UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

A new statistical framework for testing association of genetic variants within Highlander.

Dissertation presented by
Simon BOUTRY

for obtaining the Master's degree in
Mathematical Engineering

Supervisor(s)
Pierre-Antoine ABSIL, Miikka VIKKULA

Reader(s)
Raphaël HELAERS, Emilie RENARD

Academic year 2016 - 2017

I would like to dedicate this master thesis to my mother Pascale who did everything for me and made me the way I am today, to my family and all my friends for their unconditional support and all the time that I had the chance to share with them.

Acknowledgements

The completion of this master thesis would not have been possible without the participation, the trust and support received by so many people over the past years. Even though it would be impossible to name all the people I would like to thank, I would like to dedicate a special attention and thank to

all the members of the GEHU team. Miikka Vikkula for having me in his team. Raphael Helaers for all his time, explanations and help. Nisha Limaye for her trust and the work that she entrusted to me. Pascal Brouillard for all his explanations. Mustapha Amyere for his collaboration and enthusiasm. Pierre-Antoine Absil for accepting my proposition of thesis and for being my promoter. Emilie Renard for her constant help throughout the year. Sophie Lucas for having me in the first place at de Duve Institute. Cécile Delannay for all the work she is doing and that she did for me. Jess and Powpow for their precious corrections. My mathematical teacher of last year in Highschool who told me I would never be good at maths and could not succeed at university.

Abstract

The purpose of this master's thesis is the implementation of a new statistical framework based on the SKAT package, developed at Harvard, supporting the main class of association tests that have been developed nowadays (Burden, SKAT and SKATO). The framework has been developed in order to be used within Highlander, a software developed at GEHU lab that centralizes all their variant data and annotations from NGS (Next Generation Sequencing, mainly Whole Exome Sequencing). The purpose of this framework will be to fasten and guide the investigator through the identification of genetic regions (e.g. genes) which are most likely to be involved in rare diseases.

Table of contents

List of figures **xiii**

List of tables **xix**

Nomenclature **xxi**

1 Introduction **1**

2 Genomics : understand the origin of the data **5**

2.1 DNA and its components 5

2.1.1 Nucleotides and nucleic acids 6

2.1.2 Amino acids, genes and proteins 6

2.1.3 Chromosome, allele and zygosity 7

2.1.4 From genotype to phenotype 8

2.1.5 DNA structure and replication 8

2.1.6 Genetic variation, variant, polymorphism and mutation 9

2.2 Next Generation Sequencing 11

2.2.1 DNA collection and conservation 12

2.2.2 Polymerase chain reaction 12

2.2.3 Sequencing technology 13

2.2.4 Post sequencing : alignment, variant calling and annotations 15

2.3 Highlander : a genomic database 16

2.3.1 Database 16

2.3.2 Data sources 16

2.3.3 Variant filtering made easy 18

2.3.4 Make mutation identification easier 18

3 An overview of dedicated statistical test **19**

3.1 Single-Variant Tests 20

3.2 Region-Based Aggregation Tests 20
3.2.1 Burden Tests 22
3.2.2 Adaptive Burden Tests 23
3.2.3 Variance-Component Tests 24
3.2.4 Omnibus Tests 25
3.2.5 Others methods 26
3.2.6 Summary of methods 27

4 Methods **29**
4.1 the SKAT package 29
4.1.1 Supported classes of tests 30
4.1.2 Different choices of kernel 31
4.1.3 P values computation 35
4.1.4 An optimal adjustment : the SKATO test 36
4.1.5 Advantages of SKAT package 37
4.2 Rare variants association studies design for GEHU laboratory 40
4.2.1 Data retrieval and case control design 40
4.2.2 Specific Genotype matrix construction with our *creatGenoMatrix* function 42
4.2.3 Analysis step 43

5 Results **47**
5.1 A global review and characterization of the available tests 48
5.1.1 Testing the kernels 49
5.1.2 Testing the p values computation methods 51
5.1.3 A global tests characterization 52
5.1.4 Summary and recommendations 52
5.1.5 A validation case: the *CM / CLP* association study 54
5.2 A real data application 56

6 Conclusions **59**

References **63**

Appendix A Supplementary informations **69**
A.1 Nucleotides and nucleic acids 69
A.2 Amino acids 70
A.3 Illustration of a chromosome and its inner structure 70

A.4 Classical representation of a DNA molecule . . . 71
A.5 Polymerase chain reaction . . . 72
A.6 Next generation sequencing evolution . . . 74
A.7 Comparison of error in sequencing technologies . . . 74
A.8 Illustration of a BAM file . . . 75
A.9 Illustration of a VCF file . . . 75
A.10 Partial list of tools and resources to annotate DNA sequence variants . . . 76
A.11 List of Software Packages for Rare-Variant Association . . . 77
A.12 Portion of coding for 2wayIX kernel . . . 78
A.13 Little mistake in SKAT package tutorial . . . 79
A.14 Illustration of the small-sample adjustment method . . . 80
A.15 Illustration of raw data retrieval from Highlander . . . 81
A.16 Illustration of our function *creatGenoMatrix* . . . 81
A.17 Illustration of a possible value for our input parameter *variant* . . . 82
A.18 Detailed description of the input parameter of *creatGenoMatrix* function . . . 82
A.19 A brief explanation on each test's name meaning . . . 83

Appendix B Results of section 5.1 : a global review and characterization of the available tests 85
B.1 A global review and characterization of the available tests . . . 85
B.1.1 Testing the kernels . . . 85
B.1.2 Testing the p value computation methods . . . 88
B.1.3 A global tests characterization . . . 92
B.1.4 Summary and recommendations . . . 102
B.1.5 A validation case : the *CM / CLP* association study . . . 104

Appendix C Results of section 5.3 : a real data application 107
C.1 Summary and context of current investigation . . . 108
C.1.1 Abstract . . . 108
C.1.2 Confidential excerpt from paper . . . 108
C.2 Computational analysis . . . 109
C.3 P value analysis . . . 110
C.4 Comparing the ranking between the different class of tests . . . 110
C.5 Comparing the ranking between the different experiments . . . 115
C.6 Summary of all experiments . . . 119
C.7 Pair enrichment testing . . . 119
C.7.1 Excel file description . . . 120

List of figures

2.1 Schematic representation of the process of DNA transcription and alternative splicing. The light blue part corresponds to introns, and the 3 proteins are isoforms. The figure is drawn from [7] 9
2.2 Illustration of the process to obtain DNA sequence data and some of the common experimental source of error. The figure is drawn from [27] 11
2.3 Illustration of the post-processing steps in order to construe the data obtained via sequencing technologies. The figure is drawn from [27] 15
2.4 Architecture of Highlander : output files produced by the primary pipeline are stored in a server, variant calling being also imported in the Highlander database, and annotated by a large variety of sources. Highlander graphical user interface clients connect to the Highlander database and provide end-users a comprehensive set of tools and functionalities to analyze variants. Figure drawn from [55] 17

3.1 Summary of Statistical Methods for Rare-Variant Association Testing, table drawn from [62]. 28

5.1 Summary table of the main observations resulting from both the output value analysis and the computational time analysis for each class of tests. 50
5.2 Summary table of the main observations resulting from both the output value analysis and the computational time analysis for each class of tests. 51
5.3 Summary table of the main observations resulting from both the output value analysis and the computational time analysis for each class of tests. 52
5.4 Summary table of all tested different implementations. The value represents the performance of each test on the biggest size input in the last stage of the computational time analysis performed above 53

A.1 Detailed representation of DNA molecule. P stands for phosphate group, S stands for the sugar and the four nucleotides are respectively represented by the letters A, T, C, G. The figure is drawn from http://malemodelspicture.net/ 69
A.2 The 20 common amino acids of proteins. The figure is drawn from [57]. . . 70
A.3 From left to right : schematic representation of DNA double helix structure, DNA and histones, chromatine, supercoiled DNA and chromosome. The figure is drawn from http://bkmn.tk/chromosome-structure-diagram/ 70
A.4 Classical representation of DNA molecule in its common form. The four nucleotides are respectively represented by the letters A, T, C, G. The figure is drawn from [24] 71
A.5 Illustration of the polymerase chain reaction cycles for the amplification of a particular DNA sequence. Primers are represented in green, original template of DNA to be duplicated in blue and copies of the latter in pink. The figure is drawn from [9] 73
A.6 Next Generation Sequencing methods evolution time line. Figure drawn from [30] 74
A.7 Comparison of error rates and error types for assembled paired-end MiSeq reads and unidirectional reads from the Ion Torrent PGM platform derived from a mock-community mixture of 20 bacterial DNAs. Errors for single nucleotide substitutions, homopolymer indels (homoindels), other indels outside homopolymer tracts, and compound errors (event involving two or more categories) are shown separately (figure drawn from [59]) 74
A.8 Illustration of a VCF file, figure drawn from https://insidedna.me/tutorial . . 75
A.9 Illustration of a VCF file, figure drawn from https://wiki.nci.nih.gov/ 75
A.10 Non exhaustive list of prediction programs drawn from [2] 76
A.11 List of software packages for rare variant association drawn from [62] . . . 77
A.12 Portion of coding for 2wayIX kernel, the latter is taken from the function Kernel.R within the SKAT package. The code of function Kernel.R can be found at https://github.com/leeshawn/SKAT/blob/master/R/Kernel.R 78
A.13 Screen shot of an *R* session to prove little mistake in SKAT package tutorial 79
A.14 $-\log_{10}$ Q-Q plots of observed versus expected p values for the ALI exome sequence data for six methods : burden tests (linear (**N**) and weighted (**W**)), SKAT, SKATO, adjusted SKAT and adjusted SKATO. The x axis represents $-\log_{10}$ expected p values, and the y axis represents $-\log_{10}$ observed p values. A total of 6488 genes with at least four rare variants were tested for associations with ALI severity 80

A.15 Screen shot of the 10 first rows of raw data retrieval for patient for the real data application with consensus prediction greater or equal to 2 and *cadd_phred* greater or equal to 15 and a restricted group of control individuals. 81
A.16 Screen shot of the header of our function *creatGenoMatrix* responsible for the creation of the genotype matrix. 81
A.17 Screen shot of a possible value for input parameter *variant* of our function *creatGenoMatrix*. 82

B.1 Plot of the computational time for different square genotype matrix dimension (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available kernels and the same p value method computation. 86
B.2 Plot of the p values for different phenotype vectors (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available kernels and the same p value method computation. 86
B.3 Output value analysis : p value summary table for each tests and each experiment 86
B.4 Output value analysis : time value summary table for each tests and each experiment 87
B.5 Computational time analysis : p value summary table for each tests and each experiment 87
B.6 Computational time analysis : time value summary table for each tests and each experiment 87
B.7 Plot of the computational time for different square genotype matrix dimension (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available p value method computation and the same kernel. 88
B.8 Plot of the p values for different phenotype vectors (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available p value method computation and the same kernel. 89
B.9 Output value analysis : p value summary table for each test and each experiment 89
B.10 Output value analysis : time value summary table for each test and each experiment 90
B.11 Computational time analysis : p value summary table for each test and each experiment 90
B.12 Computational time analysis : time value summary table for each test and each experiment 91

B.13 Plot of computational time (with y axis going from 0 to 500 seconds) for different square genotype matrix dimensions (x axis) for all kinds of tests (burden, SKAT and SKATO) implementing all the available combinations of kernels and p value method computations in order to highlight the huge diversity of tests. 93
B.14 Plot of the p values (on the left), and computational time (on the right, with y axis going from 0 to 30 seconds) for different case-controls (x axis) for all kinds of tests (burden, SKAT and SKATO) implementing all the functional combinations of kernels and p value method computations in order to highlight the huge diversity of tests. 94
B.15 Plot of the p values (on the left), and computational time (on the right, with y axis going from 0 to 30 seconds) for different case-control (x axis) for all kind of tests (burden, SKAT and SKATO) implementing all the functional combinations of kernels and p value method computations in order to highlight the huge diversity of tests. 95
B.16 Output value analysis : p value summary table for each test and each experiment 96
B.17 Output value analysis : time value summary table for each tests and each experiment 97
B.18 Computational time analysis : p value summary table for each test and each experiment (part 1) 98
B.19 Computational time analysis : p value summary table for each test and each experiment (part 2) 99
B.20 Computational time analysis : time value summary table for each test and each experiment (part 1) 100
B.21 Computational time analysis : time value summary table for each test and each experiment (part 1) 101
B.22 Summary of the 8 categories of test across the 3 class of tests. Within each category there is a recommended test (in green), and some functional and computationally fast (in blue), and functional but time consuming tests (in yellow). 102
B.23 Table of the p values obtained for each test and each gene. Here the filters are activated along with $exac\_af < 0.05$ criteria and $gonl\_af < 0.05$. 105
B.24 Table of the p values obtained for each test and each gene. Here the filters are activated along with $exac\_af < 0.01$ and $gonl\_af < 0.01$. 105
B.25 Table of the p values obtained for each test and each gene. Here without any filters. 105

B.26 Table of the p values obtained for each test and each gene. Here the only filtering options are *exac_af* < 0.05 and *gonl_af* < 0.05. 105

C.1 Summary of the average and total computational time (computed on the 93 tests) required for each test in the basic experiment (first set of filters) . . . 110
C.2 Summary of the minimum, maximum and standard deviation based on the 93 p value computed by each test on each gene for the basic experiment (using first set of filters) 110
C.3 Plot of the average ranking (y axis) based on the burden tests for each gene (x axis) for the basic experiment (using first set of filters) 111
C.4 Plot of the average ranking (y axis) based on the SKAT tests for each gene (x axis) for the basic experiment (using first set of filters) 112
C.5 Plot of the average ranking (y axis) based on the SKATO tests for each gene (x axis) for the basic experiment (using first set of filters) 113
C.6 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the basic experiment (using first set of filters) 114
C.7 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the basic experiment (using second set of filters) 115
C.8 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the basic experiment (using third set of filters) 116
C.9 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the restricted experiment (using second set of filters) 117
C.10 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the restricted experiment (using third set of filters) 118
C.11 Table showing the number of genes that have passed a tes with a pvalue $< 5\%$, per tests and also per experiment. Last row shows the total time taken per experiment. 119

List of tables

Nomenclature

Acronyms / Abbreviations

A Adenine

BAM Binary Alignment Map

bp Base Pair

CAST Cohort Allelic Sums Test

C Cytosine

CLP Cleft Lip and/or Palate

CM-AVM Capillary Malformation-Arteriovenous Malformation

CMC Combined and Multivariate Collapsing

DNA Deoxyribonucleic acid

dNTPs Deoxynucleotide Triphosphates

EC Exponential-Combination

EMMPAT The Evolutionary Mixed Model For Pooled Association Testing

EREC The Estimated Regression Coefficient

GEHU Human Molecular Genetics

G Guanine

GUI Graphical User Interface

GWAS Genome Wide Association Study

IBS Identity By State

KBAC Kernel-Based Adaptive Cluster

MAC Minor Allele Count

MAF Minor Allele Frequency

mRNA messenger Ribonucleic acid

NGS Next Generation Sequencing

OR Odds Ratio

PCR Polymerase Chain Reaction

RNA seq RNA sequencing

RNA Ribonucleic acid

SKATO Optimal Sequence Kernel Association Test

SKAT Sequence Kernel Association Test

SNP Single Nucleotide Polymorphism

SNV Single Nucleotide Variant

SSU The Sum of Square Score

T Thymine

U Uracil

VCF Variant Call Format

VT Variable Threshold

WST Weighted-Sum Test

Chapter 1

Introduction

Since the beginning of 2005, genome-wide association studies (GWASs) have been broadly used in order to discover the genetic architecture of complex diseases and quantitative traits (see chapter 2). Investigations of the genetic origin of human diseases had usually relied on oversimplified but somewhat useful dichotomy between rare (monogenic and simple) and common (multigenic and complex) diseases [3]. However, there is an important difference in our understanding between these 2 categories. Even though GWAS already allowed the identification of more than 1000 genetic variations linked with many diseases [69], it is not the case for rare disease. This is due to the fact that usual strategies used for gene discovery have limited power in this situation (small samples of patients, genetic origin might be distributed over several small changes in the patient's genetic code that are more difficult to detect because individually, they don't have substantial impact, etc). Nevertheless, there are more than 5000 rare monogenic diseases and for half of these, the underlying genetic model is still unknown [12]. Moreover, an important part of diseases previously thought as common (classified as common using the old classical dichotomy; see above) are now being considered as a heterogeneous collection of rare monogenic disorders which, again, are not known [43]. All this supports the great importance of developing new tools able to handle non classical common diseases and of creating a new study design that overcomes all the limitations encountered by the classical GWAS studies.

Thanks to recent advances in sequencing technologies, gene identification strategies have drastically changed (see chapter 2). The emergence of this Next Generation Sequencing (NGS) had changed the way genetic information is looked at and how it is analyzed (see chapter 2). For example, now, as sequencing machines enable to identify so many genetic information, strategies need to be carefully set up in order to efficiently and robustly prioritize (see chapter 2) the required information (e.g pathogenic changes of the genetic code, or

the part of it that might play a role in the disease) [12]. More importantly, these NGS technologies allow for the first time to handle the desired rare genetic information [62].

By taking into account all the factors that acted as a limitation for the classical GWAs studies, but also considering all the particularities of rare disease and relying on the newly created NGS technologies, several brand new rare association studies and tests have been proposed (e.g Burden tests, variance-component tests, Optimal Sequence Kernel Association Test (SKATO), etc.). They overcome the need of big sample size or effect size and do not require multiple test corrections as the classical single variant statistical test did (see chapter 3).

