SNPs equal to K, the number of causal SNPs.

As mentioned before, we also applied a marginal approach, the univariate logistic regression model where each SNP is tested separately for association with the disease. For each SNP we considered four possible models of inheritance (dominant, recessive, additive and codominant) and take the minimum p-value of the four models. We set a SNP to be associated with the response when the p-value was smaller than the significance level ($\alpha = 0.05$) after adjusting for multiple testing using the Benjamini and Hochger FDR correction (Benjamini and Hochberg, 1995).

As a measure of performance of both approaches for detecting an interaction of K-th order we provide N_J, the number of simulated samples (over 100) for which the methods were able to detect at least J causal SNPs, $J \in \{1, 2, 3, \ldots, K\}$. The best performance is achieved when $N_K = 100$, meaning that the method was able to detect all K causal interacting SNPs in all simulated samples. The results are summarized in tables 5.2 to 5.5, each one corresponding to a different value of K. Each row corresponds to the different combinations of risk and prevalence parameters and each cell provides the value N_J for the Optimal AUC algorithm and for the marginal logistic regression in parenthesis.

The results prove a clear advantage of the Optimal AUC over the marginal logistic regression in all the simulated scenarios. For instance, in the first row of Table 5.2, that summarizes second order interactions, we observe that the Optimal AUC was able to detect both interacting SNPs 81% of the times while the marginal approach only detected both SNPs 67% of the times. The advantage of the Optimal AUC over the univariate approach is more evident as the number K of causal interacting SNPs increases. In the first row of Table 5.3 (third order interactions), the Optimal AUC detected the three causal SNPs 87% of the times while the marginal approach only 27% of the times. For interactions of order 4 and 5 the advantage is even more extreme. In the first row of Table 5.4, the Optimal AUC detected the 4 interacting SNPs 88% of the times while the marginal approach only 3% of the times and in the first row of Table 5.5, the Optimal AUC detected the 5 causal SNPs 96% of

the times while the marginal approach was never able to detect the five SNPs. We obtained similar results for a prevalence of 0.10 and a prevalence of 0.02, thus, we can conclude that the prevalence has not an important impact on the performance of both approaches. Instead, their performance is affected by the risk level. When the risk is low, we observe an important reduction of the number of causal SNPs detected. But, even in this case, the Optimal AUC performs much better than the univariate logistic regression.

Table 5.2: *Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting two interacting causal SNPs.*

Scenario		Number of detected causal SNPs	
Prevalence	Risk	1	2
0.1	High	100 (97)	81 (67)
0.02	High	100 (95)	79 (56)
0.1	Low	97 (77)	59 (24)
0.02	Low	89 (81)	42 (14)

Table 5.3: *Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting three interacting causal SNPs.*

Scenario		Number of detected causal SNPs		
Prevalence	Risk	1	2	3
0.1	High	100 (97)	98 (72)	87 (27)
0.02	High	100 (98)	99 (73)	87 (31)
0.1	Low	95 (74)	70 (17)	40 (1)
0.02	Low	93 (61)	69 (14)	37 (1)

Table 5.4: *Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting four interacting causal SNPs.*

Scenario		Number of detected causal SNPs			
Prevalence	Risk	1	2	3	4
0.1	High	100 (94)	100 (69)	99 (29)	88 (3)
0.02	High	100 (88)	100 (58)	98 (18)	89 (3)
0.1	Low	85 (54)	64 (16)	39 (1)	9 (0)
0.02	Low	75 (35)	58 (4)	26 (0)	8 (0)

Table 5.5: *Performance of Optimal AUC algorithm and univariate logistic regression (in parenthesis) for detecting five interacting causal SNPs.*

Scenario		Number of detected causal SNPs				
Prevalence	Risk	1	2	3	4	5
0.1	High	100 (77)	100 (44)	99 (14)	99 (0)	96 (0)
0.02	High	100 (79)	99 (37)	94 (16)	86 (3)	78 (0)
0.1	Low	69 (27)	51 (6)	29 (0)	16 (0)	3 (0)
0.02	Low	64 (30)	36 (3)	18 (0)	7 (0)	2 (0)

5.4.2 Alzheimer results

In this section we present the results for the Alzheimer disease association study introduced in chapter 1.2. As described there, the data corresponds to a genome-wide association study conducted by Reiman et al. (2007) with $502,627$ SNPs genotyped for 1411 subjects of whom 861 were cases and 550 controls. An indicator variable is also available that reports whether the individual is carrying the apolipoprotein E (ApoE) $\epsilon 4$ allelic variant, which is the best established genetic risk factor for late-

onset Alzheimer's disease (Reiman et al., 2007). Here we focus our analysis on the Reelin signaling pathway with the goal of identifying genetic risk profiles or interactions in genes within this pathway and to analyze potential interactions between ApoE genotypes and those SNPs. For this, we identified 32 genes related to the Reelin pathway and extracted 682 SNPs from Reiman's database within these genes (Table 1.4).

We applied the Optimal AUC algorithm for each gene separately. We conducted a global analysis with all the patients, and a stratified analysis by the ApoE-ϵ4 indicator. We set the number of SNPs explored in each Optimal AUC process to be m = 10. Afterwards, pruning was performed for those significant results. Table 5.6 shows the results for the stratified analysis. No significant subsets of SNPs were detected in the global analysis.

For non carriers of ApoE-ϵ4 we obtained three significant interactions. The first result corresponds to a second order interaction between SNPs rs2855563 and rs6426554 in PSEN2 gene. This gene is located on the long arm of chromosome 1 and provides instructions for making a protein called presenilin 2 that plays a role in processing amyloid precursor protein. Patients with an inherited form of AD carry mutations in the presenilin proteins or the amyloid precursor protein. The second significant finding is also a second order interaction between SNPs rs11030102 and rs11030104 in BDNF gene, located on the short arm of chromosome 11. The protein encoded by BDNF promotes the survival of neurons by playing a role in their growth, differentiation and maintenance. The BDNF protein helps to regulate synaptic plasticity, which is the ability of connections between neurons (synapses) to change and adapt over time in response to experience. The last significant result is the interaction between SNPs rs2636277 and rs41346745 in ABL2 gene which encodes a non-receptor tyrosine-protein kinase. In brain, it may regulate neurotransmission by phosphorylating proteins at the synapse. ABL2 acts also as a regulator of multiple pathological signaling cascades during infection.