This master thesis was carried on at de Duve institute which is a multidisciplinary biomedical research institute hosting several laboratories of the faculty of medicine of the Catholic University of Louvain as well as the Brussels branch of the Ludwig Institute[1]. More precisely, this work was designed within the Human Molecular Genetics (GEHU) laboratory. The latter studies the genetic and molecular causes of human disease, with a focus on vascular anomalies, lymphedema, cleft lip and palate, cancers and hypermobility (which falls into the rare diseases classical classification). Because the molecular mechanism of many are still unknown, the goal of GEHU research is to discover the genetic variations that enable them to ultimately come up with an adequate treatment for the patient. This is why GEHU uses cutting-edge genetic techniques and NGS technologies on patient samples for gene-discovery. Therefore, the goal of this work is to bring clear and useful tools which would fasten and improve that first but crucial identification step via a dedicated association test (see chapter 3). It is important to keep in mind this context, because it will have an impact on the choice made through the research and implementation.

This work is the result of a one year investigation that started with a global review of the literature which ended up with the choice of a particular software package called SKAT[2] developed at Harvard by Ionita-Laza I, Lee S, WU M C et al. [68, 69, 33, 32, 22]. Based on the later, we present for the first time a deep analysis and characterization of each supported test, summarized by some guidelines and recommendations and a practical implementation of a new statistical framework designed for the GEHU specific questions.

This paper is organized as follow : first, some basic and necessary notions of genomics will be exposed (chapter 2). This is necessary in order to be able to understand the data

[1] https://www.deduveinstitute.be/fr
[2] https://www.hsph.harvard.edu/skat/

behavior and all the possible sources of errors. It is easy to use a statistical test and obtain results. But in order to get meaningful results and be able to fully understand them, one needs to understand the origin and meaning of the inputs. It will also be the opportunity to cover NGS topics and the pipeline used to retrieve the data. Last but not least, it ends up with a description of the in-house built software Highlander present at GEHU laboratory in which this work has to be integrated.

Chapter 3 will be a classical state of the art summarizing the available association test already developed.

Chapter 4 can be seen as the results of the investigation and reading phase but it has also been greatly influenced by the results obtained. It will focus on a particular choice (as mentioned above, the SKAT package) in which we proposed a clear and simple description of the supported options. This will also be the opportunity to explain this particular choice given the specific context of this master thesis. The chapter will end up with our new framework and its particular implementation designed for the GEHU laboratory.

Chapter 5 will present our different results and will also be the opportunity to perform real data application within an actual gene discovery study.

Finally, chapter 6 will conclude on the main results and clearly expose the main contributions of this paper. But more importantly, it will be the opportunity to expose a number of proposals and ideas for future research.

Chapter 2

Genomics : understand the origin of the data

This chapter will go through a global introduction of genomics with a focus on the main aspects underlying next generation sequencing [1]. It will start with a simple introduction of the main object under study, namely Deoxyribonucleic acid (DNA). Its composition, how it is produced and used in the sequencing process. To go one step further into the understanding of genomics data, one needs to go through a review of the existing sequencing processes and technologies. Then, the post sequencing step consisting of handling all the data collected for the sake of crucial information retrieval (e.g. information about a patient's disease origin, specific function of a gene, etc.) needs to be fully understood. Last but not least, Highlander software is introduced. At then end of this chapter, the reader should understand the high complexity of the data and the need of a new tool in order to guide the investigator in the identification process.

2.1 DNA and its components

This section will review the main input of our framework, namely genetic information, and more precisely DNA. The description will start from the smallest building block (nucleotides and nucleic acids) in order to introduce bigger structure after (e.g. genes, chromosome) and present the origin of crucial information that will be use in the tests. After that, the process of DNA replication and the origin of genetic variation are introduced. It must be understood

[1] As in all application field of mathematical engineering, it is necessary to understand the input one is dealing with. Therefore, this chapter is addressed to readers with possibly no prior knowledge of this particular field for them to be able to understand better the data under study and get more insight in the following chapters.

that the little change in the DNA constitute the starting point of the analysis proposed later on.

Nearly all cells in the human body contain DNA bundled in their nucleus (except for mature red blood cells [13] and cornified cells situated in the skin, hair and nails, which contain no nucleus). As previously said, the latter constitutes the main input in genomics. Ribonucleic acid (RNA) molecules, and subsequently polypeptides of protein, can be synthesized [64] thanks to the genetics information stored in DNA molecules. Let's start with the smallest building blocks of DNA, before going through bigger structures, in order to fully understand what DNA exactly is.

2.1.1 Nucleotides and nucleic acids

The description of the data starts with the smallest building block. It is important to get that even a small change in one of them can cause a disease. Nucleotides are involved in nearly all biological processes in the body [52]. These are the components of DNA and RNA. Because they regulate many metabolic functions [2], they represent a fundamental molecular class in genetics. A nucleotide (see bottom frame in figure A.1) is made of 3 these main elements: a phosphate group, a sugar and a purine or pyrimidine base [52].

The phosphate (see black in figure A.1) group can be seen as the backbone of DNA molecules. It can have several forms (isolated, by two or even three, respectively mono, dipho and tri) and is linked to the sugar molecule. The latter (see in blue in figure A.1) is generally derived from a ribose (in ARN) or desoxyribose (in DNA) [64]. The last element is split into two categories [51]. There are two purine bases, namely adenine (denoted by A in green in figure A.1) and guanine (denoted by G in black in figure A.1), and three pyrimidine bases, namely thymine (denoted by T in red in figure A.1), cytosine (denoted by C in blue in figure A.1) and uracil (denoted by U). The latter pyrimidine base is only present in RNA molecules and replaces the thymine base (that are only present in DNA molecules) [64]. It will be shown later on that a unexpected change in the purines or pyrimidines bases constitute the crucial information collected in order to form the input of the statistical test.

2.1.2 Amino acids, genes and proteins

Amino acids are made of the nucleotides. Moreover, they can be represented by their 3 abbreviations letter, as shown in figure A.2. Each amino acid has a precise number of

[2]The whole range of biochemical processes that occur within any living organism, http://www.medicinenet.com/, consulted on 02/03/2017

predefined sequence of nucleotides. Union of numerous amino acids, linked by peptide bonds, form polypeptide chains.

Genes are specials units of chromosomal DNA. They can be defined as a DNA's segment responsible for the formation of a polypeptide [52]. Therefore, one gene corresponds to one polypeptide. In eukaryotic, genes are composed of coding (exons) and non-coding (intron) part.

Proteins take part in nearly all chemical reactions in living organisms. Their importance is confirmed as they act as an enzyme[3], catalyzing[4] chemicals reaction in living cells [52]. Proteins are formed by one or several polypeptides. This explains why more than one gene can be involved in a protein synthesis. Each protein has a well defined amino acids sequence. The latter contains important information about the function and origin of the protein. Each protein has its own particular physicochemical properties given by its linear amino acids series and its tridimensional structure [52]. It should already be clear to the reader that a small change in a nucleotide (basic building block) can have a impact on a particular protein which in turn can cause a disease.

2.1.3 Chromosome, allele and zygosity

In human cells, DNA is actually wrapped around several chromosomal proteins called histones [58]. This forms the chromatin. This chromatin constitutes a supercoil DNA which in turn allows to build the chromosome (see figure A.3). A chromosome is a thread-like structure of nucleic acids and protein, carrying genetic information in the form of genes[5]. Each chromosome is composed of one single DNA molecule containing approximately from 200 (chromosome Y) up to 3000 (chromosome 1) genes [52] [6].

The human genome is structured into 23 chromosome pairs (22 pairs of autosomes[7] and one pair of allosome[8]) [64]. These 46 DNA molecules are contained in each cell's nuclei. Members of a pair of chromosomes (called homologous chromosome) carry matching genetic information, that is to say that they have the same gene in the same sequence. [58]. However,

[3]Enzymes are usually protein molecules with a characteristic sequence of amino acids that folds to produce a specific three-dimensional structure, which gives the molecule unique properties. Another molecule with catalytic activity is ribozyme, an enzyme made of RNA rather than protein, http://www.biology-online.org/dictionary, consulted on 02/03/2017

[4]To speed up a process, especially a chemical or biochemical reaction, http://www.biology-online.org/dictionary, consulted on 02/03/2017

[5]Definition from Oxford dictionary, https://en.oxforddictionaries.com/ consulted on 1/12/2016

[6]National Center for Biotechnology Information (US). Genes and Disease [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 1998-. Chromosome Map, consulted online on 20/02/2017

[7]An autosome is a chromosome that is not an allosome

[8]An allosome is a sex chromosome

at any particular locus[9], they may have either identical or slightly different forms of the same gene, called alleles [58]. The latter will be the most important information to be taken into account as input. In each pair of chromosome, one of them is inherited by the father, and the other one by the mother. Zygosity can be defined as the degree of similarity between alleles at a particular locus. The three main classes[10] are heterozygous and homozygous reference and homozygous alternative. The first one describes the case in which the alleles are different for a particular gene, while the second and third ones are situations in which the alleles are the same but have either the reference (dominant) form or the alternative (recessive) form. Therefore, in order to account for genetic variation, it will be shown that available test make extensive use of alleles and their zygosity.

2.1.4 From genotype to phenotype

The genotype can be defined as the genetic constitution of an individual organism or as the types of alleles found at a locus in an individual [64]. The genotype is distinct from the expressed features, or phenotype. The latter is the physical characteristic of a cell or organism, as determined by the genetic constitution [64]. There is a fundamental distinction between genotype and phenotype because there is no one-to-one correspondence between genes and traits. Indeed, most complex traits (e.g. hair color) are influenced by many genes. Most traits are also influenced by the environment. This means that the same genotype can result in different phenotypes, depending on the environmental contribution, and likewise, that the same phenotype can result in more than one genotype[16]. While studying genomics, the objective is to understand how the genotype is linked to the phenotype of an individual or a group of individuals. Therefore association tests will use as main input the phenotype and the genotype (via the different alleles zygosity).

2.1.5 DNA structure and replication

DNA has a well known double stranded helix structure[11] as shown in figure A.4. This structure explains two aspects of DNA : replication and genetic information transfer. Chemical structures of nucleotide dictate DNA's spatial bond [52]. Therefore, inside DNA's helix structure, adenine will always be in front of a thymine, and guanine in front of a cytosine. As a

[9] particular position on a chromosome

[10] Here, zygosity is defined in a bioinformatic way, indeed this will be clearer in chapter 3 section 3.1.2. For biological definition of zygosity, the reader can refer to [64, 58]

[11] For your information, DNA can have several conformations, DNA-B, DNA-Z and DNA-A. The second one is the most common form in nature [64, 52]

consequence, a nucleotide sequence of one DNA strand allows the determination of the other strand. Base specific matching is the most important DNA's structural characteristic[52].

Genetic information lies in nucleotides base pairs sequence[64, 52]. As said before, each 3 base pair stand as a code (called codon) for an amino acid [65]. Therefore, a codon sequence determines an amino acids sequence, Which in turn, form a polypeptide (gene's product).

The way DNA is copied important because it can explain one way genetic variation can be introduced. Indeed, nucleotide base sequence is first copied (transcription) based on one of the two DNA's strands into another information carrying a molecule called pre-mRNA (messenger RNA). The latter is mature mRNA's precursor. In order to get final mRNA's form, non-coding sequences present in pre-mRNA are removed. An important step, called splicing, consists of assembling all the exons. Exons are spliced in a variety of different ways to produce a set of different mRNA, thereby allowing a corresponding set of different proteins (called isoforms) to be produced from the same gene [1]. The latter is illustrated in figure 2.1.

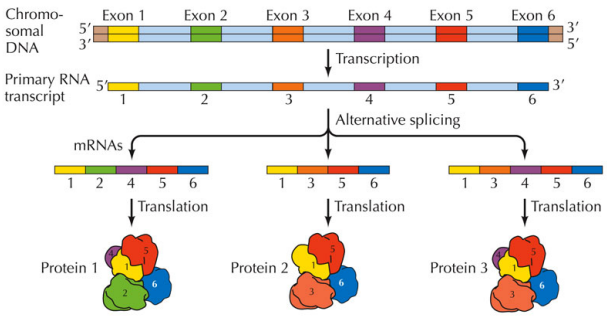


Fig. 2.1 Schematic representation of the process of DNA transcription and alternative splicing. The light blue part corresponds to introns, and the 3 proteins are isoforms. The figure is drawn from [7]

After that, the mRNA nucleotide base sequence is used as a matrix in order to create an amino acids sequence corresponding to the codon sequence (translation). For eukaryotic cells, DNA replication can start at different point called replicons.

2.1.6 Genetic variation, variant, polymorphism and mutation

It is well known that the nucleotide sequence in the DNA of any 2 human being is 99.9% identical [58]. Therefore, the very small portion of DNA that differs is of great interest because it must be responsible for explaining all differences between individuals (brown or blond hair, affected by cancer or not, etc.).

A variant is defined as any change in the nucleotide sequence or arrangement of DNA [58]. It can be uniquely identified thanks to 4 information, namely, the chromosome and the position number together with the reference and alternative forms (it can be seen in figure A.15, columns *chr*, *pos*, *reference* and *alternative* allow to uniquely identify each variant).

These elements can be described as [18] :

- *chr* and *pos* : chromosome and position give the exact position where the variant is.
- *reference* : corresponds to the reference base that varies in the population. Note that it is always given in the forward strand.
- *alternative* : corresponds to the alternative base that varies in the population. Note that it is always given in the forward strand.

Genetic variation can occur at three different levels : variation that affects the number of chromosomes in the cell (genome variation), variation that modifies the structure of an individual chromosomes (chromosome variation) and variation that changes an individual genes (gene mutation) [58]. Only the latter type of variation will be of interest for this paper.

Gene variants can be introduced during DNA replication or because the system fails to repair DNA after damage [58]. They also can be introduced by chemical agents (called mutagen) or radiation. They represent a link between heredity and environment [52].

In principle, there are three types of punctual variants (involving one nucleotide) : replacement of a base-pair by another (substitution), loss of base-pair (deletion), and introduction of a new base-pair (insertion). There are two kinds of substitution : replacement of a pyrimidine by the other pyrimidine, or replacement of the purine by the other one (transition) and replacement of a purine by a pyrimidine or conversely (transversion) [52]. This kind of variant might change the codon, and therefore produce the wrong amino acid. A nucleotide substitution that results in an amino acid change is called " a missense variant" [64]. When deletion or insertion is not a multiple of three (i.e, it is not an integral number of codon) and it occurs in the exon, the reading frame is altered. The resulting variants are called frameshift variants [58]. Finally, a point variant in the sequence that transforms the normal coding codon into one of the three stop codons (see figure A.2) is called "nonsense variant" [58]. Variants also differ by their frequency (proportion of people having the same variant in the population). An important intuitive statistic is the so called minor allele frequency (MAF) (broadly used in genetics) which can be defined as the frequency of the less (or least) frequent allele in a given locus and a given population [70]. By convention, rare variants are present in less or equal to 1% (equivalently MAF $\leq 1\%$) of the population while common variants are present in more than 1% (MAF $> 1\%$).

Polymorphism is defined as the presence in a population of two or more relatively common forms of a gene, chromosome or genetically determined trait [9]. It can be viewed as the existence of a variant. The simplest and most common of all polymorphisms are single nucleotide polymorphisms (SNPs) [58]. This kind of variant only differs from one single nucleotide at one given position. They are frequent and occur on average once every 1000 base pairs [52, 58]. By convention, a variant is called "common" when it is found in more than 1% of chromosomes in the general population. In contrast, alleles with frequencies below 1% are called "rare variants" [58]. There is no simple correlation between allele frequency and the effect of the allele on health.

By convention, a mutation is a variant that is likely responsible for causing a disease. This work will focus on rare (SNPs) variants. But the reader should keep in mind that there exists several others forms of genetic variation that might explain the origin of the disease.

This section has introduced all the building block and entities behind the data.

2.2 Next Generation Sequencing

This section will go through all the pipelines used to collect the data (from the raw data up to its final form) as illustrated in figure 2.2. The way DNA of a patient is collected, conserved and the process is is going through in order to be useful are critical. Indeed they all introduce errors that cannot be ignored. After that, the sequencing technology that has the role to transform the genetic information into informatics data will be explored. The last part of this section focus on how the data collected are then enriched.

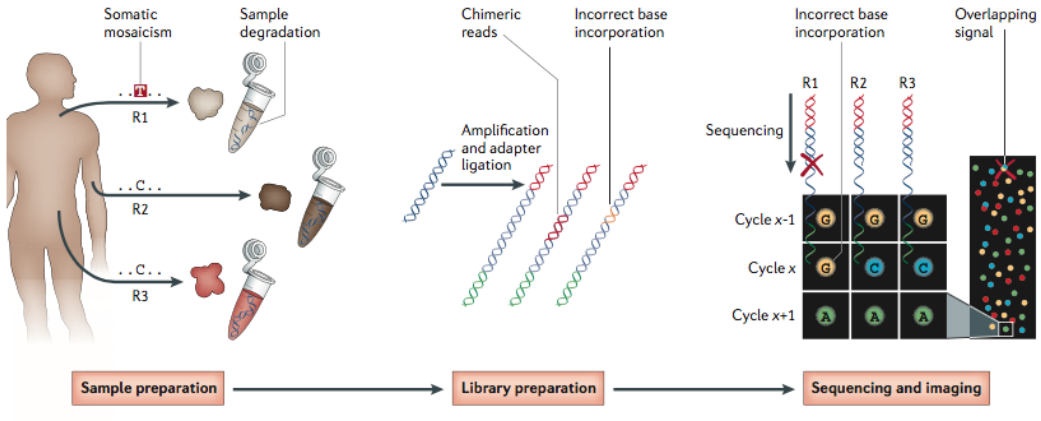


Fig. 2.2 Illustration of the process to obtain DNA sequence data and some of the common experimental source of error. The figure is drawn from [27]

The emergence of NGS has revolutionized the study of genetics and provided valuable resources for other scientific disciplines. As NGS becomes more widely accessible, its use has extended beyond basic research and into broader clinical contexts. It is therefore increasingly important to take into account the errors that arise in the sequencing process. These errors can stem from the bioinformatic analysis and from experimental steps [27].