Table 5.6: *Significant sets of SNPs obtained by Optimal AUC algorithm from Alzheimer's disease data set.*

ApoE-ϵ4	Gene	SNPs	
non carriers	PSEN2	rs2855563	rs6426554
	BDNF	rs11030102	rs11030104
	ABL2	rs2636277	rs41346745
carriers	RELN	rs1705107	rs4727582
		rs2245617	rs3905915
		rs17310949	rs4727583

For the stratum of ApoE-ϵ4 carriers, a significant subset of 6 SNPs was detected in the RELN gene (rs1705107, rs4727582, rs2245617, rs3905915, rs17310949 and rs4727583). The RELN gene is located on the long arm of chromosome 7 and encodes a protein called reelin that plays a role in layering of neurons in the cerebral cortex and cerebellum. Reelin likely plays a role in many brain processes, including the extension of axons and dendrites, which are specialized outgrowths from nerve cells that are essential for the transmission of nerve impulses. Reelin may also regulate synaptic plasticity.

The Optimal AUC algorithm identified possible sets of SNPs that are jointly associated with Alzheimer's disease in interaction with ApoE-ϵ4. These results should be confirmed in independent studies.

The information about gene functions in this section was compiled from the Genetics Home Reference (GHR, 2015), the National Center for Biotechnology Information (NCBI, 2015) and the UniProt Protein knowledgebase (The UniProt Consortium, 2015).

Bibliography

Benjamini, Y. and 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)*, pages 289–300.

Boulesteix, A.-L. and Slawski, M. (2009). Stability and aggregation of ranked gene lists. *Brief Bioinform*, 10(5):556–568.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Bureau, A., Dupuis, J., Falls, K., Lunetta, K. L., Hayward, B., Keith, T. P., and Van Eerdewegh, P. (2005). Identifying snps predictive of phenotype using random forests. *Genet Epidemiol*, 28(2):171–182.

Calle, M. L. and Urrea, V. (2011). Letter to the editor: Stability of random forest importance measures. *Brief Bioinform*, 12(1):86–89.

Calle, M. L., Urrea, V., Boulesteix, A.-L., and Malats, N. (2011). AUC-RF: a new strategy for genomic profiling with random forest. *Hum Hered*, 72(2):121–132.

Calle, M. L., Urrea, V., Malats, N., and Van Steen, K. (2010). mbmdr: an r package for exploring gene-gene interactions associated with binary or quantitative traits. *Bioinformatics*, 26(17):2198–2199.

Calle, M. L., Urrea, V., Vellalta, G., Malats, N., and Steen, K. V. (2008). Improving strategies for detecting genetic patterns of disease susceptibility in association studies. *Stat Med*, 27(30):6532–6546.

Cantor, R. M., Lange, K., and Sinsheimer, J. S. (2010). Prioritizing gwas results: A review of statistical methods and recommendations for their application. *Am J Hum Genet*, 86(1):6–22.

Diaz-Uriarte, R. (2007). Genesrf and varselrf: a web-based tool and r package for gene selection and classification using random forest. *BMC Bioinformatics*, 8:328.

Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3.

García-Closas, M., Malats, N., Silverman, D., Dosemeci, M., Kogevinas, M., Hein, D. W., Tardón, A., Serra, C., Carrato, A., García-Closas, R., Lloreta, J., Castaño-Vinyals, G., Yeager, M., Welch, R., Chanock, S., Chatterjee, N., Wacholder, S., Samanic, C., Torà, M., Fernández, F., Real, F. X., and Rothman, N. (2005). Nat2 slow acetylation, gstm1 null genotype, and risk of bladder cancer: results from the spanish bladder cancer study and meta-analyses. *Lancet*, 366(9486):649–659.

GHR (2015). National Library of Medicine (US). Genetics Home Reference [Internet]. Bethesda (MD): The Library; 2015 Feb 22. Available from: http://ghr.nlm.nih.gov.

Guey, L. T., García-Closas, M., Murta-Nascimento, C., Lloreta, J., Palencia, L., Kogevinas, M., Rothman, N., Vellalta, G., Calle, M. L., Marenne, G., Tardón, A., Carrato, A., García-Closas, R., Serra, C., Silverman, D. T., Chanock, S., Real, F. X., Malats, N., and , E. P. I. C. U. R. O. B. C. S. i. (2010). Genetic susceptibility to distinct bladder cancer subphenotypes. *Eur Urol*, 57(2):283–292.

Gui, J., Moore, J. H., Kelsey, K. T., Marsit, C. J., Karagas, M. R., and Andrew, A. S. (2011). A novel survival multifactor dimensionality reduction method for

detecting gene-gene interactions with application to bladder cancer prognosis. *Hum Genet*, 129(1):101–110.

I.H.C. (2003). The international hapmap project. *Nature*, 426(6968):789–796.

Jakobsdottir, J., Gorin, M. B., Conley, Y. P., Ferrell, R. E., and Weeks, D. E. (2009). Interpretation of genetic association studies: markers with replicated highly significant odds ratios may be poor classifiers. *PLoS Genet*, 5(2):e1000337.

Janssens, A. C. J. W., Aulchenko, Y. S., Elefante, S., Borsboom, G. J. J. M., Steyerberg, E. W., and van Duijn, C. M. (2006). Predictive testing for complex diseases using multiple genes: fact or fiction? *Genet Med*, 8(7):395–400.

Janssens, A. C. J. W., Moonesinghe, R., Yang, Q., Steyerberg, E. W., van Duijn, C. M., and Khoury, M. J. (2007). The impact of genotype frequencies on the clinical validity of genomic profiling for predicting common chronic diseases. *Genet Med*, 9(8):528–535.

Jiang, H., Deng, Y., Chen, H.-S., Tao, L., Sha, Q., Chen, J., Tsai, C.-J., and Zhang, S. (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 5:81.

Kraft, P., Wacholder, S., Cornelis, M. C., Hu, F. B., Hayes, R. B., Thomas, G., Hoover, R., Hunter, D. J., and Chanock, S. (2009). Beyond odds ratios–communicating disease risk based on genetic profiles. *Nat Rev Genet*, 10(4):264–269.

Li, C. and Li, M. (2008). Gwasimulator: a rapid whole-genome simulation program. *Bioinformatics*, 24(1):140–142.

Lu, Q. and Elston, R. C. (2008). Using the optimal receiver operating characteristic curve to design a predictive genetic test, exemplified with type 2 diabetes. *Am J Hum Genet*, 82(3):641–651.

Lu, Q., Obuchowski, N., Won, S., Zhu, X., and Elston, R. C. (2010). Using the optimal robust receiver operating characteristic (roc) curve for predictive genetic tests. *Biometrics*, 66(2):586–593.

Marchini, J., Donnelly, P., and Cardon, L. R. (2005). Genome-wide strategies for detecting multiple loci that influence complex diseases. *Nat Genet*, 37(4):413–417.

McIntosh, M. W. and Pepe, M. S. (2002). Combining several screening tests: optimality of the risk score. *Biometrics*, 58(3):657–664.

Moonesinghe, R., Liu, T., and Khoury, M. J. (2010). Evaluation of the discriminative accuracy of genomic profiling in the prediction of common complex diseases. *Eur J Hum Genet*, 18(4):485–489.

NCBI (2015). National Library of Medicine (US). National Center for Biotechnology Information [Internet]. Bethesda (MD): The Library; 2015 Feb 22. Available from: http://www.ncbi.nlm.nih.gov.

Pepe, M. S. (2003). *The statistical evaluation of medical tests for classification and prediction*. Oxford University Press.

Pepe, M. S., Janes, H., Longton, G., Leisenring, W., and Newcomb, P. (2004). Limitations of the odds ratio in gauging the performance of a diagnostic, prognostic, or screening marker. *Am J Epidemiol*, 159(9):882–890.

Pepe, M. S., Longton, G., Anderson, G. L., and Schummer, M. (2003). Selecting differentially expressed genes from microarray experiments. *Biometrics*, 59(1):133–142.

Reiman, E. M., Webster, J. A., Myers, A. J., Hardy, J., Dunckley, T., Zismann, V. L., Joshipura, K. D., Pearson, J. V., Hu-Lince, D., Huentelman, M. J., Craig, D. W., Coon, K. D., Liang, W. S., Herbert, R. H., Beach, T., Rohrer, K. C., Zhao, A. S., Leung, D., Bryden, L., Marlowe, L., Kaleem, M., Mastroeni, D., Grover, A., Heward, C. B., Ravid, R., Rogers, J., Hutton, M. L., Melquist, S., Petersen,

R. C., Alexander, G. E., Caselli, R. J., Kukull, W., Papassotiropoulos, A., and Stephan, D. A. (2007). Gab2 alleles modify alzheimer's risk in apoe epsilon4 carriers. *Neuron*, 54(5):713–720.

Rice, D. S. and Curran, T. (2001). Role of the reelin signaling pathway in central nervous system development. *Annu Rev Neurosci*, 24:1005–1039.

Ritchie, M. D., Hahn, L. W., and Moore, J. H. (2003). Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. *Genet Epidemiol*, 24(2):150–157.

Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., and Moore, J. H. (2001). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *Am J Hum Genet*, 69(1):138–147.

Samanic, C., Kogevinas, M., Dosemeci, M., Malats, N., Real, F. X., Garcia-Closas, M., Serra, C., Carrato, A., García-Closas, R., Sala, M., Lloreta, J., Tardón, A., Rothman, N., and Silverman, D. T. (2006). Smoking and bladder cancer in spain: effects of tobacco type, timing, environmental tobacco smoke, and gender. *Cancer Epidemiol Biomarkers Prev*, 15(7):1348–1354.

Schaffner, S. F., Foo, C., Gabriel, S., Reich, D., Daly, M. J., and Altshuler, D. (2005). Calibrating a coalescent simulation of human genome sequence variation. *Genome Res*, 15(11):1576–1583.

Seshadri, S., Drachman, D. A., and Lippa, C. F. (1995). Apolipoprotein e epsilon 4 allele and the lifetime risk of alzheimer's disease. what physicians know, and what they should know. *Arch Neurol*, 52(11):1074–1079.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, 9:307.

Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007). Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics*, 8:25.

The UniProt Consortium (2015). Uniprot: a hub for protein information. Nucleic Acids Res. 43: D204-D212. Available from http://www.uniprot.org.

Van Steen, K. (2012). Travelling the world of gene-gene interactions. *Brief Bioinform*, 13(1):1–19.

Wei, Z., Wang, K., Qu, H.-Q., Zhang, H., Bradfield, J., Kim, C., Frackleton, E., Hou, C., Glessner, J. T., Chiavacci, R., Stanley, C., Monos, D., Grant, S. F. A., Polychronakos, C., and Hakonarson, H. (2009). From disease association to risk assessment: an optimistic view from genome-wide association studies on type 1 diabetes. *PLoS Genet*, 5(10):e1000678.

Wray, N. R. and Goddard, M. E. (2010). Multi-locus models of genetic risk of disease. *Genome Med*, 2(2):10.

Wray, N. R., Goddard, M. E., and Visscher, P. M. (2007). Prediction of individual genetic risk to disease from genome-wide association studies. *Genome Res*, 17(10):1520–1528.

Wray, N. R., Yang, J., Goddard, M. E., and Visscher, P. M. (2010). The genetic interpretation of area under the roc curve in genomic profiling. *PLoS Genet*, 6(2):e1000864.

Wright, F. A., Huang, H., Guan, X., Gamiel, K., Jeffries, C., Barry, W. T., de Villena, F. P.-M., Sullivan, P. F., Wilhelmsen, K. C., and Zou, F. (2007). Simulating association studies: a data-based resampling method for candidate regions or whole genome scans. *Bioinformatics*, 23(19):2581–2588.

Ye, C., Cui, Y., Wei, C., Elston, R. C., Zhu, J., and Lu, Q. (2011). A non-parametric

method for building predictive genetic tests on high-dimensional data. *Hum Hered*, 71(3):161–170.

APPENDIX A

Functions for the simulation of genetic risk profiles

SNPgenerate function

Simulation of genotype data on the assumption of independent SNPs.

```
SNPgenerate <- function(n,maf){
 x <- NULL
 for(i in 1:length(maf)){
  p <- maf[i]
  genfreq <- c((1-p)^2,2*p*(1-p),p^2)
  x[[paste("SNP",i,"_",maf[i],sep="")]] <-
   sample(factor(c(0,1,2)), n, replace = TRUE, prob = genfreq)
 }
 return(as.data.frame(x))
}
```