The high demand for low-cost sequencing has driven the development of high-throughput sequencing technologies that parallelize the sequencing process, producing thousands or million of sequences at the same time. The recent sequencing technologies are able to lower the cost of DNA sequencing beyond what is possible with standard methods [60, 14]. On the other hand, this kind of technology introduces a new challenge : how to recombine all those reads into one single sequence. This is done with alignment algorithms. As expected, the latter also introduce errors that should not be ignored.

In order to be able to use next generation sequencing, two issues must be fixed : DNA conservation and DNA multiplication. NGS requires huge quantity of DNA. Moreover, the place and time where DNA is sequenced is most of the time different from where the DNA is collected. The two issues and their respective solutions together with the errors they can introduce must be taken into account when dealing with the data coming from NGS technologies. Last but not least, the sequencing technology used also matters. The latter also introduced technology specific error.

2.2.1 DNA collection and conservation

The first step consists of collecting DNA and conserving it in the best optimal way possible. Sequencing errors and biases can already arise here. Indeed, sample degradation and contamination during sample isolation and preservation can occur. Moreover, low amounts of high quality, or low quality genomic material can increase amplification errors (during PCR[12]) and decrease sequenced output quality [27].

2.2.2 Polymerase chain reaction

Polymerase chain reaction is a remarkable and very powerful way to copy DNA (see appendix *A*.5 for a more detailed explanation). The latter procedure is able to produce as many copies as desired of a DNA sequence[13]. It is a really simple and powerful in vitro technique that uses a couple of simple properties in order to exponentially amplify a targeted region of DNA.

[12] explained in the next section

[13] https://en.oxforddictionaries.com/, consulted on 1/12/2016

Error produced by PCR

During DNA replication, one error occurs approximately once every 10^5 bases [52]. When an error occurs during replication (e.g, cytosine included instead of adenine), it will create a mutation that will be spread into all daughters sequences. Given mutation's localization, it can have functional consequences. The problem lies in the fact that the mutation detected does not exist in patient. Therefore, the replication process might introduce mutation and therefore constitute a bias in the following analysis. Most of the time, PCR amplification alters mononucleotide and dinucleotide repeat lengths [6]. The latter is usually called polymerase "slippage" and is probably caused by slipped strand mispairing [34]. Other errors can occur because of users, machine failure, primers biases and so on [27].

2.2.3 Sequencing technology

First of all, let's distinguish the different targets: genes, an exome, all genome, RNA seq, etc. - all the different techniques required to split up the DNA. Nowadays, we generally sequence only the exome (part of the genome that contains genes) which represent only 1% to 2% of the whole genome [56, 20]. In the same way, it is important to note the fact that there are several different sequencing platforms and pipelines for NGS [56, 30, 20]. Each of them working differently and using different properties and technologies. Therefore, they produce different data, with specific errors and different level of confidence. Indeed, one study reported that when variants detected in the same sample by the 1000 genomes project and the Complete Genomics platform were compared, 19% of the single-nucleotide variants (SNVs) derived were unique to one dataset [50]. This proves that origin of data[14] has to be taken into account in the post-analysis step. This is likely due to differences in technology for data collection.

For all these reasons, it is important to understand how sequencing methods work. NGS is a terminology used to describe the very latest sequencing technology which has been around since 2007 (see figure A.6). It has replaced the Sanger sequencing and the DNA Microarrays technique. This technique involved attaching DNA to a tiny slide and then measuring other genomes by letting their DNA or RNA stick to that DNA. Nowadays, new methods that look at single molecules already exist [15].

[14] Origin of data stands for all the decision made along the line in order to produce them (e.g, data collection, PCR, sequencing technologies, etc.)

[15] Consulted on 01/12/2016, Introduction to Genomic Technologies, Steven Salzberg, PhD & Jeff Leek, PhD `https://www.coursera.org/learn/introduction-genomics`

Most of the data treated in this master thesis comes from two sources : Ion torrent and Illumina[16]. In the latter, 100 - 150 bp (base pair) reads are used [20]. Each nucleotide going to the sequencer is fluorescently labelled, with the colour corresponding to the base (depending on the convention[17]). One nucleotide at a time is going through the sequencer and an image is taken each time. Thanks to the fluorescent signal, the machine will be allowed to know which base is at any position. Optics-based sequencers (such as Illumina) use visible light. Indeed, they collect photons from arrays to form a visual image. This allows the detection of how bases are incorporated into the genome. As expected, in the Illumina process, substitution errors can arise [27]. Moreover, Illumina has shown a sequence-specific error profile[26]. This profile might be due to either single-strand DNA folding or sequence-specific alterations in enzyme preference [27].

On the other side, Ion Torrent makes use of a different approach: it translates directly chemically encoded information (A, C, G, T) into digital information (0, 1) on a semiconductor chip[18]. Indeed, even though optics-based technology are extraordinarily accurate, they are also extremely expensive. This is the reason why Ion Torrent uses computer software along with integrated circuits and complementary metal-oxide semiconductors (CMOS, as used in laptop or mobile phone). The technology also makes use of an electrochemical detection system called ion-sensitive field-effect transistors (ISFET)[19]. These transitors detect ions when they are released by the DNA polymerase during sequencing by DNA synthesis. All of these electronics aim at detecting and analyzing the release of a hydrogen ion (or proton) which occurs each time a nucleotide triphosphate is added. A little pH change caused by the proton release will be detected by a CMOS sensor. The Ion Torrent sequencer will call the base, going directly from chemical information to digital information. The result is a sequencing technology that is simpler, faster, more cost-effective and scalable.

Both platforms commit errors, but at a different rate. Indeed, it has been shown [59] that the error rate of Illumina libraries (average of 0.9 errors per 100 bases) is lower than the one of Ion Torrent sequencing (with an averages of 1.4 and 1.5 errors per 100 bases for read sequences from the reverse and forward directions, respectively). This results in an illustration in figure A.9

This shows the importance of knowing the technology used to sequence the data before using them. It is highly recommended to work only with data coming from the same sequencing platform.

[16]https://www.illumina.com/, consulted on 02/03/2017

[17]In Highlander, T is colored in red, C in blue, A in green and G in brown [56] while in Illumina it is yellow for G, green fot T, blue for C and red for A [20]

[18]http://bitesizebio.com/

[19]https://www.thermofisher.com

2.2.4 Post sequencing : alignment, variant calling and annotations

The important evolution of sequencing technologies over the last year has changed the problem faced beforehand. Before, sequencing was expensive and time consuming. Indeed, the first attempt to sequence the human genome (The Human Genome Project[20]) required several laboratories all over the world during for 13 years and 3 costing billion dollars. Nowadays, it is possible to sequence an exom within a day for less than 700$. This can be partially explained by the fact that now, the DNA to be sequenced is split into smaller pieces (100 to 200 base pair). Once all those smaller pieces have been sequenced, we face a new problem because the DNA was not sequenced in its natural order, but all the little pieces in a random way. Alignment is the procedure of comparing two or more sequences by looking for a series of individual characters or character patterns that are in the same order in the sequence [44].

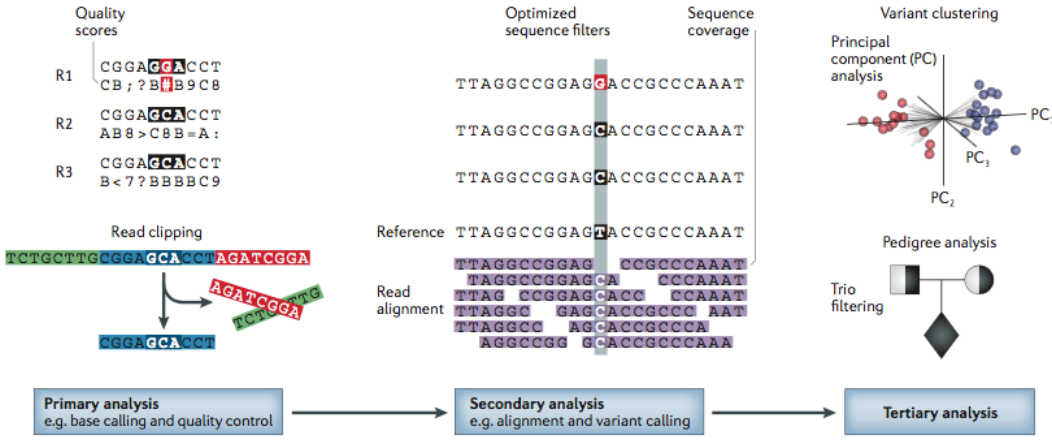


Fig. 2.3 Illustration of the post-processing steps in order to construe the data obtained via sequencing technologies. The figure is drawn from [27]

Because of the new sequencing technology, all the little strands that have been sequenced have to be reorganized. Of course, this procedure is not trivial and introduces some uncertainty. This is why the same frame in the DNA will be sequenced several times and covered by different little strands: it is called "read depth". This allows the alignment process to be easier and more reliable. Another step is the variant called procedure. The procedure can be viewed as the conclusion that there is a nucleotide change given some reference at a specific position in an individual genome [31]. Therefore, the read-alignment methods and variant-calling algorithms also introduce errors. The output of these steps are two (huge)

[20] https://www.genome.gov

files, namely the Binary Alignment Map (BAM)[36] (illustrated in figure *A*.8) and Variant Call Format (VCF)[53] (illustrated in figure *A*.9) formats. They contain all the necessary information in order to analyze and be able to draw conclusions.

The tertiary analysis as shown in figure 2.3 will be described in the next section.

Now the reader has an idea of a general pipeline in order to collect data in a next generation sequencing based study and is aware of all the sources of error present.

2.3 Highlander : a genomic database

This section will present how all the data, collected as described before, are managed, classified and enriched with other information. In order to address this topic, the presentation will be based on a particular genomic database: Highlander (https://sites.uclouvain.be/highlander/). The latter was developed in-house at de Duve Institute (http://www.deduveinstitute.be/) by Raphaël Helaers [55]. The aim of Highlander is to centralize next generation sequencing data and provide end users with an intuitive graphical user interface in order to analyze it. When raw data generated by any NGS equipment (such as discussed above, Illumina's MiSeq, Life Technologies' Ion Torrent, ...) and processed by any pipelines, as long as an alignment and variant calls are produced, in BAM and VCF formats, respectively. Informations contained in those files are extracted and will be automatically annotated using multiple sources (as shown in figure 2.4).

2.3.1 Database

The MySQL (http://mysql.com) engine, one of the most popular open source solutions, was chosen because of its capacity to manage huge amount of records. This engine is suitable for most uses of Highlander.

2.3.2 Data sources

In order to investigate data through Highlander, the variant call (VCF) file must be imported into the database. The alignment file (BAM) is not mandatory but highly recommended because it brings valuable information for the analysis. In order to import the VCF file, Highlander is equipped with dbBuilder. The latter tool allows to draw variants from the VCF, annotates them using different biological databases and imports them into the Highlander database [55]. As shown in figure 2.4, Highlander relies on several biologi-

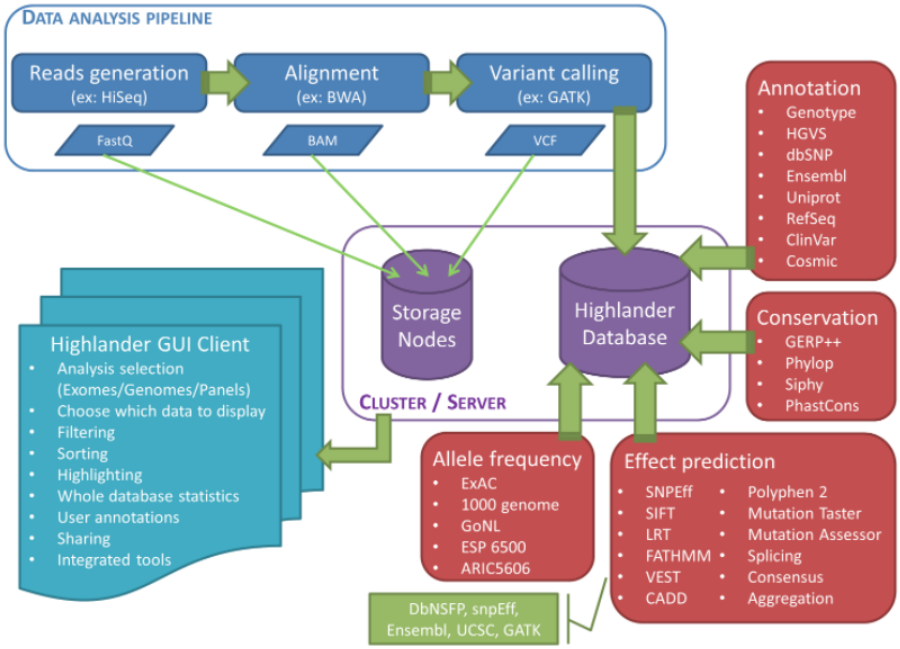


Fig. 2.4 Architecture of Highlander : output files produced by the primary pipeline are stored in a server, variant calling being also imported in the Highlander database, and annotated by a large variety of sources. Highlander graphical user interface clients connect to the Highlander database and provide end-users a comprehensive set of tools and functionalities to analyze variants. Figure drawn from [55]

cal databases such as dbNSFP [21] which gives, amongst others, predictions from different programs (Mutation Taster[22], Polyphen[23], FATHMM[24], ...), but also minor allele frequencies from 1000 Genome project [11]. It has also access to public database such as ExAC (http://exac.broadinstitute.org), GoNL [5] and RefSeq (http://www.ncbi.nlm.nih.gov/refseq/). Therefore, each variant is annotated using several sources and programs in order to predict the pathogenicity of rare missense variants. Nowadays, nearly all those prediction programs use machine learning methods [17, 21] (For example, see in annexe a partial list of tools and resources to annotate DNA sequence variants drawn from [2]).

[21] http://varianttools.sourceforge.net/Annotation/DbNSFP
[22] http://www.mutationtaster.org/
[23] http://genetics.bwh.harvard.edu/pph2/
[24] http://fathmm.biocompute.org.uk/

2.3.3 Variant filtering made easy

Once all previous steps have been executed, the user can classify and retrieve specific variants from the database thanks to filters. The results are shown in a table, with each row representing one variant from one sample, and each column describing one feature / annotation of the variant [55]. On the one hand, the common filters allow to select variants that meet a specific criteria in a given column (e.g. variants with a minor allele frequency lower than a particular percentage in ExAC, variants that have been predicted as damaging by some prediction program, variants at a specific genomic position, etc.). On the other hand, advanced filters (e.g. magic filters) allow more complicated and specific filtering such as selecting damaging variants that occur in the same gene in a set of samples, select variants that affect the same codon, etc [55]. The power of those filters (basic and advanced) described above remains in the fact that they can be combined with *AND* and *OR* logical operators.

2.3.4 Make mutation identification easier

Highlander offers a powerful and easy-to-use software for the expert[25]. The software allows extensive filtering and analysis of variants. It is designed to be able to tackle a huge variety of complex hypotheses and questions the different users might have. Indeed, it has already been used in several studies in order to identify mutations causing a range of vascular anomalies (e.g. see in [37], in [63], etc) [55].

However, this is still a huge amount of information to be handled for a human being, which is why additional features will be added in Highlander in order to make this "data exploration" phase faster. The following chapter will explain and explore the concept of region-based aggregation tests. It is being widely used in genetics and could fasten drastically the process of mutation discovery.

[25] Mainly biologists, doctors and laboratory assistants

Chapter 3

An overview of dedicated statistical test

As discussed in the previous chapter, gene identification is the first step in order to understand how protein and disease behave, which at then end serves as a starting point for developing new treatments [12]. The main challenge here is to find a way to prioritize information so that the investigator knows the path to follow, with some points to start with. Therefore, the main idea is to organize the information according to the question that has to be tackled. In order to save as much time as possible and make mutation discovery quicker, the list of variants should be ordered using one principal criteria: the top of the list (or ranking) should be the ones that are causing - or playing a role in - the disease under study. Given all the errors present in data (see chapter 2), it is clear that obtaining such a clear answer is not realistic. Therefore, one must settle for an organization, given the probability of being responsible - or taking part in the disease.

This chapter will explore already existing methods for rare variants association testing that have been developed. Even though sequencing provides an unprecedented opportunity to investigate the roles of rare variants in complex diseases, detecting these variants in sequencing-based association studies remains a substantial challenges [62]. As explained in the first chapter, the problem lies in the fact that when trying to identify rare variants (MAF < 0.5 %), most of the mainstream approaches are underpowered or ineffective [35, 62]. Because of the growing interest in rare variants identification (see chapter 1), several tests and packages have been developed over the last decade. Therefore, this section will give an overview of association tests for rare variants.

An important remark is that we won't talk about or explore the power of the tests, neither in the following review of literature, nor in the results. This is simply because most papers and references present in the literature already and extensively focus on that aspect in order to ensure the power of the newly proposed test. Thanks to all the work done prior to this paper, we can state that the tests used in this work are powerful for the situation under study.

3.1 Single-Variant Tests

The classical approach in Genome Wide Association Studies (GWASs) in order to test for association between genetic variants and complex traits is a single-variant test. The association between each variant and a trait is typically evaluated by linear regression for continuous traits and by logistic regression for binary traits [62]. The latter allowed to identify thousands of trait-associated loci. If the sample size is important enough, this method can allow low-frequency variant identification. As an example, in a sample of 8000 people, the test was able to detect an association between insulin processing (trait) and variants in *SGSM2* (gene). The latter had a $MAF = 1.4\%$ [62]. Knowing that, by convention, a variant is considered to be rare with a $MAF < 0.5\%$, single-variant tests can still be used in this case if the sample sizes are large enough and/or the effect sizes are large [48]. But the test undergoes an important loss of power when dealing with rare variants. For example, with an odds ratio (OR)[1] of 1.4, the sample sizes required in order to achieve 80% power are $6,400$, $54,000$, and $540,000$ for a MAF $= 0.1$, 0.01, and 0.001, respectively, if one assumes 5% disease prevalence and a significance level of 5×10^{-8} [62]. Therefore, if some assumptions are respected (sample sizes are large enough, effects are very large, MAF not too small) single-variant tests can still be useful. Nevertheless, in the scenario that investigators are dealing with in the GEHU lab, sample size are way too small and the MAF very small (because associated diseases are very rare) and moreover effects size are, in most of the case, unknown. Therefore, other test are needed.