LDgenerate function

Simulation of genotype data on the assumption of SNPs in LD.

```
LDgenerate <- function(x,r,maf=NULL,mafx=NULL){
 if(missing(mafx)){
  freq <- table(x)
  if(length(freq)!=3)
    stop("Error in GeneraSNPinLD: x must be 3 levels")
  mafx <- 1 - (freq["1"]+2*freq["0"])/(2*sum(freq))
 }
 if(missing(maf)) maf <- rep(mafx,length(r))
 if(length(maf)!=length(r))
  stop("r and maf arguments have different lengths")
 z <- NULL
 for(i in 1:length(r)){
  maf2 <- maf[i]
  d <- r[i]*sqrt(mafx*(1-mafx)*maf2*(1-maf2))
  if(d>0)  d <- min( d, min((1-mafx)*maf2,mafx*(1-maf2)) )
```

```
  if(d<0)  d <- max( d, -mafx*maf2 )
  fAA <- (1-mafx)^2
  fAa <- 2*(1-mafx)*mafx
  faa <- mafx^2
  fAB <- (1-mafx)*(1-maf2) + d
  fAb <- (1-mafx)*maf2 - d
  faB <- mafx*(1-maf2) - d
  fab <- mafx*maf2 + d
  p0.0 <- fAB^2 / fAA
  p1.0 <- 2*fAB*fAb / fAA
  p2.0 <- fAb^2 / fAA
  p0.1 <- 2*fAB*faB / fAa
  p1.1 <- (2*fAB*fab + 2*fAb*faB) / fAa
  p2.1 <- 2*fAb*fab / fAa
  p0.2 <- faB^2 / faa
  p1.2 <- 2*faB*fab / faa
  p2.2 <- fab^2 / faa
  condProb <- rbind("0"=c(p0.0,p1.0,p2.0), "1"=c(p0.1,p1.1,p2.1),
    "2"=c(p0.2,p1.2,p2.2))
  colnames(condProb) <- c("0","1","2")
  y <- rep(NA,length(x))
  ind0 <- !is.na(x) & x==0
  y[ind0] <-
    sample(c(0,1,2), sum(ind0), replace=TRUE, prob=condProb["0",])
  ind1 <- !is.na(x) & x==1
  y[ind1] <-
    sample(c(0,1,2), sum(ind1), replace=TRUE, prob=condProb["1",])
  ind2 <- !is.na(x) & x==2
  y[ind2] <-
    sample(c(0,1,2), sum(ind2), replace=TRUE, prob=condProb["2",])
  aux <- paste("LDx_",r[i],"_SNP",i,"_",maf2,sep="")
  z[[aux]] <- factor(y, levels=c(0,1,2))
 }
 return(as.data.frame(z))
}
```

RiskGenerate function

Simulation of disease outcome from multiple independent SNPs.

```
RiskGenerate <- function(data,RR,p){
 LRsnp <- function(x,p,RR1,maf=NULL){
  if(missing(maf)){
   freq <- table(x)
   maf <- 1 - (freq["1"]+2*freq["0"])/(2*sum(freq))
  }
  RR2 <- RR1^2
  g0 <- (1-maf)^2
  g1 <- 2*maf*(1-maf)
  g2 <- maf^2
  e <- p*g0/(g0 + RR2*g2 + RR1*g1)
  a <- RR2*e*g2/g0
  c <- RR1*e*g1/g0
  b <- g2 - a
  d <- g1 - c
  f <- g0 - e
  LR0 <- e*(1-p)/(f*p)
  LR1 <- c*(1-p)/(d*p)
  LR2 <- a*(1-p)/(b*p)
  if(LR0<0 | LR1<0 | LR2<0)
    stop("Incompatible combination of RR and p")
  return(c(LR0=LR0,LR1=LR1,LR2=LR2))
 }
 if(length(RR)>1){
   LRschema <- NULL
   for(i in 1:ncol(data))
    LRschema <- rbind(LRschema,LRsnp(data[,i],p=p,RR1=RR[i]))
 }
 else{
   LRschema <- t(sapply(data,LRsnp,p=p,RR1=RR))
 }
```

```
 LR <- NULL
 for(i in 1:nrow(LRschema)) LR <- cbind(LR, LRschema[i,data[,i]])
 rownames(LR) <- rownames(data)
 log.odd <- apply(log(LR),1,sum) + log(p/(1-p))
 odd <- exp(log.odd)
 prob <- odd/(1+odd)
 unif <- runif(length(prob))
 y <- ifelse(prob>unif,1,0)
 return(y)
}
```

SNPInteract function

Simulation of disease outcome under a genetic interaction disease model.

```
SNPInteract <- function(data,RR,p,hrp=NULL){
 if(missing(hrp)){
  mafData <- sapply(data, function(x){
    freq <- table(x)
    return(1 - (freq["1"]+2*freq["0"])/(2*sum(freq)))})
  lmaf <- mean(mafData)
 }
 else{
  lmaf <- sqrt(hrp)
 }
 n <- nrow(data)
 maf <- lmaf
 RR1 <- RR
 RR2 <- RR1^2
 g0 <- (1-maf)^2
 g1 <- 2*maf*(1-maf)
 g2 <- maf^2
  e <- p*g0/(g0 + RR2*g2 + RR1*g1)
```

```
 a <- RR2*e*g2/g0
 c <- RR1*e*g1/g0
 b <- g2 - a
 d <- g1 - c
 f <- g0 - e
LR0 <- e*(1-p)/(f*p)
LR1 <- c*(1-p)/(d*p)
LR2 <- a*(1-p)/(b*p)

Partition <- function(p1,p2,index=NULL){
 if(is.null(index)){
  groups <- paste(p2,p1,sep=".")
 }
 else{
  groups <- p2
  groups[index] <- paste(p2[index],p1[index],sep=".")
 }
 x <- as.numeric(factor(groups))
 return(x)
}

PartitionVar <- function(data){
 if(class(data)=="data.frame"){
  partition <- as.numeric(factor(data[,1]))
   for(i in 2:ncol(data)) partition <- Partition(data[,i],partition)
 }
 else{
  partition <- as.numeric(factor(data))
 }
 return(partition)
}
part <- PartitionVar(data)

aux <- factor(part)
aux <- factor(aux,levels=sample(levels(aux)))
```

```
 aux <- as.numeric(aux)
 oscor <- order(aux)
 AA <- round(n*(1-maf)^2,0)
 Aa <- round(n*2*(1-maf)*maf,0)
 aa <- n-AA-Aa
 X <- as.factor(c(rep(0,AA),rep(1,Aa),rep(2,aa)))
 X <- X[order(oscor)]

 LRschema <- c(LR0,LR1,LR2)
 LR <- LRschema[X]
 names(LR) <- rownames(data)
 log.odd <- log(LR) + log(p/(1-p))
 odd <- exp(log.odd)
 prob <- odd/(1+odd)
 unif <- runif(n)
 y <- ifelse(prob>unif,1,0)
 return(y)
}
```

Bin2Cont function

Simulation of continous outcome from binary outcome.

```
Bin2Cont <- function(y,mean0,sd0,mean1,sd1){
 n <- length(y)
 n0 <- sum(y==0)
 n1 <- sum(y==1)
 if(n!=n1+n0) stop("Error in y")
 z <- rep(NA,n)
 z[y==0] <- rnorm(n0,mean0,sd0)
 z[y==1] <- rnorm(n1,mean1,sd1)
 return(z)
}
```

Bin2Surv function

Simulation of survival time outcome from binary outcome.

```
Bin2Surv <- function(y,tmax,shape0,scale0,shape1,scale1){
 n <- length(y)
 n0 <- sum(y==0)
 n1 <- sum(y==1)
 if(n!=n1+n0) stop("Error in y")
 z <- rep(NA,n)
 z[y==0] <- rweibull(n0,shape0,scale0)
 z[y==1] <- rweibull(n1,shape1,scale1)
 tuni <- runif(n,0,tmax)
 ind <- z<=tuni
 cens <- 1*ind
 z[!ind] <- tuni[!ind]
 return(data.frame(time=z,cens=cens))
}
```

APPENDIX B

AUCRF (R language package)

Package 'AUCRF'

August 29, 2013

Type Package

Title Variable Selection with Random Forest and the Area Under the Curve

Version 1.1

Date 2012-03-19

Author Victor Urrea, M.Luz Calle

Maintainer Victor Urrea <victor.urrea@uvic.cat>

Depends R (>= 2.11.0), randomForest

Description Variable selection using Random Forest based on optimizing the area-under-the ROC curve (AUC) of the Random Forest.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2012-03-19 11:12:24

NeedsCompilation no

R topics documented:

AUCRF 2
AUCRFcv 4
OptimalSet 5
plot.AUCRF 6

Index **8**

AUCRF *Variable Selection with Random Forest and the Area Under the Curve*

Description

AUCRF is an algorithm for variable selection using Random Forest based on optimizing the area-under-the ROC curve (AUC) of the Random Forest. The proposed strategy implements a backward elimination process based on the initial ranking of the variables.

Usage

```
AUCRF(formula, data, k0 = 1, pdel = 0.2, ranking=c("MDG","MDA"), ...)
```

Arguments

`formula`	an object of class `formula`: a symbolic description of the model to be fitted. The details of model specification are given in `Details`.
`data`	a data frame containing the variables in the model. Dependent variable must be a binary variable defined as `factor` and codified as `1` for positives (e.g. cases) and `0` for negatives (e.g. controls).
`k0`	number of remaining variables for stopping the backward elimination process. By default `k0=1`.
`pdel`	fraction of remaining variables to be removed in each step. By default `pdel=0.2`. If `pdel=0`, only one variable is removed each time.
`ranking`	specifies the importance measure provided by `randomForest` for ranking the variables. There are two options MDG (by default) for MeanDecreaseGini and MDA for MeanDecreaseAccuracy.
`...`	optional parameters to be passed to the `randomForest` function. If no arguments are specified, default arguments of `randomForest` function will be used.

Details

The AUC-RF algorithm is described in detail in Calle et. al.(2011). The following is a summary:

Ranking and AUC of the initial set:
Perform a random forest using all predictor variables and the response, as specified in the `formula` argument, and compute the AUC of the random forest. Based on the selected measure of importance (by default MDG), obtain a ranking of predictors.

Elimination process:
Based on the variables ranking, remove the less important variables (fraction of variables specified in `pdel` argument). Perform a new random forest with the remaining variables and compute its AUC. This step is iterated until the number of remaining variables is less or equal than `k0`.

Optimal set:
The optimal set of predictive variables is considered the one giving rise to the Random Forest with the highest OOB-AUCopt. The number of selected predictors is denoted by Kopt

Value

An object of class AUCRF, which is a list with the following components:

call	the original call to AUCRF.
data	the data argument.
ranking	the ranking of predictors based on the importance measure.
Xopt	optimal set of predictors obtained.
OOB-AUCopt	AUC obtained for the optimal set of predictors.
Kopt	size of the optimal set of predictors obtained.
AUCcurve	values of AUC obtained for each set of predictors evaluated in the elimination process.
RFopt	the randomForest adjusted with the optimal set.

References

Calle ML, Urrea V, Boulesteix A-L, Malats N (2011) "AUC-RF: A new strategy for genomic profiling with Random Forest". Human Heredity. (In press)

See Also

OptimalSet, AUCRFcv, randomForest.

Examples

```
# load the included example dataset. This is a simulated case/control study
# data set with 4000 patients (2000 cases / 2000 controls) and 1000 SNPs,
# where the  first 10 SNPs have a direct association with the outcome:
data(exampleData)

# call AUCRF process: (it may take some time)
# fit <- AUCRF(Y~., data=exampleData)

# The result of this example is included for illustration purpose:

data(fit)
summary(fit)
plot(fit)

# Additional randomForest parameters can be included, otherwise default
# parameters of randomForest function will be used:
# fit <- AUCRF(Y~., data=exampleData, ntree=1000, nodesize=20)
```

AUCRFcv *Repeated cross validation of the AUC-RF process.*

Description

Performes a repeated cross validation analysis and computes the probability of selection for each variable.

Usage

```
AUCRFcv(x, nCV = 5, M = 20)
```

Arguments

x	an object of class AUCRF.
nCV	number of folds in cross validation. By default a 5-fold cross validation is performed.
M	number of cross validation repetitions.

Details

The results of this repeated cross validation analysis are (1) a corrected estimation of the predictive accuracy of the selected variables and (2) an estimate of the probability of selection for each variable.
The AUC-RF algorithm is exhaustively described in Calle et. al.(2011).

Value

The same AUCRF object passed (see AUCRF) as argument but updated with the following components:

cvAUC	mean of AUC values in test datasets of the optimal sets of predictors.
Psel	probability of selection of each variable as the proportion of times that is selected by AUC-RF method.

References

Calle ML, Urrea V, Boulesteix A-L, Malats N (2011) "AUC-RF: A new strategy for genomic profiling with Random Forest". Human Heredity. (In press)

See Also

OptimalSet, AUCRF, randomForest.

Examples

```
# Next steps take some time

# load included AUCRF result example:
# data(fit)

# call AUCRFcv process:
# fitCV <- AUCRFcv(fit)

# The result of this example is included:

data(fitCV)
summary(fitCV)
plot(fitCV)
```

`OptimalSet` *AUCRF optimal set selection.*

Description

Returns the optimal set of predictive variables selected by the AUC-RF method.

Usage

```
OptimalSet(object)
```

Arguments

`object`	an object of class `AUCRF` as the result of `AUCRF` or `AUCRFcv` functions.

Value

A data.frame with the selected variables ordered by the initial ranking, their importance values (initial ranking) and, if available, the probability of selection value measured by `AUCRFcv` function.

See Also

`AUCRF`, `AUCRFcv`.

Examples

```
data(fitCV)
OptimalSet(fitCV)
```

plot.AUCRF *Plot Method for AUCRF*

Description

The plot method for AUCRF objects.