3.2 Region-Based Aggregation Tests

The first idea that comes to mind in order to overcome the above difficulties is to aggregate all the variants in a gene or specified region instead of testing each of them individually. This is what aggregation tests do. By doing so, when multiple variants in the group are associated with a given disease or trait, the latter test achieves increasing power. Here, mainly regression-based methods with the ability to easily adjust for covariates[62] (such as population stratification, sex, ancestry-related principal components and, if needed, it can also include the constant vector for modeling the intercept[48]). The following methods are categorized according to their assumption concerning the underlying genetic model.

First of all, let's introduce the general statistical model underlying most of the rare-variant tests. Assuming that n subjects have been sequenced (data collected as explain in chapter 2) in a region (could be à gene, severals genes, exome, etc) containing p variants. Let's denote

[1] An odds ratio is a measure of association between an exposure and an outcome [42]

by y_i the phenotype of subject i, with mean μ_i. For each patient i, there are (potentially) q $\mathbf{X}_i = (x_{i1}, ..., x_{iq})$ covariates, and also $\mathbf{G}_i = (g_{i1}, ..., g_{ip})$ allele counts for p variants present in the region. By convention, the genotype matrix $\mathbf{G}$ is filled with 0, 1 or 2 according to

- 0 if homozygous reference (also called major homozygous[2]) : observed nucleotide is the same as the reference[3] for both allele.
- 1 if heterozygous : observed nucleotide is the same as the reference for one allele, but differs for the other allele.
- 2 if homozygous alternative (also called minor homozygous[4]): observed nucleotide is different as the reference for both allele.

As an example, if $G_{ij} = 0$, this means that patient i is homozygous reference for the position j, or in other words, that he doesn't have variant j.

Furthermore, let's assume that y_i follows a distribution in the quasi-likelihood family and consider the following generalized linear model [62, 2, 68] :

$$\begin{cases} \mu_i = \alpha_0 + \mathbf{X}_i\alpha + h(\mathbf{G}_i)\beta + \varepsilon_i & \text{(continuous trait)} \\ logit(\mu_i) = \alpha_0 + \mathbf{X}_i\alpha + h(\mathbf{G}_i)\beta & \text{(binary trait)} \end{cases} \tag{3.1}$$

where

- α_0 is an intercept
- $\alpha = (\alpha_1, ..., \alpha_q)^T$ are the regression coefficients for the covariates $\mathbf{X}_i$
- $\beta = (\beta_1, ..., \beta_p)^T$ are the regression coefficients for the allele counts $\mathbf{G}_i$
- $h(\cdot)$ is an arbitrary function that has a form only defined by a positive semidefinite kernel function $K(\cdot, \cdot)$
- ε is an error term

It will be shown later that model 3.1 is just a special case (namely a linear kernel). Indeed, more complex choices can be made.

[2] http://genome.sph.umich.edu/wiki/Rare_variant_tests

[3] Here, reference refers to the reference nucleotide sequence used in order to compare the sequence of the patient. The reference used is *GRCh*38 and can be found at https://www.ncbi.nlm.nih.gov/grc/human/data

[4] http://genome.sph.umich.edu/wiki/Rare_variant_tests

The score statistic of the marginal model for variant j is defined as

$$S_j = \sum_{i=1}^{n} g_{ij}(y_i - \hat{\mu}_i) \tag{3.2}$$

where $\hat{\mu}_i$ is the estimated mean of y_i under the null hypothesis ($H_0 : \beta = 0$) and is obtained by application of the null model $\mu_i / logit(\mu_i) = \alpha_0 + \mathbf{X}_i\alpha$ [62]. Note that, the statstics S_j will be positive if variant j increases the risk of disease or trait values, and it will be negative if variant j decreases the risk of disease or trait values.

3.2.1 Burden Tests

Probably the most famous class of aggregation tests is called burden tests. These collapse information of multiple genetic variants into a single genetic score and test for association between this score and a trait [62, 35, 33]. The classical and basic approach simply count all the number of minor alleles across all variants in the region of interest. Therefore, the summary genetic score is

$$C_i = \sum_{j=1}^{p} w_j g_{ij} \tag{3.3}$$

where w_j is the weight for a variant j. In comparison with the previous model, this is identical to putting $\beta_j = w_j\beta$ in the regression model in equation (3.1) and testing $H_0 : \beta = 0$ in the simplified model $\mu_i / logit(\mu_i) = \alpha_0 + \mathbf{X}_i\alpha + C_i\beta$ [62]. The corresponding score statistic to test $H_0 : \beta = 0$ is then

$$Q_{Burden} = \left(\sum_{j=1}^{p} w_j S_j \right) \tag{3.4}$$

a p value can be obtained by comparison to a chi-square distribution with 1 degree of freedom [62, 48]. The summary genetic score C_i can be modified in order to accommodate various assumptions about disease mechanism. This is the case, for example, of the cohort allelic sums test (CAST) or the combined and multivariate collapsing (CMC). They are subclasses of Burden test. The CAST makes the hypothesis that the presence of any rare variant increases disease risk. Therefore, considering a dominant model, the test sets the genetic score $C_i = 0$ if no minor alleles are present in the region and $C_i = 1$ otherwise.

Let us note that, even though classical burden tests make use of the regression method, some also have been constructed outside this framework. This is the case of the above CMC test. The latter collapses rare variants, as in the CAST, but with three main differences. It

uses different MAF categories, evaluates the joint effect of both common and rare variants (not only rare ones) through Hoteling's test (not using a single regression) [62]. Another alternative to regression is to obtain p values by permutation. This is done in the weighted-sum test (WST) of Madsen and Browning. The latter uses the Wilcoxon rank-sum test.

One needs to be careful regarding the strongest assumption behind all Burden tests : it is assumed that all rare variants under study (in the region of interest) are causal and associated with the trait of interest with the same direction and magnitude of effect (after adjustment for the weights)[69, 62, 33]. If the latter isn't respected, this could end up in a substantial loss of power.

3.2.2 Adaptive Burden Tests

In order to overcome the natural limitation of the burden tests described in the previous section, another approach was proposed and led to the creation of adaptive burden tests. Therefore, these methods are robust and can handle null variants and both trait-increasing and trait-decreasing variants correctly [35]. For example, the aSum (developed by Han and Pan) [15] is made of 2 steps. First, it begins with an estimation of the possible direction of effect for each variant. Then it performs the classical burden test, using the estimated directions as weights. Therefore, it will assign $w_j = -1$ when β_j is likely to be negative and $w_j = 1$ otherwise [62]. The aSum test is a first example of data adaptive burden test. The authors (Han and Pan) recommend to use the aSum-P test. The latter uses a proper permutation in order to compute p values for the aSum test [15].

the step-up approach selects the optimal combination of rare variants and aggregate them into a single group [19]. It refines the procedure of the aSum and also allows to assign $w_j = 0$ when a variant is unlikely to be associated with a trait [62].

Another data adaptive test is the estimated regression coefficient (EREC) which, as the name indicates, assigns weights by estimating a regression coefficient of each variant [38]. The idea behind EREC is that true regression coefficient β_j is an optimal weight to maximize power[62]. But the latter is difficult to achieve for rare variants. Indeed, β_j estimates are unstable when the minor allele count (MAC) is small [62, 38]. In this case, the EREC test tries to stabilize the estimates by adding a small constant to the estimated β_j, which might reduce the optimality of the EREC test and make it behave more like a burden test (not data adaptive anymore) [32]. The p values are estimated thanks to parametric bootstrap [62].

As a third way to improve classical burden tests, the variable threshold (VT) is based on the assumption that the link between allele frequency and effect size might vary greatly with the intensity of selection which motivates a variable threshold approach [54]. VT test

choses optimal frequency thresholds for burden tests of rare variants and assesses statistical significance by permutation or analytically [38, 62].

The kernel-based adaptive cluster (KBAC) method first classify variant, and then test for association by using kernel-based adaptive weighting [62]. Here, covariates (such as sex, population stratification, etc) can be taken into account in the analysis. The authors (Liu and Leal) demonstrated, on the one hand, that KBAC has a greater power than other rare variant analysis methods such as the CMC and WSS, and on the other hand, in the presence of variant misclassification and gene interaction, KBAC is particularly advantageous [40]

This section showed that adaptive burden tests constitute an improved version of the classical burden methods. Indeed, they rely on fewer assumptions concerning the underlying genetic model. The reader can also note that many of the tests described in this section are based on two-step procedures. However, as limitations, one can observe, first, that the estimation of regression coefficients of individual variants is often difficult and unstable for rare variants, and secondly, the permutation needed by most adaptive tests in order to estimate p values makes them computationally intensive [62].

Studies have compared many adaptive tests and have come to the conclusion that they share similar power to that of variance-component and combined tests (see in the next section) [4].

3.2.3 Variance-Component Tests

Variance component tests were introduced in order to allow flexibility regarding two main hypothesis concerning variants : direction of effect and effect size. Indeed, tests are able to handle both protectives and deleterious variants, as well as variants with effect sizes of different magnitude [48]. This new class of methods, instead of grouping rare variants from a region of interest, resorts to random-effects models[35]. These methods test for association by evaluating the distribution of genetic effects for a group of variants [62]. Tests that are part of this class include especially C-alpha [47], the sum of square score (SSU) test [66], the evolutionary mixed model for pooled association testing (EMMPAT) [28], and the sequence kernel association test (SKAT) [22, 32, 33, 69, 68]. They assess the distribution of the aggregated score test statistics[5] of each variants [62]. Let us already point out the fact that, if there are no covariates, SKAT reduces to a C-alpha test. But, contrary to the later, SKAT can also handle SNP-SNP interaction [69]. Going back to model 1 (see equation (3.1)), a random effects model states that regression coefficients β_j follow a distribution with mean 0 and variance $w_j^2\tau$ [48]. In this case, the test hypothesis for association reduces to $H_0 : \tau = 0$

[5] eventually using weights

(in others words, all effects β_j being equal to 0) by using a variance-component score test (as described above). As an example, the SKAT test statistic is given by

$$Q_{SKAT} = \sum_{j=1}^{p} w_j^2 S_j^2 \tag{3.5}$$

Clearly, one can see that equation 3.5 is simply a weighted sum of squares of single-variant score statistics S_j [62]. Contrary to burden tests (see equation (3.4)), SKAT aggregates S_j^2 instead of S_j which make it able to handle both protective variants with and deleterious variants.

As a global remark concerning any large-sample-based methods, when dealing with binary traits, large-sample-based p value computation might produce inaccurate type I errors rates if sample sizes or total MAC are small [62]. In this situation, if the number of cases and the number of control (e.g. affected patient and healthy people) are equal, the false positive rates can be underestimated. On the other hand, if they are not equal, the latter can be overestimated Lee [32]. To address the latter problem, Lee and al [32] developed a moment-based method capable of adjusting the asymptotic null distribution using estimates of the exact small-sample variance and kurtosis of the test statistic [62]. Despite this, if the MAC is very low, obtaining accurate p value estimates might still require bootstrap or a permutation approach.

3.2.4 Omnibus Tests

Also called combination tests, omnibus tests combine burden and variance-component tests. The main idea behind them is to aggregate evidence from multiple complementary sources [48]. On the one hand, variance-component tests are more powerful than burden tests if the region of interest has many non causal variants or if the causal variants are both protective and deleterious (as explain above). On the other hand, burden tests are more powerful than variance-component tests if the region of interest has a high proportion of causal variants with the same direction of association [62]. Because both previously described situations can happen, it is better to use omnibus tests.

For example, Derkach et al. [10] designed a method that combine p values of the two tests (burden and variance-component) with Fisher's method, and a permutation in order to compute the significance of the test [48]. The Fisher statistic takes the form

$$Fisher = -2\log(p_{Burden}) - 2\log(p_{SKAT}) \tag{3.6}$$

where p_{Burden} and p_{SKAT} are p values obtained from burden and SKAT tests respectively [62].

Another approach is to use convex combination of burden test and the SKAT statistics with the weight coefficient predetermined or data adaptive. Lee et al. [22, 32, 33, 69, 68] suggested the SKAT-O : a linear combination of the following form

$$Q_\rho = \rho Q_{Burden} + (1-\rho) Q_{SKAT} \qquad 0 \leq \rho \leq 1 \tag{3.7}$$

where parameter ρ can be viewed as a pairwise correlation among the genetic effect coefficients β_j in equation (3.1) [62]. As one can expect, the parameter ρ is likely to be unknown. Therefore, SKAT-O computes an approximate optimal weight coefficient, $0 \leq \rho \leq 1$ heuristically. It will evaluate p values over a range of ρ values and selects the value which leads to more evidence for association[6] [48]. One of the strength of the SKAT-O relies in the fact that it's asymptotic p value can be computed efficiently with one dimensional numerical integration tools [62, 48].

For the reader, another example similar to the SKAT-O, namely the EMMPAT model, is well described in [48].

Omnibus tests reach robust power and outperforms burden tests or variance-components tests in a wide range of scenarios. But if the assumptions[7] behind them (burden or variance-component) is largely true, then they can be less powerful than any of these tests [35]. Because investigators rarely have much prior information about the underlying genetic architecture in real life applications, omnibus tests constitute an attractive choice. As a global remark on combinations tests, let's point out the fact that a naive procedures consisting of picking the minimum p value of different methods will most of the time yield to an inflated type I error rate [62].

3.2.5 Others methods

There are still other methods that do not fit the above categories but are also designed for rare variant association. The latter include Bayesian approaches. Because of computational issues, they have not been as popular for rare variants as the frequentist approach [48]. This section will focus on one main example, namely the exponential-combination (EC) test, but the reader can refer to [48] for more examples.

As seen previously , burden and variance-component tests use linear and quadratic sums of S_j respectively, whereas the EC test uses an exponential sum of S_j^2 [39]. It is based on the

[6] In others words, the minimum p value computed over the grid

[7] Reminder burden assumes all variants in region of interest have same direction and magnitude, whereas variance-component deals with variants with different direction and magnitude

assumption that only one or very few variants in a gene or region is a causal variant [39]. The EC test statistics is as follow

$$Q_{EC} = \sum_{j=1}^{p} \exp\left(\frac{S_j^2}{2var(S_j)}\right) \tag{3.8}$$

As usual, if the assumption fits well the unknown underlying genetic model then the test will outperform the others. In the EC test case, if only a very small proportion of variants are causing the disease, thanks to the exponential function increasing very quickly as S_j^2 increases, it will have higher power than the above tests [62]. Obviously, if too many variants are associated with the trait, then the EC test will performs poorly. Moreover, in order to estimate p values, the permutations method is required because the null distribution of Q_{EC} is not known.

3.2.6 Summary of methods

The table 3.1 shows a good summary of the available dedicated statistical tests already existing[8]. Moreover, the reader can find a list of software packages implementing some of the tests described above in appendix C

In real life application (e.g. biomedical research, gene identification process in order to better understand a particular disease), the true disease model is not known and might vary. Nevertheless, as seen in this section, the power of each test depends on the true disease model [62]. As a result, the best option available is to rely on omnibus tests because they can adapt themselves and better fit the possible true hidden model that investigators are trying to discover.

[8] Another table can be found in [48] and also in http://genome.sph.umich.edu/wiki/Rare_variant_tests

	Description	Methods	Advantage	Disadvantage	Software Packages
Burden tests	collapse rare variants into genetic scores	ARIEL test, CAST, CMC method, MZ test, WSS	are powerful when a large proportion of variants are causal and effects are in the same direction	lose power in the presence of both trait-increasing and trait-decreasing variants or a small fraction of causal variants	EPACTS, GRANVIL, PLINK/SEQ, Rvtests, SCORE-Seq, SKAT, VAT
Adaptive burden tests	use data-adaptive weights or thresholds	aSum, Step-up, EREC test, VT, KBAC method, RBT	are more robust than burden tests using fixed weights or thresholds; some tests can improve result interpretation	are often computationally intensive; VT requires the same assumptions as burden tests	EPACTS, KBAC, PLINK/SEQ, Rvtests, SCORE-Seq, VAT
Variance-component tests	test variance of genetic effects	SKAT, SSU test, C-alpha test	are powerful in the presence of both trait-increasing and trait-decreasing variants or a small fraction of causal variants	are less powerful than burden tests when most variants are causal and effects are in the same direction	EPACTS, PLINK/SEQ, SCORE-Seq, SKAT, VAT
Combined tests	combine burden and variance-component tests	SKAT-O, Fisher method, MiST	are more robust with respect to the percentage of causal variants and the presence of both trait-increasing and trait-decreasing variants	can be slightly less powerful than burden or variance-component tests if their assumptions are largely held; some methods (e.g., the Fisher method) are computationally intensive	EPACTS, PLINK/SEQ, MiST, SKAT
EC test	exponentially combines score statistics	EC	is powerful when a very small proportion of variants are causal	is computationally intensive; is less powerful when a moderate or large proportion of variants are causal	no software is available yet

Abbreviations are as follows: ARIEL, accumulation of rare variants integrated and extended locus-specific; aSum, data-adaptive sum test; CAST, cohort allelic sums test; CMC, combined multivariate and collapsing; EC, exponential combination; EPACTS, efficient and parallelizable association container toolbox; EREC, estimated regression coefficient; GRANVIL, gene- or region-based analysis of variants of intermediate and low frequency; KBAC, kernel-based adaptive cluster; MiST, mixed-effects score test for continuous outcomes; MZ, Morris and Zeggini; RBT, replication-based test; Rvtests, rare-variant tests; SKAT, sequence kernel association test; SSU, sum of squared score; VAT, variant association tools; VT, variable threshold; and WSS, weighted-sum statistic.

Fig. 3.1 Summary of Statistical Methods for Rare-Variant Association Testing, table drawn from [62].

Chapter 4

Methods

This chapter will first introduce and motivate a special choice of tests that has been made in order to achieve this master thesis' goal: make gene identification easier. In order to enrich the existing literature, we expose here a complete and precise overview of the package we chose to use. Nowadays, it is still unclear what is available in it and there are no articles that list the possibilities. Therefore first section will give a depth description of the supported class of tests. Then it will go through the different choices of kernel and the weighting procedure available. After that it will quickly review the p value computation methods. A special section will be dedicated to the unique and special case that is the SKATO. Then, a full explanation of the reasons and motivations that have pushed us to select the SKAT package. This will be followed by the presentation of our new statistical framework that we implemented and used for the kind of question arising in the laboratory of Human Molecular Genetics. We will describe it steps by steps starting with the data retrieval and case-control design. Then we will present our proposed function (*creatGenoMatrix*) that enable to automatically transforms the data from Highlander into data that can be handle by the SKAT package. Finally, it will go through all the process within the analysis step.

4.1 the SKAT package

The previous chapter introduced briefly a wide range of dedicated statistical methods, all tackling rare variants association problems. Here, the packages software chosen, called SKAT[1], will be described more in details along with more exhaustive theoretical explanations of the underlying models. The following will be mainly based on the corresponding articles [68, 69, 33, 32, 22].

[1] http://www.hsph.harvard.edu/skat, http://cran.r-project.org/ web/packages/SKAT

The freely available SKAT package was developed by several investigators using the programing language *R*. The latter, along with *python*, seems to be the more indicated choice of programing languages because of their monopoly in the biostatistics world, but also for further development[2]. Namely, one of the main ideas will be to use machine learning methods which well supported by *R* or python.

The reader can also find free valuable tutorials[3] and full library description[4] online.

The main function inside the package that allow to run the test are namely the *SKAT_Null_Model* and the *SKAT*. The first one computes model parameters and residuals for the the second one. The latter performs the actual test for association. Note that some derived function from one of the 2 above have been proposed in order to handle specific situation. The table bellow shows some of the useful function of the package.

- *SKAT_Null_Model_MomentAdjust* : for small sample adjustment (see section 4.1.5 and chapter 5)
- *SKATBinary* : for binary traits (see section 4.1.3 and chapter 5)
- *SKATBinary_Single* : single variant tests for binary trait (see chapter 3)
- *SKATBinary_CommonRare* : to test for the combined effect of common and rare variants (see section 4.1.5 and chapter 6)

4.1.1 Supported classes of tests

As introduced in chapter 3, there are different classes of tests, each having their strengths and weaknesses (depending on their assumptions). Inside each class, there exists a wide range of methods, each being optimized for some situations or genetic models. One of the first and main advantages of this package is that it supports not just one, but several methods belonging to the 3 main class (burden, variance-component and omnibus tests). The user can choose between the different classes of tests simply by selecting the desired ρ parameters via the *r.corr* input parameter in the main function. We simply note here that

- $r.corr = \rho = 0$: is the default parameter value, which corresponds to variance-component test (namely SKAT)
- $r.corr = \rho = 1$: corresponds to a Burden test

[2]Indeed, this master thesis will be followed by a PhD that will improve and further deepen the work realized here

[3]http://docplayer.net/42914914-Skat-package-seunggeun-shawn-lee-june-6-2016.html

[4]https://cran.r-project.org/web/packages/SKAT/SKAT.pdf

- $r.corr = \rho \in]0,1[$: corresponds to omnibus test

This implies a huge flexibility and allows to attack a problem with different points of view using only one package. The latter simplifies drastically the integration of the tool inside Highlander. Moreover, there is no need to run different softwares and there is no problem dealing with several programming languages.

The following section will go through all the implemented methods and review all the cases the package was designed to handle.

4.1.2 Different choices of kernel

Here, the reader will be introduced to the different kernel choices available in the SKAT package. In the next chapter, we will investigate these different choices and see how they differ. It is important to remember that different kernels allow to model different genetic models. Therefore, the more flexible and adaptable the kernel is, the more complex and different problem (different disease, with different underlying genetic models) can be tackled. The element of interest here is the function $h(\cdot)$ in equation (3.1). Indeed, it is the most important element of the equation because it is the one that dictate how variant should be taken into account (their genotype) for the disease risk. Suppose $h_i = h(G_i) = \sum_{l=1}^{n} K(G_i, G_l)$, and in order to simplify and illustrate the following, the score statistics given by 3.5 will be rewritten, using matrix notation, thanks to 3.2 as

$$Q_{SKAT} = (y - \hat{\mu}_i)^T K (y - \hat{\mu}_i) \tag{4.1}$$

More precisely, K, called the kernel function, is an $n \times n$ matrix. The latter function $K(G_{i_1 j}, G_{i_2 j})$ measures the genetic similarities between individual i_1 and i_2 within the region of interest thanks to the p variants taken into account [69, 68]. The key concept is that, by choosing different kernel functions, it is possible to define new bases and associate models [32]. As seen above, it is clear that the choice of kernel impacts the underlying basis for the nonparametric function dictating the relationship between case-control status as well as the SNPs in the SNP set [68]. Moreover, any positive semidefinite function $K(G_{i_1 j}, G_{i_2 j})$ can be viewed as a potential kernel function [69].

We develop here a brief overview of the different kernel, which corresponds to the parameter *method* in the main function of the SKAT package.

The linear kernel

As a first example, and the more simple one, the linear kernel is defined by

$$K(G_{i_1 j}, G_{i_2 j}) = \sum_{j=1}^{p} G_{i_1 j} G_{i_2 j} \tag{4.2}$$

As one can notice, it is simply the usual inner product between vectors of individual i_1 and i_2. Therefore, the model is given by 3.1 (as mentionned earlier). But let's investigates more complex choices of kernel.

The linear weighted kernel

This kernel has been developed under the assumption that β_js (see model (3.1)) are independent [33]. Beginning with matrix notation, the linear weighted kernel corresponds to the choice of, in equation 4.1, $K = GWWG$ (with $W = diag(w_1, ..., w_p)$ a diagonal matrix containing all the weights of the p variants in the region of interest). More precisely, the element (i_1, i_2) is given by

$$K(G_{i_1 j}, G_{i_2 j}) = \sum_{j=1}^{p} w_j^2 G_{i_1 j} G_{i_2 j} \tag{4.3}$$

Equation 4.3 describes a first possible choice of kernel available in the SKAT package, namely, the weighted linear kernel. Of course, the choice of weights has an impact on the power [69].

The weighted quadratic kernel

The weighted quadratic kernel takes the follwing form

$$K(G_{i_1 j}, G_{i_2 j}) = (1 + \sum_{j=1}^{p} w_j G_{i_1 j} G_{i_2 j})^2 \tag{4.4}$$

It will be shown later that this particular kernel enables to handle specific - but occurring in real life application - situations.

The linear identity by state (IBS) kernel

This form of kernel defines similarities between individuals by the number of alleles sharing the same state (e.g 2 individuals sharing the same allele or variant at a particular locus) [69]. The linear IBS kernel proposed by Wessel and Schork [67] reads

$$K(G_{i_1 j}, G_{i_2 j}) = \frac{\sum_{j=1}^{p} s_{i_1, i_2}^{j}(G_{i_1 j}, G_{i_2 j})}{2L} \tag{4.5}$$

where L is the number of variants in the region of interest and $s^j_{i_1,i_2}$ is a function that maps the genotype information for individuals i_1 and i_2 at variant j to a numeric value defined as

- $s^j_{i_1,i_2} = 0$ if individuals i_1 and i_2 are homozygous for different SNP alleles (e.g. $G_{i_1 j} = CC$ and $G_{i_2 j} = GG$)
- $s^j_{i_1,i_2} = 1$ if individuals i_1 and i_2 share one allele (e.g. $G_{i_1 j} = CC$ and $G_{i_2 j} = CG$)
- $s^j_{i_1,i_2} = 2$ if individuals i_1 and i_2 share both allele (e.g. $G_{i_1 j} = CC$ and $G_{i_2 j} = CC$)

Here Wu et al. recommends to use this kernel if there are enough interactions between some causing variants within the region of interest [69].

The weighted identity by state (IBS) kernel

This kernel, as its name indicates, is simply a weighted version of the previous one. The assumptions behind is that similarity between rare alleles is more informative than similarities in common alleles [68]. Therefore it is given by

$$K(G_{i_1 j}, G_{i_2 j}) = \frac{\sum_{j=1}^{p} w_j s^j_{i_1,i_2}(G_{i_1 j}, G_{i_2 j})}{\sum_{j=1}^{p} w_j} \tag{4.6}$$

Here, more weight can be assigned to variants and can be determined a priori based on supplementary information (e.g variants annotations, possibly known genetic model, etc.) [67]. Here also, it is recommended to use this kernel if there are significant interactions between many causing variants within the region of interest [69].

The 2wayIX kernel

This kernel is a product kernel consisting of main effects and SNP-SNP interaction terms [69]. Indeed the latter can be written as[5]

$$K(G_{i_1 j}, G_{i_2 j}) = 1 + \sum_{j=1}^{p} G_{i_1 j} G_{i_2 j} + D(G_{i_1 j}, G_{i_2 j}) \tag{4.7}$$

Now that the reader has been through all the 6 available kernels of the SKAT package, one needs to fully understand how and why some kernels (here, linear weighted, weighted quadratic and weighted IBS kernel) allow weighting.

[5]Equation (4.7) was directly derived from figure A.12 see figure $A.12$ in appendix $A.12$. More precisely, Matrix $D(G_{i_1 j}, G_{i_2 j}$ in equation (4.7) is computed via the *K1_Help* function described in the same figure.

Weighting procedure

A weight, for example w_j, is modeling the relative contribution of variant j to the score statistic. Therefore, one can see that the objective is to reach the point where each variant weight w_j efficiently reflects the true impact the variant j has in the disease. But again, without the true genetic model, the best achievable solution is only an approximation. This section presents some of the most used weighting strategies used in the field. Note that the SKAT package allows a complete flexibility for the weighting procedure.

In order to focus on rare diseases (e.g a disease that touches less than 0.1% of the population), one simply has to put the weight $w_j = 1$ if the MAF of variant j is bellow the desired threshold ($MAF_j < 0.1\%$) and $w_j = 0$ otherwise. Another example of disease mechanism adaptation is illustrated by the work of Madsen and Browning. They proposed a continuous weight function in order to give more importance to rare variants. They end up with the following weighting function:

$$w_j = \frac{1}{\sqrt{(MAF_j(1-MAF_j))}} \tag{4.8}$$

Wu et al [69] also proposed the so called family of beta densities:

$$w_j = beta(MAF_j, a_1, a_2) \tag{4.9}$$

with a_1 and a_2 chosen in order to reflect the relationship between MAF and the magnitude of effects [48]. Note also that MAF_j is evaluated across cases and controls combined [69]. One can see that the equation 4.9 leaves place for lots of different combinations of choice for parameters a_1 and a_2. For example, if the assumption about the underlying genetic model is that rarer variants have larger effects, then one should put $0 < a_1 \leq 1$ and $a_2 \geq 1$ because it will increase weights of variants of interest (rarer variants) while decreasing weights of common variants. Here, Wu et al [69] suggest to put $a_1 = 1$ and $a_2 = 25$ because it allows to give higher importance to rare variants while still according some importance to variants with MAF between 1% and 5%. If one wants to strongly increase weight of rarer variants, then simply putting a smaller a_1 will have the desired impact.

As a final note, there are two weighting procedures to consider. First of all, setting $a_1 = a_2 = 1$ simply transforms a weighted kernel into a linear form. Therefore, equation (4.3) will be equal to (4.2). But the equation (4.9) also includes Madsen and Browning weight 4.8 (see equation 4.8) as a special case (choice of $a_1 = a_2 = 0.5$).

Obviously, any annotations and supplementary information on variants can be used for weight construction, and any personalized weighting procedure can be used. Therefore, we

also recommend to use all valuable information available such as annotation tools, effect prediction and more (see figure (2.4)).

4.1.3 P values computation

Here we simply review all the possible methods available in the SKAT package in order to compute p values. In the next chapter, we will investigate them in order to classify them, in particular thanks to computational resources needed by each of them. The p value can be viewed as the summary of the association test analysis. Indeed, after regressing the trait of interest onto each SNP in the targeted region, tests (most of the time) generate p values in order to rank SNP (or gene, etc.) based on these p values using a threshold to determine whether a SNP (or gene, etc.) should be pushed forward for validation[6] [68]. Here, we need to consider two cases: using the classical *SKAT* function or the *SKATBinary*. Indeed, several methods to compute p values are implemented, but slightly differ from one another. We list them here:

P value computation methods available in *SKAT* function

This function implements 3 different methods. The first one, *Davies* is an exact method that computes the p value by inverting the characteristic function of the mixture chi square [33]. The second one *Liu* is an approximation method that matches the first three moments [69]. Finally *Modified Liu* (*liu.mod*) is a modified version of the *Liu* method that matches kurtosis instead of skewness to improve tail probability approximation.

P value computation methods available in *SKATBinary* function

The authors of the SKAT package have developed an efficient resampling method to compute p values in the special cases of binary traits. This method corresponds to another input variable called *method.bin*. Therefore, when using the *SKATBinary* function, one can choose one of the six resampling methods: efficient resampling (method.bin ="*ER*"), quantile adjusted moment matching (method.bin ="*QA*"), moment matching adjustment (method.bin ="*MA*"), No adjustment (method.bin ="*UA*"), adaptive efficient resampling (method.bin ="*ER.A*") and finally hybrid (method.bin ="*Hybrid*") which is the default value.

Hybrid method selects a method based on 3 informations: the total MAC, the number of individuals with minor alleles and the degree of case-control imbalance[7].

[6] Validation step consists of biological models and experimentations in order to confirm the role of the identified variant (or gene, etc.)

[7] Information directly drawn from https://cran.r-project.org/web/packages/SKAT/SKAT.pdf

It is important to note that the parameter *method* in the *SKATBinary* is not well described[8]. Indeed, it is said that the possible parameter value *Burden* simply corresponds to parameter choice *davies* and *r.corr = 1*. But actually, the *davies* method will never be used because the default method for this function is the *Hybrid* method. A more detailed explanation can be found in appendix *A*.13.

4.1.4 An optimal adjustment : the SKATO test

This section is dedicated to as special choice of parameter value, namely the *optimal.adj*. Indeed, the latter is more complex than the others. Even though it is very unclear and hidden from the reader (in the tutorial or articles). *Prima facie*, the *optimal.adj* simply is another possible p value computation method. But actually it is a complete test, defining its own kernel, using its own p value computation method and more over, choosing itself whether it has to behave like a burden test, like a SKAT test or as an omnibus test.

As said previously, the weighted linear kernel is based on the assumption that β_js are independent. Therefore, when a majority of variants in the region of interest are associated with the phenotype with the same effect size and direction, Burden tests can outperform SKAT. This is due to the fact that the kernels described above do not account for correlation in β. This is why Lee S. et al. [33] offered a new family of kernels capable of incorporating correlation among the variant effects. They end up with a family of kernel of the following form:

$$K_\rho = GWR_\rho WG' \tag{4.10}$$

where R_ρ is the correlation matrix of β such that:

$$R_\rho = (1-\rho)I + \rho 11' \tag{4.11}$$

where ρ is the same as the one in equation (3.7), 1 is a vector with each entry equal to one. Therefore, with $\rho = 0$, K_ρ is equal to the weighted linear kernel and the test will behave as a SKAT test, on the opposite, when $\rho = 1$, K_ρ is equal to the weighted linear kernel but the tests will behave as a squared burden test (A complete mathematical description of the test can be found in [33]).

When it comes to the ρ parameter determination, the authors proposed a method to select ρ to maximize power. Indeed, this is the only omnibus test that do not require the user to fix ρ, which is better because in most of the situations, investigators do not have much

[8] https://cran.r-project.org/web/packages/SKAT/SKAT.pdf

information about it. Instead of using the usual score statistics, the authors used the minimum of the p values as a test statistic. They obtained:

$$T = \inf_{0 \leq \rho \leq 1} p_\rho \tag{4.12}$$

where p_ρ corresponds to the p value computed with Q_ρ. In order to obtain T, the actual implementation of this test within the SKAT package simply performs a grid search across some fixed value of ρ: set a grid $\rho = (0, 0.1^2, 0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1)$ (information found at https://cran.r-project.org/web/packages/SKAT/SKAT.pdf), then the test statistic $T = \min\{p_{\rho 1}, ..., p_{\rho_b}\}$. The resulting optimal test, called SKATO corresponds to a best linear combination of *SKAT* and burden tests that maximize power [33].

4.1.5 Advantages of SKAT package

In this section, we will review some of the most interesting advantages of this SKAT package. On the one hand, thanks to the flexibility offered by this particular software (nearly all kinds of tests supported, large choice of kernel, methods to compute p values), a wide range of problem can already be tackled. On the other hand, the authors of this package paid a particular attention to interesting situations such as epistatic effect (described bellow), small sample adaptation (which is truly needed in real life application) and combined effect of rare and common variants.

Accommodating epistatic effect and prior information under the SKAT

One of the particularity of logistic kernel machine based test remain in the fact that it is built on a statistical framework allowing for flexible modeling of epistatic and non linear SNP effects [68]. Epistasis describes the interaction of severals genes at different loci, where the effect of one of these particular genes depends on the presence of one or more modifier genes (e.g. fur color of Labrador is given by the interaction of two genes, the first determines the fur color whereas the other gene controls the expression of the first gene[9]). Therefore, having at disposal a package capable of handling epistatic effects might be recommended because many diseases have such underlying genetic models. Indeed, individual SNPs may not have individually significant effects, but their interaction might have larger impact [68].