Usage

```
  ## S3 method for class 'AUCRF'
plot(x, which=c("auc","ranking","psel"), showOpt=TRUE, digits=4,
       maxvars=NULL, ...)
```

Arguments

x	an object of class AUCRF as the result of AUCRF or AUCRFcv functions.
which	specifies the information to plot. There are three options: (1) "auc" (by default) to plot the curve of AUCs in the backwards elimination process, (2) "ranking" to plot the importance measure in initial ranking of each variable, and (3) "psel" to plot the probability of selection of each variable. The "psel" option is only available if a cross validation is performed by AUCRFcv function. For option (1), showOpt and digits arguments can be specified for more details (see below). For options (2) and (3), the number of variables to plot and their order preference can be specified by maxvars and order arguments, respectively (see below).
showOpt	(only applied if "auc" option is specified in wich argument). If showOpt=TRUE, the optimal subset is emphasised in the plot.
digits	(only applied if "auc" option is specified in wich argument and showOpt or showThres are TRUE). Specifies the number of decimal digits for representing the optimal AUC in the plot.
maxvars	(only applied if "ranking" or "psel" options are specified in wich argument). Number of variables to include in the plot. The specified number of variables with highest importance measure (initial ranking) will be plotted. If maxvars=NULL (by default) the selected variables will be plotted. (For large number of variables, their names can be illegible in the plot)
...	other graphical parameters (see par).

Examples

```
data(fitCV)

# Plotting the AUC in the AUCRF backward elimination process:
plot(fitCV)

# Plotting the probability of selection of the selected variables:
plot(fitCV, wich="psel")
```

```
# Plotting the 20 variables with highest probability of selection:
plot(fitCV, wich="psel", maxvars=20)
```

AUCRF font code

AUCRF function

Variable Selection with Random Forest and the Area Under the Curve.

```
AUCRF <-
function(formula,data,k0=1,pdel=0.2,ranking=c("MDG","MDA"),...){

AUC.randomForest <-
function(rf,clase=1){
r <- rank(rf$votes[,as.character(clase)])
rd <- mean(r[rf$y==clase])
nd <- sum(rf$y==clase)
nnd <- length(rf$y)-nd
return((rd-nd/2-0.5)/nnd)
}

MDGRanking <-
function(formula,data,...){
fitRF <- randomForest(formula,data=data,...)
  mdgRanking <- sort(fitRF$importance[,"MeanDecreaseGini"],decreasing=
      TRUE)
  return(mdgRanking)
}

MDARanking <-
function(formula,data,...){
fitRF <- randomForest(formula,data=data,importance=TRUE,...)
  mdaRanking <- sort(fitRF$importance[,"MeanDecreaseAccuracy"],
      decreasing=TRUE)
  return(mdaRanking)
}
```

```
t <- 0
cl <- match.call()
mf <- match("formula", names(cl), 0L)
y <- eval(eval(cl[[mf]])[[2]],data)
if(!is.factor(y) && length(levels(y))!=2) stop("Outcome must be a
    factor with two levels")
if(pdel<0 || pdel>=1) stop("pdel must be in the interval [0,1)")

ranking <- match.arg(ranking)
switch(ranking,
         MDG = {ranking <- MDGRanking(formula,data,...); ImpMes <- "MDG
             "},
         MDA = {ranking <- MDARanking(formula,data,...); ImpMes <- "MDA
             "},
         stop("Not valid ranking")
       )


mf <- match("formula", names(cl), 0L)
yname <- as.character(eval(cl[[mf]])[[2]])
  vars <- names(ranking)
  AUCcurve <- data.frame()
  auxThres <- 0
  auxMaxAUC <- 0
  k <- length(vars)
while(k>=k0){
        fitRF <- randomForest(formula,data=data[,c(yname,vars[1:k])
            ],...)
        getAUC <- AUC.randomForest(fitRF)
        if(getAUC>=auxMaxAUC){
                auxMaxAUC <- getAUC
                auxThres <- auxMaxAUC-t
        }
        if(getAUC>=auxThres) RFopt <- fitRF
        AUCcurve <- rbind(c(k,getAUC),AUCcurve)
```

```
        k <- k-as.integer(pdel*k)-1
}

colnames(AUCcurve) <- c("k","AUC")
maxAUC <- max(AUCcurve$AUC)
opthreshold <- maxAUC-t
optimal <- AUCcurve[AUCcurve$AUC>=opthreshold,][1,]

objectList <- list()
objectList$call <- cl
objectList$data <- data
objectList$ranking <- ranking
objectList$Xopt <- names(ranking)[1:(optimal$k)]
objectList$"OOB-AUCopt" <- optimal$AUC
  objectList$Kopt <- optimal$k
objectList$AUCcurve <- AUCcurve
objectList$RFopt <- RFopt
objectList$ImpMeasure <- ImpMes
  class(objectList) <- "AUCRF"
  return(objectList)
}
```

AUCRFcv function

Repeated cross validation of the AUC-RF process.

```
AUCRFcv <-
function(x,nCV=5,M=20){

AUC.votes <-
function(votes,y=NULL,clase=1){
if(missing(y) || is.null(y)) y <- votes$y
r <- rank(votes[,as.character(clase)])
```

```
rd <- mean(r[y==clase])
nd <- sum(y==clase)
nnd <- length(y)-nd
return((rd-nd/2-0.5)/nnd)
}

cl <- match.call()
switch(class(x),
  "AUCRF" = {
      callRF <- x$call
      data <- x$data
      callRF$data <- as.name("newData")
      yname <- as.character(eval(x$call$formula)[[2]])
  },
stop("x must be a AUCRF object.")
)

cvAUC <- NULL
varnames <- colnames(data)[colnames(data)!=yname]
nSelect <- rep(0,length(varnames))
names(nSelect) <- varnames
for(m in 1:M){
  CV <- list()
  mpredict <- NULL
  indPermuted <- matrix(c(sample(rownames(data)),rep(NA,nCV-nrow(data)
     %%nCV)),ncol=nCV,byrow=TRUE)
  for(k in 1:nCV){
  indTest <- indPermuted[,k]
  indTest <- indTest[!is.na(indTest)]
  indTrain <- rownames(data)[!(rownames(data) %in% indTest)]
  newData <- data[indTrain,]
    kaucRF <- eval(callRF)
      mpredict <- rbind(mpredict, predict(kaucRF$RFopt,newdata=data[
         indTest,],type="vote"))
      nSelect[kaucRF$Xopt] <-  nSelect[kaucRF$Xopt]+1
```

```
   }
   mvotes <- data.frame(y=data[,yname],mpredict[rownames(data),])
   class(mvotes) <- c("votes","data.frame")
   colnames(mvotes) <- c("y","0","1")
    cvAUC <- c(cvAUC, AUC.votes(mvotes))
  }

  if(class(x)=="AUCRF")
  objectList <- x
  else
    objectList <- list()

  objectList$cvAUC <- mean(cvAUC)
  objectList$Psel <- nSelect/(M*nCV)
  objectList$callcv <- cl
  class(objectList) <- c("AUCRFcv","AUCRF")
  return(objectList)
}
```