In order to accommodate model (3.1) for epistatic effects, Wu et al. [69] simply replace expression $G_i\beta_i$ by a positive semidefinite kernel function $K(\cdot,\cdot)$ - here, the weighted linear kernel, the weighted quadratic kernel and the weighted IBS kernel. As a general comment,

[9]Example drawn from http://www.biology-online.org/dictionary/Epistasis

variance-component tests, such as the SKAT, aggregate associations between variants and phenotypes via a kernel matrix, which allows for SNP-SNP interactions such as epistatic effects [32].

Small-sample adaptation

Another important characteristic of the package is that Lee et al. [32] have developed an analytic moment matching adjustment method that adjusts the asymptotic null distribution for small sample. This is a key feature because most of the analysis we are doing inside the GEHU lab are based on very small samples (e.g. 50 cases, 50 controls). The SKAT package applies (via the input parameter *Adjustment* in the *SKAT_Null_Model* function), regardless of the choices made (type of tests, kernel, p value methods, etc.), this small-sample adaptation when the total number of individuals (cases and controls together) is less than 2000. This procedure is based on a precise estimation of the variance and kurtosis of the sample at hand. This procedure allows to gain power and make therefore the test perform better than their unadjusted counterparts in small samples studies [32]. As proposed by Zhang and Lin [71], in order to reduce the biais due to small sample, the restricted maximum likelihood (REML) estimator of the variance component should be used. The reader can persuade himself of the improved results thanks to the small-samples adaptation via the results obtained by Lee at al. [32] (all the mathematical model and reasoning can be found in the same reference) or simply by taking a look at the dedicated appendix $A.14$ which only shows the summarized graphical results.

Family based association tests

Family based association tests that have been proposed [23] have a clear correspondence for the proposed tests for population based test presented above. It wasn't the purpose in our case to focus on family studies, but as it can be seen in the test proposed by Ionita-Laza at al. [23], it is clear that the SKAT package already handles family based studies, and it will be easy to incorporate the new design into the package.

Combined effect of rare and common variants

Last but not least, Ionita-Laza et al. [22] introduced several tests enable to evaluate the cumulative effects of rare and common variants and are available in the SKAT package. This is important because it opens the door to a whole new kind of analysis and others diseases. Indeed, via the *SKAT_CommonRare* function, the user has access to 7 new tests

specially designed to handle situation in which the underlying genetic model is explained by a combined effect of rare and common variants.

We will not go any further into this discussion because this master thesis focuses on rare variants. But the reader should keep in mind that this enlarges even more the type of disease and hypothesis that the SKAT package is able to handle. Indeed, in the future, this will certainly be explored and implemented in Highlander to allow the investigators to reach even more working hypothesis.

Binary and continuous phenotypes testing

Here we quickly insist on the fact that model (3.1) along with the score statistic (3.2) can be used to model both continuous binary traits. We focus and suggest here an implementation of the discrete case because of its higher relevance inside GEHU laboratory. Nevertheless, all the theoretical discussion is completely the same for both types of phenotype[10]. When it comes to our implementation, switching from binary to continuous, it is done effortless by simply switching a parameter in one function (as it will be shown in the next section) and computing the desired phenotype vector (also explained in the next section).

Final remark on the SKAT package choice

We reviewed above a list of reasons that have motivated our choice of package. Indeed, it offers huge flexibility and can handle many different investigators hypothesis. Let's recall that GEHU investigators usually work on rare diseases, which underlying genetic model, is unknown or at least barely known. This is where all the potential of this package comes into play by allowing nearly all possible combination of classes of tests, each designed and optimized for some particular cases. Thanks to that, investigators, by simply pushing a single button, have access to a full analysis with different points of view exploring several hypothesis.

As a general comment, we want the reader to keep in mind that an important element of the choice is based on further developments. Indeed, dedicated statistical tests for identification of genetic association of variants have been created massively since approximately 2010 [48]. The field is brand new and evolves quickly. Therefore, one needs to have a freely available and editable software package (such as SKAT package)at its disposal.

Now that the reader has a global overview of the SKAT package, the next section will introduce our framework that make the link between Highlander and this package while making the exploration much faster and automated.

[10]NB: phenotype and trait are used as synonyms here

4.2 Rare variants association studies design for GEHU laboratory

This section will review our framework and how we did use the SKAT package along with Highlander. Moreover, we review here the main choices made in order to obtain the optimal solution to hasten GEHU investigation process. This choices presented here are the results of one year of investigation and discussion with the GEHU team members. The resulting implementation has therefore been developed in the hope to be the optimal first step for region identification (e.g gene, pairs of genes, etc.) for rare diseases with potentially unknown underlying genetic model and caused by rare variants. Therefore, choices are specific and might be not directly applicable to an other laboratory (e.g. investigators working on diseases mainly due to common variants). Nevertheless, our framework was developed with the idea to be easily adaptable to another laboratory.

Before introducing any results (see next chapter), we simply present here the process that we implemented in order to carry any analysis. One important characteristic that we tried to maintain throughout the process is adaptability. The code should be easy and ready to use for several data composition, investigators hypothesis, filters, etc. As it will be shown latter, thanks to this simple and efficient framework, we will achieve to hasten the investigation process.

4.2.1 Data retrieval and case control design

First of all, we collect data directly from Highlander in the form of huge data frame (as illustrated in figure (A.15)) where each row is a variant and each column has some valuable information about it. Note that this can also be included into the analysis (e.g. via covariates matrix, etc.). During this steps, the investigator is free to choose which variant will be potentially considered in the analysis. In order to reach maximum flexibility, we only require very few mandatory information and format (in order for our code to run properly). The database, in our case *highlanderDB*, should be a MySQL database such that each row corresponds to a different variant and each column contains some annotation or supplementary information about the given variant. Moreover, the database should include, at least, the information that allow to identify an unique way a variant (as discussed in section 2.1.6: the chromosome and position number, the reference and alternative form) but we highly recommend to include:

- *Pathology*: a column allowing for selecting patient having the same disease. This can be omitted if, and only if, the investigator gives two different data sets (one including only patients affected with same disease, the other one only controls)
- *Platform*: Platform used in order to sequence and detect the given variant

Theses information can be larger than that and could include even more specific criteria. The most important point to be remember is that these highly recommended criteria should ensure both that the data actually makes sense and that they are as homogenized as possible. Indeed it is important that all the information comes from the same kind of source (e.g. all data have been generated by the same machine with a given particular specific errors). Obviously, for the data to make sense, they should contain only patients with the same disease and the controls should include only healthy people. This last comment leads us to a small comment on the data sets nowadays.

Limitation of genetic test nowadays and the ideal case-control scenario

One of the main limitation that should never be forgotten when dealing with genetic tests, more specifically for association testing in case-control study, is that we do not have access to true controls. Indeed, even though DNA sequencing is cheaper than ever (as seen in chapter 2), we do not have access to any database gathering only healthy people. On the other hand, for some diseases, it is difficult to qualify an individual as healthy. Indeed, he (or she) can be a carrier but has not contracted the disease yet or been diagnosed. If we add to this the fact that genetic data are certainly the most sensitive data of all, sharing that database will be highly difficult. Each single institute should have its own private "healthy" database, which seems, nowadays, unachievable. The ideal case-control scenario cannot be reached and data set creation should therefore rely on alternatives, such as picking patients having another disease (as controls) that is genetically far and different as possible from the one under investigation. In order to do so, we choose to rely on expertise and knowledge of the GEHU investigators. They know that data have to be as homogeneous as possible and select their controls trying to respect that. Concerning the case and control data (which can be viewed as two different data sets merged in one for the analysis), we highly recommend to ensure that both are retrieved and filtered, using the exact same criteria in order to keep the data set as homogeneous as possible.

As a final comment on the data retrieval, let's see another option that we left wide open to the investigator in order to fit as mush as possible the hypothesis about the disease.

Keeping the power of variant filtering

Because this implementation was designed in order to be fully integrated into Highlander, we wish to keep all the power of filtering offered by it. Therefore all the filters available in Highlander are also present in our data retrieval procedure, and investigators can run test exactly on the desired variants.

One should understand that playing with the filters can drastically change the tests outcomes. Indeed this implies different hypothesis about the possible underlying genetic model. Obviously, changing filters can change significantly the variants included in the data set to be analyzed. As a final comment, we recommend to run the same analysis, several times using different filter values and also different control groups (as it will be illustrated and explained in the next chapter).

4.2.2 Specific Genotype matrix construction with our *creatGenoMatrix* function

Once the data have been retrieved from Highlander, the most crucial step in order to perform any analysis, consists in building a genotype matrix (as seen in model for region based aggregation tests in chapter 3). This relies entirely on data drawn from Highlander's database and can be viewed as the main link between Highlander and the SKAT package. Here again, we put notable effort in order to obtain a code as flexible as possible to be able to fit as much hypothesis and scenario as the GEHU investigators could have. We implemented a function called *creatGenoMatrix* able to construct the genotype matrix of a specific region and variants of interest. There are 5 inputs parameters (as illustrated in figure (A.16)): *data_patientOfInterest_case*, *data_patientOfInterest_control*, *position_name*,*position_value* and an optional parameter *variant*. For a complete description of these, the reader can refer to appendix $A.16-18$ (header of the function, illustration of a possible input parameter, and a complete description of each input parameter).

The output of the *creatGenoMatrix* function is a genotype matrix. Let's recall that (see section 3.2) element $G_{i,j}$ gives to the zygosity ($G_{i,j} \in \{0,1,2\}$) of patient i for variant j in the region of interest. As the reader will see in chapter 5 and 6, this function is powerful in the sense that it allow to drastically hasten the data preparation and therefore the link in between Highlander and the SKAT package. Moreover, it has been developed in order to be able to construct any genotype matrix given the input criteria.

4.2.3 Analysis step

Let's first note that the kind of analysis we are conducting here can be viewed as a tertiary analysis in an NGS post processing pipeline (as illustrated in figure (2.3)). This section will go through the systematic steps that we implemented in our analysis procedure. Note that the main function used within the package are[11]

- *SKAT_Null_Model*: get parameters and residuals from the null model
- *SKAT_Null_Model_MomentAdjust*: get parameters and residuals from the null model for small sample adjustment
- *SKAT*: test for association between a set of variants in a targeted region for continuous or dichotomous phenotypes using kernel regression framework
- *SKATBinary*: test for association between a set of variants in a targeted region for binary phenotypes with asymptotic and efficient resampling methods

The phenotype vector

The phenotype vector, also called "trait vector" (as we call it), will be, in our case, a vector whose elements y_i (as in equation (3.2) are $y_i = 0$ if individual i is a control (supposed not having the disease of interest), or $y_i = 1$ if individual i is a patient having the disease of interest. But let's notice that the vector could be continuous and account for more complex traits. We decided to focus on binary traits because it seems to be more requested by investigators in the GEHU lab. Even though our implementation is already functional and ready to handle a given continuous trait vector. Indeed, it is possible to enter it as an input parameter. If no phenotype vector is given (which is always the case for the analysis we ran), the latter is automatically computed via the data retrieval step described before.

The covariates and the model

The chapter 3 showed that the model used can easily adjust for covariates. Here again, we did not run any analysis including covariates. But our implementation already contains a block responsible for creating automatically a covariate matrix given Highlander information or simply by taking into account covariates matrices directly created by the investigators. Notice that, adjusting for covariates implies a prior knowledge of the disease and reliable

[11] A complete description of the following functions can be found at https://cran.r-project.org/web/packages/SKAT/SKAT.pdf

data. Because we focused on the worst case scenario (total unknown disease and underlying genetic model), we did not run any experiment with covariates.

Because we never included any covariates, our model was always the simplified version of (3.1) such as:

$$logit(\mu_i) = \alpha_0 + h(\mathbf{G}_i)\beta \tag{4.13}$$

the weighting procedure

This simply corresponds to parameters *weights.beta* and *weights* in both *SKAT* and *SKAT-Binary* functions. The first parameter simply corresponds to a numeric vector for the β weights used in the several weighted kernels. This parameter will be ignored if the second one is not null. The latter is used, if and only if, the user wants to use particular weighting procedure (as seen in section 4.1.2) or even its own designed weights given the intuitions and hypothesis that has to be challenged.

Null model and the combination of tests

The null model is easily computed by one of the two built-in functions of the package, namely *SKAT_Null_Model* and *SKAT_Null_Model_MomentAdjust*. We designed our implementation such that it will adapt itself and use the most adequate one. In other words, if the sample (number of cases and controls together) size is less than 2000 [32], we automatically use the second function which has been designed to handle this small sample situation [33]. As explained above, we based our analysis on the simplified model (4.13) which corresponds to a simple parameter choice.

In order to be able to attack unknown underlying genetic models, we choose to implement all possible tests available in the package with an easy way to adapt it to the current hypothesis being tested. This means that we implemented all feasible and functional combination[12] of: kind of test (Burden, variance-component, SKATO), type of kernel (see section 4.1.2) and all methods for p value computation (see section 4.1.3). Moreover, we do this for both *SKAT* function and its optimized counterpart for binary trait *SKATBinary*. We also developed an optimized version of this script including only the recommended tests (this optimized version results of the different tests we run, see next chapter).

The following chapter will present our results and recommendations. It will also be the opportunity to present the last step of the analysis of the pipeline described in the last section

[12] See next chapter for more details about the combinations and their performance

of this chapter which is the results presentation and how the desired information is presented to the investigator.

Chapter 5

Results

This chapter will present our main results obtained throughout the year[1]. Here we develop a more detailed characterization of each possible test supported by the SKAT package. Indeed, through the literature, there are no investigation comparing the behavior of several tests or even simply confront the computational efficiency of each of them. Actually, only very few of the supported tests have been illustrated (through articles and very few examples in the tutorial). As said previously in this paper, most of the analysis available in the literature are power analysis. The objective of all the test that will be presented bellow is to select create a battery of test, computationally efficient, and representative of all the available class of tests existing. On the other hand, it is also the opportunity to verify that the new framework is indeed efficient and able to hasten the work of the investigator while providing useful and clear results.

In the meantime, we seize the opportunity to test some of our hypotheses listed bellow:

1. Given all the possible combinations of tests, there are probably redundant ones that we can get rid of (e.g 2 tests giving always the same really close p value).

2. Suppose some tests having, in most of the time, the same behavior: their p value are close, no matter the data. One of them, namely the least computationally expensive, should be used without having to use the others at all.

Important preliminary remark

In order to facilitate the results presentation, we will only focus here on the main results whereas some other detailed results will be exposed in the appendix B and C. Moreover, as the

[1]Note that the experiments presented here cover only a small part of the severals tests that have been made during the year.

reader will see, because of the important diversity of tests, we choose to refer to each of them thanks to a specific given name. We greatly suggest that the reader starts reading appendix *A*.19 at first - "A brief explanation of each test's name meaning" - to better understand the results presented bellow. Indeed, in order to simplify the presentation and focus on the results' interpretations, we will not explain the tests involved for each figure . Each time, names will be presented on the right side of the figure, and we suppose that the reader is able to understand the meaning of it. Last but not least, when trying to characterize each test, we will always be referring to its output value, namely a p value, and its computational time performance. The latter is of primary importance knowing that genetic data is probably one of the most complex data.

When dealing with time (e.g. computational time analysis), all the values are in seconds (running time of the test).

5.1 A global review and characterization of the available tests

This is probably the most important section because this is where the SKAT package will be explored and exposed to the reader. As mentioned above, we will do a computational time analysis and output value analysis in parallel. We pay attention to only present the most important and interesting results while avoiding to loose important information or drown the reader into too much information. We will only expose the global results, while more detailed information can be found separately for each class of test in the drive[2]

This section is the opportunity to test every single possible forms of tests available. We will also verify if tests truly respond to data change, or if some of them seem to have a classical behavior (e.g. a test giving always p values in a same range of values will be useless).

The computational time analysis is constructed as follow: as input data (namely phenotype vector and genotype matrix), we take the data given as example in the SKAT package and used through the tutorial[3]. To see how different tests behave, we simply run several experiments using growing genotype matrix. Therefore, we will double the size of the input parameters, starting with a genotype matrix $G^{67\times67}$ for the first experiment and ending up with a $G^{1072\times1072}$. We use a simple naive form in order to update the input parameter. On

[2]see folder *Computed_value_analysis* and *Methods_classification* in https://drive.google.com/open?id=0B-vvkjCEwvYkQ2w1UHhEODA4d2M

[3]Tutorial available here: http://docplayer.net/42914914-Skat-package-seunggeun-shawn-lee-june-6-2016.html

the one hand, the phenotype vector is simply updated as $y^{(i)} = (y^{(i-1)}y^{(i-1)})^T$, on the other hand, the genotype matrix is updated as follow:

$$G^{(i)} = \begin{pmatrix} G^{(i-1)} & G^{(i-1)} \\ G^{(i-1)} & G^{(i-1)} \end{pmatrix} \tag{5.1}$$

As expected, using this naive update, the matrix will form easy groups to handle and to discriminate. Therefore, we expect all tests to output really low p values (this is indeed observed and shown in several graphs see in the drive). The main interest of this analysis is to see the behavior of the tests with similar input but increasing size.

The output analysis is constructed as follow: starting again from the data given as example in the SKAT package, we run several experiments with, this time, the same genotype matrix, but also a randomly modified phenotype vector. This can be seen as a change in the group of patient and controls. Thus, this time we do not expect any particular evolution in the computational time. On the other side, we clearly expect to see a different output value for each run. Obviously, the first objective is to be able to retrieve the same results as the authors did (even though they did not provide many examples), but also see if there are any similar behaviors between different tests.

Each experiment consists of computing the p value using different tests while measuring the time consumed by each of them. We run three types of experiments: the first one focuses on exploring the different kernels (see 4.1.2) available, the second one explores all the p value computation methods (see 4.1.3). Last but not least, the last experiment takes all the possibilities of tests in order to obtain a global overview of the performance. Thanks to the last experiment, we suggest here, for the first time, a detailed view of every tests supported until now by the package. Moreover, we will not only state their existence, but also categorize them according to some criteria.