OptimalSet function

AUCRF optimal set selection.

```
OptimalSet <-
function(object){
  if(is.null(object$Psel))
        out <- data.frame("Name"=object$Xopt,"Importance"=
           object$ranking[object$Xopt])
  else
        out <- data.frame("Name"=object$Xopt,"Importance"=
           object$ranking[object$Xopt], "Prob.Selection"=object$Psel[
           object$Xopt])
```

```
  rownames(out) <- NULL
  return(out)
}
```

summary.AUCRF function

Summary method for AUCRF.

```
summary.AUCRF <-
function(object,...){
  return(object)
}
```

print.AUCRF function

Print method for AUCRF.

```
print.AUCRF <-
function(x,...){
  cat("\nNumber of selected variables: Kopt=",x$Kopt,"\n")
  cat("AUC of selected variables: OOB-AUCopt=",x$"OOB-AUCopt","\n")
  if(!is.null(x$cvAUC)) cat("AUC from cross validation:",x$cvAUC,"\n")
  cat("Importance Measure:",x$ImpMeasure,"\n")
  cat("\n")

  if(is.null(x$Psel))
    res <- data.frame("Selected Variables"=x$Xopt, Importance=x$ranking
        [x$Xopt])
  else
        res <- data.frame("Selected Variables"=x$Xopt, Importance=
            x$ranking[x$Xopt], Prob.Select=x$Psel[x$Xopt])
  rownames(res) <- NULL
```

```
  print(res)
}
```

plot.AUCRF function

Plot method for AUCRF.

```
plot.AUCRF <-
function(x, which=c("auc","ranking","psel") ,showOpt=TRUE, digits=4,
    maxvars=NULL, ...){
  cl <- match.call()
  which <- match.arg(which)
  opar <- par("cex", "pch")
  on.exit(par(opar))
  n <- ifelse(is.null(maxvars),x$Kopt, maxvars)
  par(cex = max(1 - 0.1 * n %/% 10, 0.5), pch = 19)

  switch(which,
        psel={
        if(is.null(x$Psel))
          cat("No Psel information available. See AUCRFcv help.\n")
    else{
      imp <- x$Psel[x$Xopt]
          imp <- sort(imp,decreasing=T)
          if(!is.null(maxvars)) imp <- sort(x$Psel,decreasing=T)[1:
              maxvars]
      if(is.null(cl$pch)) dotchart(imp[length(imp):1],pch=par("pch")
         ,...)
      else dotchart(imp[length(imp):1],...)
      if(is.null(cl$xlab)) title(xlab="Probability of selection")
          }
        },
        ranking={
```

```
    imp <- x$ranking[x$Xopt]
if(!is.null(maxvars)) imp <- sort(x$ranking,decreasing=T)[1:maxvars
   ]
if(is.null(cl$pch)) dotchart(imp[length(imp):1],pch=par("pch"),...)
else dotchart(imp[length(imp):1],...)
if(is.null(cl$xlab)) title(xlab=x$ImpMeasure)
    },
    auc={
            par(opar)
type <- ifelse(is.null(cl$type), "o", eval(cl$type))
pch <- ifelse(is.null(cl$pch), 20, eval(cl$pch))
col <- ifelse(is.null(cl$col), 2, eval(cl$col))

if(is.null(cl$ylim)){
  maxAUC <- max(x$AUCcurve$AUC)
  minAUC <- min(x$AUCcurve$AUC)
  r <- maxAUC-minAUC
  ylim <- c(max(0,minAUC-r), min(1,maxAUC+r))
}
else{
  ylim <- eval(cl$ylim)
}

ylab <- ifelse(is.null(cl$ylab), "OOB-AUC", eval(cl$ylab))
xlab <- ifelse(is.null(cl$xlab), "Number of selected variables",
   eval(cl$xlab))

AUCcurve <- x$AUCcurve

m <- match(c("x","wich","showOpt","digits","maxvars","type", "pch",
    "col", "ylim", "ylab", "xlab"), names(cl), 0L)
plotCall <- cl[-m]
plotCall[[1]] <- as.name("plot")
plotCall$x <- as.name("AUCcurve")
plotCall$type <- as.name("type")
```

```
    plotCall$pch <- as.name("pch")
    plotCall$col <- as.name("col")
    plotCall$ylim <- as.name("ylim")
    plotCall$ylab <- as.name("ylab")
    plotCall$xlab <- as.name("xlab")
    eval(plotCall)

    if(showOpt){
      text(x$Kopt,x$"OOB-AUCopt"+(ylim[2]-ylim[1])/10, paste("OOB-
          AUCopt = ",round(x$"OOB-AUCopt",digits)," (Kopt = ",x$Kopt,")
          ",sep=""), pos=4, cex=0.7, offset=0)
      text(x$Kopt,x$"OOB-AUCopt","|", pos=3)
      if(!is.null(x$cvAUC))
        text(x$Kopt,x$"OOB-AUCopt"+1.5*(ylim[2]-ylim[1])/10, paste("
            cvAUC = ",round(x$cvAUC,digits),sep=""), pos=4, cex=0.7,
            offset=0)
    }
  }
  )
  invisible()
}
```

APPENDIX C

Functions for the identification of high-order interactions using the likelihood-ratio score

trapz and auc.numeric functions

Computes the AUC measure for ROC curves and scores.

```
trapz <- function (x, y){
  idx = 2:length(x)
  return(as.double((x[idx] - x[idx - 1]) %*% (y[idx] + y[idx - 1]))/2)
}

auc.numeric<- function(x,y,clase=1){
  r <- rank(x)
  rd   <- mean(r[y==clase])
  nd <- sum(y==clase)
  nnd <- length(y)-nd
  return((rd-nd/2-0.5)/nnd)
}
```

ROCpart function

Computes optimal ROC for a partition.

```
ROCpart <- function(x, y, eps=1, type=c("auc","roc","lr","rs","all")){
  type <- match.arg(type)
  tab <- table(x,y)
  marginTab <- prop.table(tab+eps,margin=2)
  LR <- marginTab[,"1"]/marginTab[,"0"]
  if(type=="lr") return(LR)
  RS <- LR[as.character(x)]
  ranko <- order(LR,decreasing=TRUE)
  ROC <- apply(marginTab[ranko,],2,cumsum)
  ROC <- rbind(c(0,0),ROC)
  AUC <- trapz(ROC[,"0"],ROC[,"1"])
  switch(type,
     auc = out <- AUC,
```

```
      roc = out <- ROC,
      rs  = out <- RS,
      all = out <- list(AUC=AUC,ROC=ROC,RS=RS,LR=LR)
  )
  return(out)
}
```