5.1.1 Testing the kernels

The main point here is to see how a different choice of kernel might impact on the performance. In order to let the kernel choice be the single difference among all the tests, we choose to compute the p values thanks to the *SKAT* function, setting parameter *method* to davies for the burden and skat test. Note that one should pay attention to the fact that SKATO tests are using a different p value method computation, therefore the observed difference cannot be totally given to the kernel choice. Nevertheless, it is meaningful to see how they behave compared to each other.

The main results are presented in appendix $B.1.1$, and supplementary material can be found on the drive (See *Computed_value_analysis* and *Methods_classification*). Figures $B.1$ and $B.2$ summarize the results. For more clarity, we present our main conclusions for each class of tests and each analysis in the figure bellow. They are drawn from the results presented in figures $B.3 - 6$

Kernel analysis	Computational	Value
Burden	Kernel choice do not seems to have an impact. Burden tests do not tend to suffer from the increasing size of inputs, which is an interesting feature.	Kernel choice seems to play an major role. The tests differ by the p value returned, but also by the way they react to the new data (rate of change between two point)
Variance-component	it seems that, both linear and weighted linear kernel are slight slower than the weighted quadratic, the weighted IBS, the linear IBS and the 2wayIX. But as the input grows, those are outperforms by the 2 first ones.	Here 3 different behaviors can be observed. First of all, the linear weighted and the weighted IBS kernels seem to behave in a similar way (close p values and rates of change). The same comment applies to the 2wayIX and the weighted quadratic kernels. We can add that the linear kernel, in general, tends to follow these two other in the direction, but do not gives the same p value. The last and unique behavior is the linear IBS.
SKATO	Kernel choice do not seems to have an impact.	Kernel choice seems to play an major role. The tests differ by the p value returned, but also by the way they react to the new data (rate of change between two point)

Fig. 5.1 Summary table of the main observations resulting from both the output value analysis and the computational time analysis for each class of tests.

Globally, it seems that both linear and linear weighted kernels, no matter the class of tests, seem to greatly outperform the other choices of kernels. Our second hypothesis (see the chapter 5's introduction) seems to be verified for linear weighted and the weighted IBS but also for the weighted quadratic and the 2wayIX.

More substantially, the analysis allows to confirm what has already been written [62, 4]: burden tests are faster than SKAT which are faster than SKATO. But this is not due to kernel choice but because SKATO uses an heuristic method (optimal adjustment) in order to compute the p values (see [33]).

The next section will present now the same analysis, but this time challenging all the p values computation method. It will be the opportunity to see if all the observed behaviors above will remain unchanged in this case.

5.1.2 Testing the p values computation methods

The main point here is to see how a different choice of p values computation method (both in the *SKAT* and *SKATBinary* function) can impact on the performance. Each test will use a linear kernel. The main results are presented in appendix *B*.1.2, and supplementary material (e.g detailed results for each class of test) can be found on the drive (See *Computed_value_analysis* and *Methods_classification*). Figures *B*.7 and *B*.8 summarize the results. Here also we based our main comments (presented figure 5.2 based on the data of figures *B*.9 – 12.

P value computation method analysis	Computational	Value
Burden	The ER and QA method performs poorly while the other methods do not suffer from input size augmentation.	The p value computation method clearly do not impact on the result.
Variance-component	Again ER, but this time followed closely by the ER.A method performs poorly. The QA again suffers from input size augmentation. The other methods do perform well, but the UA achieve the best result.	The behavior are close and similar, but this time p value computation method has a small impact.
SKATO		As in the Burden case, the choice do not have any impact on the output value.

Fig. 5.2 Summary table of the main observations resulting from both the output value analysis and the computational time analysis for each class of tests.

First of all, the attentive reader will have noticed that the output value analysis does not include all the tests present in the computational time one. This is surprisingly due to the fact that these tests do not work and throw errors, even while performing on the example data available within the SKAT package. As said previously, when performing an association study, it is not possible to predict the resulting data structure (namely the genotype matrix), therefore, we recommend to avoid the use of these kinds of tests.

Computationally, again, Burden tests are globally faster than variance-component ones, which are in turn faster than SKATO.

Figure *B*.8 shows something really interesting. Even though the data where generate randomly, it illustrates the fact that SKATO are a linear combination of Burden and variance-component tests. In these 5 experiments, SKATO tests tend to react in the same way as Burden tests.

5.1.3 A global tests characterization

Crossing the two previous sections, we suggest an overview of all the combinations of tests available.

The main results are presented in appendix *B*.1.3, and supplementary material (e.g detailed results for each class of test) can be found on the drive (See *Computed_value_analysis* and *Methods_classification*). Figures *B*.13 and *B*.14 summarize the results. Here also we based our main comments (presented figure 5.3 based on the data of figures $B.16 - 21$.

Global analysis	Computational	Value
Burden	4 different observed behavior, but only the ones that are able to keep their running time under control are of interest.	3 interesting results : there are only 2 different categories of tests, namely the linear and the weighted. These categories can sometimes react in opposite directions. Moreover, they often give very different p values.
Variance-component	Even though most of the tests quickly become time consumming, a few of them achieve to keep computational time under control.	Here we identify 4 categories in order to capture all the possible behavior, even though there are only 2 different behavior.
SKATO	More than half of the test do control their running time while the other rapidly become computationally expensive.	Same observations as in the Burden case.

Fig. 5.3 Summary table of the main observations resulting from both the output value analysis and the computational time analysis for each class of tests.

Even though the size of the input is playing a major role in the running time discrimination, one should also note that some tests can be computationally more expensive given different data structures (e.g. non sparse genotype matrix, which does not occur in practice). Such behavior can be detected, for example, thanks to a computation time graph while performing the output value analysis (e.g. see figure *B*.15).

The global output value analysis allows us to validate our second hypothesis (see introduction of chapter 5). Here, we clearly identify 8 different classes of tests (2 Burden tests, 4 variance-component tests and 2 SKATO tests illustrated in figure *B*.8) which are representative of all the tests available through the set of input parameter combination.

5.1.4 Summary and recommendations

The conclusions and recommendations made in this section are the results the combinations of the several observations done above. Thanks to the second analysis (output value analysis) that has validated our second hypothesis, we are able to present to the reader well defined categories. Within these, tests will always output extremely close p values. On the opposite, depending the input data, each category might output very different values. Through figure

5.4 but also figures $B.16-22$, in order to highlight our recommendations, we used a coloring system such as[4]:

1. *Green*: recommended test representative of its category and least computational expensive.
2. *Blue*: tests that can handle any genotype matrix and good running time.
3. *Yellow*: tests that can handle any genotype matrix but time consuming.
4. *Orange*: tests than might throw error given certain input but are computationally efficient.
5. *Red*: tests than might throw error given certain input and time consuming.
6. *Brown*: Not tested.

	SKAT				SKATBinary						
	davies	liu	liumod	optimal.adj	ER	QA	MA	UA	ER.A	Hybrid	
Linear	0,7683	0,6482	0,,4482		86,8185	21,1034	0,7076	0,3555	0,4278	0,8614	Burden (r.corr = 1)
Linear weighted	0,6271	0,5667	0,4461		88,3558	20,6071	0,7662	0,2991	21,4738	0,8775	
weighted quadratic											
Linear IBS											
weighted IBS											
2wayIX											
Linear	6,1967	5,4592	5,1700		517,9163	100,9668	2,6265	0,4001	352,0439	4,6478	SKAT (r.corr = 0)
Linear weighted	6,0109	5,0790	5,8051		393,6553	100,8938	2,6351	0,4010	367,0143	4,6667	
weighted quadratic	6,5505	1,7783	2,2032		402,8528	98,8437	2,6471	5,4693	381,3133	6,8898	
Linear IBS	6,8245	1,6961	2,0592		407,0277	97,7072	2,6466	5,6155	402,0263	6,2716	
weighted IBS	6,7765	1,7327	1,8311		397,0885	108,8556	2,6314	5,5973	358,6390	6,1831	
2wayIX	7,2640	2,0070	2,0098		400,1610	97,0359	2,6374	5,7822	380,8437	6,4607	
Linear				13,7816	435,9695	237,0908	10,0501	3,1619	398,2389	20,2171	SKATO (0 < r.corr < 1)
Linear weighted				13,2993	440,6764	216,3640	11,6067	2,7775	400,6309	20,9061	
weighted quadratic											
Linear IBS											
weighted IBS											
2wayIX											

Legends :
Recommended tests
Functional and fast in category of value
Functional but slow in catgory of value
Not functional everytime but fast
Not functional everytime and slow
Do not exist
Not fully tested

Fig. 5.4 Summary table of all tested different implementations. The value represents the performance of each test on the biggest size input in the last stage of the computational time analysis performed above

[4]Note that the computational time analysis is summarized via this coloring, while the output value analysis is summarized via the categories

We did not fully test[5] the category in brown for the following reasons: it represents an infinite class of tests (infinite possible value of parameter *r.corr* (or ρ in the mathematical model) between 0 and 1), and we decided to focus on worst case scenarios in this paper (possibly unknown disease) which implies that we do not have any information about the right parameter value to set. Moreover, these are part of the reasons that pushed Lee S. et al [33] to develop the SKATO (represented by *p_linear_skato* ans *p_linear_weighted_skato*).

The details of the category can be found in figure *B*.16. Note that in this figure, there are indeed 2 recommended tests for 6 categories. This is because we want our recommendation to be as flexible as possible. Indeed, this will allow, following the latter categories, to handle both discrete and continuous cases[6]. The detailed recommended tests for each case are listed in appendix *B*.1.4.

Because of the results obtained, we changed our implementation so that it chooses only the optimal combination of recommended tests (only one per category) given the input data (discrete or continuous).

Finally, it is worth to highlight that the SKAT package has not been developed further into the class of omnibus tests. Moreover, we did not find any attempt in the literature to combine a Burden test along with another variance-component test than the SKAT, and not even with a SKAT using a different kernel than the linear or weighted linear. Indeed, the literature has interesting examples we can refer the reader to [61] and [49] seems to explore other directions.

For the following experiment (until the end of this paper, an experiment will refer to a specific choice of filtering options and patients and controls groups) in this paper, and in order to prove that even tests in the *blue* category can be used in our framework (because computationally efficient), we will use the group of tests described in appendix *B*.1.4 that differs from the recommended category.

5.1.5 A validation case: the *CM / CLP* association study

In this section, we present the very first case-control association study we run in order to check the true power of the implementation we set up. The main results are exposed in the dedicated appendix *B*.1.5 (figures $B.23 - 26$) and more detailed information (e.g. graph of computational time, rankings, etc.) can be found on the drive[7]. This validation case had to check whether the battery of recommended tests was able to successfully identify some

[5] Even though we did run several experiments, and some results can be found in the drive. The main observation is that these tests using the linear weighted kernel do not work on any data. On the other end, the ones using linear weighted belong to the blue category.

[6] The *SKATBinary* function will throw an error if inputs are continuous.

[7] In folder *association_testing* (see all folders beginning with *CM_*)

genes that are playing a well recognized role in Capillary Malformation-Arteriovenous Malformation (CM-AVM). Indeed, several mutations within specific genes have been formally identified (e.g. at GEHU laboratory, Mustapha et al. has identified mutations in gene $EPHB4$ [45]). Therefore we can use these confirmed information in order to check the correctness and powerfulness of our framework. The set up of this analysis (filters, number of case/control, etc.) can be found in appendix $B.1.5$.

Out of the 13 genes, we successfully identified 10 of them while 3 of them were not identified (namely *DLL4*, *HEY2* and *NOTCH4*). The fact that we did not succeed to identify the latter might be caused by several factors (e.g. see all errors factors presented in chapter 2). But more importantly we notice the importance and powerfulness of our framework. Indeed, the results (and therefore the identified genes) are drastically dependent on the filters used. We prove the importance of running all category of test using different filtering criteria. The output of the analysis does not depend on the investigator's luck while running a single test. Indeed, the latter has a really small probability of having interesting results because they depend on too many choices and option that would take a very long time to be fully tested. Moreover, storing, combining and presenting these informations also act as a great barrier for the biologist or doctor.

This clearly shows the importance and powerfulness of our framework. Indeed, the latter allows an easy and automated way to perform an important amount of tests (meaning that it can attack the data under much more point of view) while summarizing all the information in a clear way for the investigator. One can easily notice that, using only the SKAT package as available, it will be time consuming: data retrieval is different for each experiment because of the different filters, possibly not using the most efficient tests. Moreover, the analysis might not be useful either because it will require to test all categories of tests for each gene. Indeed, $EPHB4$ has been detected only by 2 categories. At then end, combining all the information will be an impossible task for the investigator that will experience severe difficulties in interpreting all this information.

Given this analysis, it seems that the SKAT package, and more globally the dedicated statistical test under study in this paper, seems to loose tremendous power if used as a one-shot experiment. Indeed, for these tests to work well, several hypotheses must be explored. The latter are translated by the exploration of different filtering methods and criteria.

We therefore make a global recommendation: case-control association testing for region based test should always be integrated in a more global environment (allowing to perform several tests, and also taking charge of collecting and more importantly presenting in a clear way) and should be directly linked to a filtering tool (allowing the data retrieval automation

and flexibility). In this way, all the power of these tests can truly be reached and used in order to guide and help the investigator in his research.

5.2 A real data application

This section is based on an analysis that is currently taking place at GEHU laboratory[8]. The main results are exposed in the dedicated appendix *C* and more detailed information (e.g. graph of computational time, rankings, etc.) can be found on the drive[9].

Results of the computational analysis are illustrated in figure *C*.1 and confirm that the set of test used, even though they are not the recommended ones, remarkably behave and control their time consumption. This allows us to confirm that not only tests belonging to the *green* category, but also the ones of the *blue* can be used for real data exploration.

Results of the p values analysis are illustrated in figure *C*.2. It clearly confirms what has been shown in the different analyses done previously: each test drastically responds to data[10].

A new way to present information was introduced here on the demand of the investigator. Now, the framework also outputs the rankings (one for each class of tests, but also a general one). The different rankings obtained can be seen in figure $C.3-6$. The latter are simply an average of the rank given by the different individual tests. It shows up, as said previously (but also predicted by the theory), that each class of tests gives a different ranking. And given the previous analysis, we clearly understand the importance of having separated rankings. Indeed, some genes might never be detected by a class of test, simply because the latter is based on hypothesis that cannot match the genetic activity in the gene that is causing the disease.

These rankings are useful to compare the various experiments. They can be found in figure $C.7-10$. Comparing these allows to confirm the impact of filtering on one hand, but also to discover another (yet predictable) source of variability in the results on the other: the case-control design (explained in more details in appendix *C*). Here, the only advise would be to try the same experiment with possibly different controls (not only one fixed control group). This is clearly the most challenging problem for the moment. Indeed, the ideal case would be to have at disposal a data base of "healthy individuals" as well annotated as the one we have for the patient. But knowing[11] that the costs to obtain a whole-exome sequence

[8] Ziyad S et al. (in preparation), an abstract and **confidential** excerpt from the paper made entirely by Nisha Limaye are available in appendix *C*.1

[9] In folder *association_testing* (see all folder beginning with *Test_*)

[10] Indeed, one of the fear was that some tests might always give p values in the same range (e.g in between 0.2 and 0.5) which will make them totaly useless

[11] https://www.genome.gov/27565109/the-cost-of-sequencing-a-human-genome/

is now generally below 1000 $ (around 600 $), it would be costly. More importantly, these kinds of data are classified (and with reasons) as one of the most sensitive. Therefore it will require first an ethical debate on a societal level, but also secure cryptographic scheme to protect them without introducing new errors[12].

As a global summary, we present all the experiments and all the tests within a single table, in order to show the number of genes (region of interest) that has passed the test with a p value less than 5 %. Information is illustrated at figure *C*.11. Obviously, the more the filtering is restrictive, the least genes passed the test. More significantly, we notice a difference between similar experiments that only differs by the control group used. But one needs to be careful with the interpretation. On the one hand, this could imply that a more homogeneous group could help obtaining more significant results. But on the other hand, if the diseases present in the two groups (patients and controls) do share some genetic structure, then better results might be obtained with the heterogeneous controls.

The last row in tabular (see figure *C*.11) shows that the five experiments, each of them using 8 tests 93 times (for each position, namely gene) make a total of 3720 tests running in less than 17,65 minutes (one test performed on one gene in one particular experiment takes approximately $0,0047$ seconds). Note that this is not even the fastest tests (*blue* category) that has been used here.

As a conclusion, this analysis ensures that we indeed proposed a well functional and automated frameworks to run (important numbers of) tests with a huge flexibility and a special care for data presentations in order to truly help the investigator.

This analysis also allowed to point out the problem and effect on results of case and control construction.

To go further into the work presented in this paper, we can add that this series of experiments presented here have led to another module that we developed in order to assess the pair of genes enrichment [13]. This pair enrichment module is described in appendix *C*.7. The latter should be developed deeper and integrated into the framework in the future, in order to test potential pair or group of genes interaction based on the results obtained through the association analysis.

Throughout this chapter, we did test a huge range of possibilities and hypotheses that have helped, given the results obtained at each stage, to build the automated framework. It is clear that real data exploration will help improve the interpretation of data. One of

[12]It is not the point of this paper to go further into this debate, nevertheless, the reader can find a useful reading here [25] about cryptography. Knowing that, nowadays, no scheme is fully secure, and the most efficient scheme might introduce uncertainty in the data, which would be pointless.

[13]Result available in the drive, in folder *Pari_gene_enrichment_testing*

the main next step is to convince the scientific[14] community to give access to their data for further validation (both for the tests but also for the research). Note also that there are still unexplored parameters such as the weighting procedure and the model (possibly including covariates) that can help to adapt each test so it fits even better the disease model. Indeed, in this chapter, we considered only the worst case scenario in wich very few information is known about the true disease model, therefore we did not focus on these parameters.