Partition and PartitionVar functions

Generates partitions from a variable sets.

```
Partition <- function(p1,p2,index=NULL){
  if(is.null(index)){
    groups <- paste(p2,p1,sep=".")
  }
  else{
    groups <- p2
    groups[index] <- paste(p2[index],p1[index],sep=".")
  }
  x <- as.numeric(factor(groups))
  return(x)
}

PartitionVar <- function(data){
  if(class(data)=="data.frame"){
    partition <- as.numeric(factor(data[,1]))
    for(i in 2:ncol(data)) partition <- Partition(data[,i],partition)
  }
  else{
    partition <- as.numeric(factor(data))
  }
  return(partition)
}
```

OASaucEval and OASaucTest functions

Computes the optimal AUC measure from a subset.

```
OASaucEval <- function(data,y,eps){
  part <- PartitionVar(data)
  LR <- ROCpart(part, y, eps, type="lr")
  LR <- LR[as.character(part)]
  auc <- auc.numeric(LR,y)
  return(auc)
}

OASaucTest <- function(data,y,inbag,oob,eps){
  part <- PartitionVar(data)
  LR <- ROCpart(part[inbag], y[inbag], eps, type="lr")
  LR <- LR[as.character(part[oob])]
  LR[is.na(LR)] <- 1
  auc <- auc.numeric(LR,y[oob])
  return(auc)
}
```

OASnext, OAStep and OAStepRandom functions

Main Opimal AUC algorithm parts implementations.

```
OASnext <- function(data,y,part,eps){
  allpart <- apply(data,2,Partition,p2=part)
  aucs <- apply(allpart,2,ROCpart,y=y,type="auc",eps=eps)
  best <- which.max(aucs)
  aucmax <- aucs[best]
  newvar <- colnames(data)[best]
  newpart <- as.numeric(factor(allpart[,best]))
  new <- list(part=newpart, auc=aucmax, addvar=newvar)
  return(new)
```

```
}

OAStep  <- function(var1,data,y,eps,inbag=NULL,maxvars=15){
  if(class(data)!="data.frame"){
    data <- as.data.frame(data)
  }
  if(is.null(inbag)){
    flag.test <- FALSE
    inbag <- 1:nrow(data)
  }
  else{
    flag.test <- TRUE
    switch(class(inbag),
      "character" = { indnames <- rownames(data)
                  oob <- indnames[!indnames %in% inbag] },
        "integer" = { n <- nrow(data)
                    oob <- (1:n)[-inbag] },
      stop("inbag argument not valid")
    )
  }

  vars <- rep(0,ncol(data))
  names(vars) <- colnames(data)
  iter <- 1
  vars[var1] <- 1
  part <- Partition(data[inbag,var1],p2=1,index=NULL)
  auc <- ROCpart(part,y[inbag],eps,type="auc")
  testauc <- "n/d"
  if(flag.test) testauc <- OASaucTest(data[,var1],y,inbag,oob,eps)
  while(sum(vars>0)<maxvars){
    iter <- iter + 1
    newstep <- OASnext(data[inbag,vars==0],y[inbag],part=part,eps)
    vars[newstep$addvar] <- iter
    part <- newstep$part
    auc[iter] <- newstep$auc
```

```
    if(flag.test) testauc[iter] <- OASaucTest(data[,vars!=0],y,inbag,
        oob,eps)
  }

  return(list(auc=auc,testauc=testauc,vars=names(sort(vars[vars!=0]))))
}

OAStepRandom <- function(data,y,eps,inbag=NULL,maxvars=15){
  if(class(data)!="data.frame"){
    data <- as.data.frame(data)
  }
  if(is.null(inbag)){
    flag.test <- FALSE
    inbag <- 1:nrow(data)
  }
  else{
    flag.test <- TRUE
    switch(class(inbag),
      "character" = { indnames <- rownames(data)
                  oob <- indnames[!indnames %in% inbag] },
        "integer" = { n <- nrow(data)
                    oob <- (1:n)[-inbag] },
      stop("inbag argument not valid")
    )
  }

  vars <- rep(0,ncol(data))
  names(vars) <- colnames(data)
  rvar <- sample(names(vars),1)
  part <- as.numeric(data[inbag,rvar])
  iter <- 1
  auc <- ROCpart(part, y[inbag], eps,"auc")
  vars[rvar] <- iter
  testauc <- "n/d"
  if(flag.test) testauc <- OASaucTest(data[,vars!=0],y,inbag,oob,eps)
```

```
  while(sum(vars>0)<maxvars){
    iter <- iter + 1
    newstep <- OASnext(data[inbag,vars==0],y[inbag],part=part,eps)
    vars[newstep$addvar] <- iter
    part <- newstep$part
    auc[iter] <- newstep$auc
    if(flag.test) testauc[iter] <- OASaucTest(data[,vars!=0],y,inbag,
        oob,eps)
  }
  return(list(auc=auc,testauc=testauc,vars=names(sort(vars[vars!=0]))))
}
```

FRutePurge function

Pruning the optimal subset.

```
FRutePurge <- function(rutaauc,kperms,Dades,y,inbag,eps,arperm=TRUE,
    alpha=0.05,n){
  nvars <- length(rutaauc)
  rutasnps <- names(rutaauc)
  deleting <- TRUE
  historia <- NULL
  while(deleting & nvars>1){
    if(rutaauc[nvars]<rutaauc[nvars-1]){
      historia <- cbind(historia,c(nvars,1,0,0))
      nvars <- nvars-1
      next
    }
    if(!arperm) return(rutaauc)
    auxperm <- NULL
    for(i in 1:kperms) auxperm <- cbind(auxperm,sample(Dades[inbag,
        rutasnps[nvars]]))
    allpart <- apply(auxperm,2,Partition,p2=PartitionVar(Dades[inbag,
```

```
        rutasnps[1:(nvars-1)]]))
    aucs <- apply(allpart,2,ROCpart,y=y[inbag],type="auc",eps=eps)
    permaucrutes <- sapply(aucs,function(a) c(rutaauc[1:(nvars-1)],a))
    permars <- apply(permaucrutes,2,trapz,x=c(0:(nvars-1))/(nvars-1))
    arthres <- mean(permars) + sd(permars) * qnorm((1-alpha)^(1/n))
    rutaar <- trapz(x=c(0:(nvars-1))/(nvars-1),y=rutaauc[1:nvars])
    if(rutaar<=arthres){
      historia <- cbind(historia,c(nvars,2,mean(permars),sd(permars)))
      nvars <- nvars-1
      next
    }
    historia <- cbind(historia,c(nvars,0,mean(permars),sd(permars)))
    deleting <- FALSE
  }
  return(list(rutaauc[1:nvars], historia))
  return(rutaauc[1:nvars])
}
```