As said during the above discussion, it is clear that one should not rely on one single experiment, but on different ones, changing some parameters. One of the next steps in the development of our framework will be the integration of all these parameters coming from different experiments into a single one. The good news is that some authors of the SKAT package (Lee S et al. [61]) have already started developing such tools.

Last but not least, the next chapter will summarize our one-year investigation, our main results and recommendations, and will end up with some points to improve and questions to explore on order to go further.

[14] We tried to get access to data coming from [8, 29, 46, 41] without success for the moment.

Chapter 6

Conclusions

In order to conclude this master thesis, we want to give the reader a global overview of the structure of the paper, followed by a summary of our main contributions and recommendations and ending with some improvements tracks.

At the beginning, the objective of this master thesis is to give investigators of the Institute an easy access to a statistical test for association of rare variants taking into account information coming from Highlander's database. Therefore, this paper reviews the fundamentals of genomics required in order to be able to understand the origin and huge complexity of data coming from next generation sequencing. Then it introduces the concept of variants' annotation and filtering along with Highlander, a genetic database for variants' filtering developed by the bioinformatician of the Human Molecular Genetics laboratory (Helaers R.). This was followed by a literature review of all available statistical tests and classes that end up with a specific choice of software package, namely the SKAT. While giving a global overview of the latter, its choice was also motivated. Then, the specific framework developed throughout the year is covered and particular implementation choices are justified. We end up presenting a series of analysis from which we have drawn several conclusions and recommendations. Indeed, real data applications have been a great source of information that have led the implementation to its final form, which might (with a high probability) still evolve in the future.

Main contributions and main results

We suggest a useful framework in order to help and guide the investigators trough the identification process (in this paper, we mainly presented it as a gene identification, but it could be any other genetic region). Thanks to our analysis, we succesfuly classified the different available tests into well defined categories (see figure 5.4 and *B*.22). Our first real

data application pointed out the fact that even though region based tests for association of variants are powerfull, they might not correctly identify the genetic region playing a role in the disease under study. In order to return the desired region, the investigator, without much prior knowledge about the disease, should select the right category of tests with the optimal filter and the correct case-control design, which do not occur often in real life. Therefore, we proposed a statistical framework based on optimized (least computationaly expensive, and representative of all the categories) choices available within the SKAT package. Because our framework is directly connected to Highlander database, the investigator simply has to specify the different filters (and values) and the different case-control designs that have to be explored. Automatically, our framework will run the several selected statistical tests for all parameter values and the different scenarios (e.g. using different control groups) and present all the results, classified per experiment, via excell tables and graphs containing the different p values of each test, but also a global ranking and one specific ranking per class (Burden, variance-component and SKATO tests). We showed that this framework allows to drastically hasten the work, make interpretation of results much easier, and improve the results obtained compared to a single "one shot" use of the SKAT package. Thanks to our new framework and its interaction with Highlander (offering all the power of variants' filtering), we are able to remove the "chance" and time factors that might stop a single investigator from using such association tests. Moreover, in order to give the best possible answer to GEHU investigators, the implementation must have the ability to handle even the worst case scenario in which the sample size is small and no prior information (e.g. covariates) is known. Moreover it must be capable of handling different types of diseases (e.g. different underlying genetic structure). This is why we only rely on model (4.13) and whitout any covariates or specific weighting procedure.

We suggested a new statistical framework for testing association of genetic variants in order to fasten and guide the investigator through the identification phase which is the starting point of all new treatments. As Christian de Duve said:

"To overcome disease, one must first understand it<."

To go further

Due to the fact that it is only a one year master thesis, our work was limited and we could not explore everything we wanted. Therefore, we conclude this paper with a list of observations and ideas that we did not have the time to develop here but that could form the starting point of several future investigations.

Regarding the SKAT package

As mentioned in chapters 4 and 5, we did not test all possibilities offered by the SKAT package. Indeed, we focused on discrete phenotypes, while making the framework ready for continuous traits but without making any further exploration in this kind of phenotypes. Moreover, one must explore the behaviors of the tests while modifying three things that might have an impact on the results: the model used, the covariates and the weighting procedure. All these parameters imply a prior knowledge which is why, here, we did not explore them. Indeed, we focused on worst case scenarios in which we do not have any information.

We still have to explore the *SKAT_commonRare* function for combined effect of common and rare variants that is well described in [22].

On the other hand, another urgent question following our work here, is to find out how to combine the data from the various experiments. In order to tackle that question, one possible starting point could be the already existing framework for meta analysis, already available in the SKAT package [61].

Regarding our new framework

We need to implement a user interface into Highlander so that the framework will be fully integrated and could be directly used by the investigators. This interface should be a simple list of boxes to check, easy to use, and letting all the necessary options open (e.g. discrete or continuous traits, cases and controls, etc.).

A particular effort and research should be done in order to improve the data visualization and presentation. For example, an interactive graph showing the ranking, but allowing to display other information when the user places the mouse on a particular point in the graph (e.g name of the gene, all subsidiary rankings and p value, etc.).

In order to obtain more meaningful ranking, one idea would be to use the power computation methods available in the package. The latter allows to compute an average power of SKAT and SKATO for testing association between a genomic region and continuous or binary traits with a given disease model. If each output p value is weighted given that computed power, it might increase the precision of the obtained ranking.

More substantially, one concrete improvement track would be to integrate and to use the information coming from a pathway-driven gene enrichment analysis (still under development in the GEHU laboratory). Indeed, the latter allows to get insight into the underlying biology of genes that have different level of expression.

In general

One can also explore the more general question of creating (using the same idea as the SKATO with the linear and linear weighted kernels) a new omnibus test that will be a combination of a Burden test and a SKAT test using one of the other kernels (namely the linear and linear weighted IBS, weighted quadratic and the 2wayIX).

Machine learning methods could also help in order to, one day, make it possible for this kind of framework to learn from its mistakes and - why not - even make assumptions on the role of genes via all the information the algorithm will have faced, all the experiments it will have run, and the feedback given by the users.

A general problem that would slow down the research in genetic would be a lack of a true database of controls made of "healthy" people. Even though sequencing DNA became less expensive over the years, we still do not possess the required cryptographic scheme in order to ensure a total protection of these sensitive data. Moreover, it is necessary to have an ethical debate in front of that, in order to decide who should keep these data or have access to them. It would be a very complicated and serious task, but this might drastically push research forward.

As a final recommendation, the biomedical research constitutes a vast and wonderful area for a mathematical engineers looking for endless challenges and willing to have a significant impact on the world. Therefore, we should put effort on spreading the words and start attracting new engineers into the field.

"Think about the whole world"
"Keep your mind open to all possibilities"
"Stay simple"

References

[1] Alberts B, Johnson A, L. J. e. a. (2002). Molecular biology of the cell. 4th edition. *New York: Garland Science*, (From DNA to RNA).

[2] Auer, P. L. and Lettre, G. (2015). Rare variant association studies: considerations, challenges and opportunities. *Genome Medicine*, (7:16).

[3] Bamshad MJ, Ng SB, e. a. (2011). Exome sequencing as a tool for mendelian disease gene discovery. *Natural Reviews*, (12(11):745-55).

[4] Basu S, P. W. (2011). Comparison of statistical tests for disease association with rare variants. *Genet Epidemiol*, (35(7):606-19).

[5] Boomsma DI, Wijmenga C, e. a. (2014). The genome of the netherlands: design, and project goals. *Eur J Hum Genet*, (22(2):221-7).

[6] Clarke LA, Rebelo CS, e. a. (2001). Pcr amplification introduces errors into mononucleotide and dinucleotide repeat sequences. *Molecular Pathology*, (54(5):351-353).

[7] Cooper, Geoffrey M., H. R. E. (2007). The cell : A molecular approach, 4th edition. *ASM Press, Washington, DC*.

[8] Cybulski C, Lubiński J, e. a. (2015). Mutations predisposing to breast cancer in 12 candidate genes in breast cancer patients from poland. *Clin Genet*, (88(4):366-70).

[9] Daniel L. Hartl, E. W. J. (1998). Genetics, principles ans analysis, fourth edition. *Jones and Bartlett Publishers, Subdury, Massachusetts*.

[10] Derkach A, Lawless JF, S. L. (2013). Robust and powerful tests for rare variants using fisher's method to combine evidence of association from two or more complementary tests. *Genet Epidemiol*, (37(1):110-21).

[11] Genomes Project C, Auton A, e. a. (2015). A global reference for human genetic variation. *Nature*, (526(7571):68-74).

[12] Gilissen C, Hoischen A, e. a. (2012). Disease gene identification strategies for exome sequencing. *Eur J Hum Genet*, (20(5):490-7).

[13] Groving, P. M. R. (consulted on 10-02-2017). Each and every cell in our body contains dna, including blood. if we transfer blood from one to another, why is there no conflict with their dna? how is it easily mixed? *www.quora.com*.

[14] Hall, N. (2007). Advanced sequencing technologies and their wider impact in microbiology. *The Journal Of Experimental Biology*, (Vol 209 (Pt 9), 1518–1525).

[15] Han F, P. W. (2010). A data-adaptive sum test for disease association with multiple common or rare variants. *Hum Hered*, (70(1):42-54).

[16] Hartl, D. L. (2000). A primer of population genetics, third edition. *Sinauer Associates Inc., Sunderland*.

[17] Helaers, R. (2016a). An ensemble method for predicting the pathogenicity of rare missense variants. *Journal Club*.

[18] Helaers, R. (2016b). Ngs data analysis : enhanced pipeline, new database and dedicated gui.

[19] Hoffmann TJ, Marini NJ, W. J. (2010). Comprehensive approach to analyzing rare genetic variants. *PLoS One*, (5(11):e13584).

[20] Inc., I. (consulted on 22-02-2017). An introduction to next generation sequencing technology. *www.illumina.com*.

[21] Ioannidis N. M., Rothstein J. H., e. a. (2016). Revel: An ensemble method for predicting the pathogenicity of rare missense variants. *Am J Hum Genet*, (99(4):877-885).

[22] Ionita-Laza, I., L. S. e. a. (2013). Sequence kernel association tests for the combined effect of rare and common variants. *American Journal of Human Genetics*, (92, 841–853).

[23] Ionita-Laza I, Lee S, e. a. (2013). Family-based association tests for sequence data, and comparisons with population-based association tests. *Eur J Hum Genet*, (21(10):1158-62).

[24] John Reif, N. T. and Lavery, R. (consulted on 20-02-2017). Introduction to dna. *https://users.cs.duke.edu*.

[25] Katz, J. and Lindell, Y. (2007). Introduction to modern cryptography. *CRC press*.

[26] Kensuke Nakamura, Taku Oshima, e. a. (2011). Sequence-specific error profile of illumina sequencers. *Nucleic Acids Research*, (39 (13): e90).

[27] Kimberly Robasky, e. a. (2013). The role of replicates for error mitigation in next-generation sequencing. *Nature Reviews, Genetics*, (1-7).

[28] King CR, Rathouz PJ, N. D. (2010). An evolutionary framework for association testing in resequencing studies. *PLoS Genet*, (6(11):e1001202).

[29] Kraus C, Hoyer J, e. a. (2017). Gene panel sequencing in familial breast/ovarian cancer patients identifies multiple novel mutations also in genes others than brca1/2. *Int J Cancer*, (140(1):95-102).

[30] Kristel Van Steen, P. (consulted on 22-02-2017). Genetics and bioinformatics. *Montefiore Institute - Systems and Modeling GIGA - Bioinformatics ULg, slide lecture 5*.

[31] Lawrence, M. (2014). Introduction to variant calling. *https://bioconductor.org*.

[32] Lee, S., E. M. e. a. (2012). Optimal unified approach for rare variant association testing with application to small sample case-control whole-exome sequencing studies. *American Journal of Human Genetics*, (91, 224-237).

[33] Lee, S., W. M. C. and Lin, X. (2012). Optimal tests for rare variant effects in sequencing association studies. *Biostatistics*, (13, 762-775).

[34] Levinson G, G. G. (1987). Slipped-strand mispairing: a major mechanism for dna sequence evolution. *Mol Biol Evol*, (4:203–21).

[35] Li, F. (2016). Introduction to rare-variant association analysis.

[36] Li H, Handsaker B, e. a. (2009). The sequence alignment/map format and samtools. *Bioinformatics*, (25(16):2078-9).

[37] Limaye N, Kangas J, e. a. (2015). Somatic activating pik3ca mutations cause venous malformation. *Am J Hum Genet*, (97(6):914-21).

[38] Lin, D.-Y. and Tang, Z.-Z. (2011). A general framework for detecting disease associations with rare variants in sequencing studies. *The American Journal of Human Genetics*, (89, 354–367).

[39] Lin S. Chen, Li Hsu, e. a. (2012). An exponential combination procedure for set-based association tests in sequencing studies. *Am J Hum Genet*, (91(6): 977–986).

[40] Liu, D. J. and Leal, S. M. (2010). A novel adaptive method for the analysis of next-generation sequencing data to detect complex trait associations with rare variants due to gene main effects and interactions. *PLoS Genet*, (6(10)).

[41] Lolas Hamameh S, Renbaum P, e. a. (2017). Genomic analysis of inherited breast cancer among palestinian women: Genetic heterogeneity and a founder mutation in tp53. *Int J Cancer*, (141(4):750-756).

[42] Magdalena Szumilas, M. (2010). Explaining odds ratios. *J Can Acad Child Adolesc Psychiatry*, (17:117–30).

[43] McClellan J1, K. M. (2010). Genetic heterogeneity in human disease. *Cell*, (141(2):210-7).

[44] Mount, D. W. (2004). Bioinformatics, sequence and genome analysis, second edition. *Cold Spring Harbor Laboratory Press, New York*.

[45] Mustapha Amyere, Nicole Revencu, R. H. e. a. (2017). Germline loss-of-function mutations in ephb4 cause a second form of capillary malformation-arteriovenous malformation (cm-avm2) deregulating ras-mapk signaling. *Circulation*, (Volume 136, Issue 7).

[46] Nadine Jalkh, Eliane Chouery, e. a. (2017). Next-generation sequencing in familial breast cancer patients from lebanon. *BMC Med Genomics*, (10: 8).

[47] Neale BM, Rivas MA, e. a. (2011). Testing for an unusual distribution of rare variants. *PLoS Genet*, (7(3):e1001322).

[48] Nicolae, D. L. (2016). Association tests for rare variants. *Annu. Rev. Genom. Hum. Genet.*, (17:117–30).

[49] Oualkacha K, Dastani Z, e. a. (2013). Adjusted sequence kernel association test for rare variants controlling for cryptic and family relatedness. *Genet Epidemiol*, (37(4):366-76).

[50] O'Rawe, J. e. a. (2013). Low concordance of multiple variant calling pipelines : practical implications for exome and genome sequencing. *Genome Med*, (5, 28).

[51] P. Shing Ho, M. C. (2011). Dna structure: Alphabet soup for the cellular soul. *DNA Replication-Current Advances*, (chapter 1).

[52] Passarge, E. (2003). Atlas de poche de génétique, 2eme edition. *Médedine-Sciences Flammarion*.

[53] Petr Danecek, Adam Auton, e. a. (2011). The variant call format and vcftools. *Bioinformatics*, (27(15), 2156-2158).

[54] Price AL, Kryukov GV, e. a. (2010). Pooled association tests for rare variants in exon-resequencing studies. *Am J Hum Genet*, (86(6):982).

[55] Raphaël Helaers, Pascal Brouillard, e. a. (IDONTKNOW). Highlander: variant filtering made easy. *IDONTKNOW*, (IDONTKNOW).

[56] Raphaël Helaers, P. (consulted on 22-09-2016). Le séquencage à haut débit. *Laboratory of Human Molecular Genetics (GEHU)*.

[57] Riken, Y. I. . S. and Center, S. B. (consulted on 02-03-2017). The genetic code. *http://protein.gsc.riken.go.jp/sakamoto/research_en.html*.

[58] Robert L. Nussbaum, e. a. (2007). Genetics in medicine, 7th edition. *Thompson and Thompson*.

[59] Salipante SJ, Kawashima T, e. a. (2014). Performance comparison of illumina and ion torrent next-generation sequencing platforms for 16s rrna-based bacterial community profiling. *Appl Environ Microbiol*, (80(24):7583-91).

[60] Schuster, S. C. (2007). Next-generation sequencing transforms today's biology. *Nature Publishing Group*, (Vol 5 (1), 16–18).

[61] Seunggeun Lee, Tanya M. Teslovich, e. a. (2013). General framework for meta-analysis of rare variants in sequencing association studies. *Am J Hum Genet*, (93(1): 42–53).

[62] Seunggeung Lee, Goncalo R. Abecasis, e. a. (2014). Rare-variant association analysis: Study designs and statistical tests. *The American Journal of Human Genetics*, (95, 5–23).

[63] Soblet J, Kangas J, e. a. (2016). Blue rubber bleb nevus (brbn) syndrome is caused by somatic tek (tie2) mutations. *J Invest Dermatol*, (137(1):207-216).

[64] Strachan, T. and Read, A. P. (1996). Human molecular genetics. *Bios Scientific Publisher*.

[65] Trudy McKee, J. R. M. (2015). Biochemistry : the molecular basis of life, 6th edition. *Oxford University Press*, (chapter 5).

[66] W, P. (2009). Asymptotic tests of association with multiple snps in linkage disequilibrium. *Genet Epidemiol*, (33(6):497-507).

[67] Wessel, J. and Schork, N. J. (2006). Generalized genomic distance–based regression methodology for multilocus association analysis. *Am J Hum Genet*, (79(5): 792–806).

[68] Wu, M. C., K. P. e. a. (2010). Powerful snp set analysis for case-control genomewide association studies. *American Journal of Human Genetics*, (86, 929-942).

[69] Wu MC., Lee S., e. a. (2011). Rare variant association testing for sequencing data using the sequence kernel association test (skat). *American Journal of Human Genetics*, (89, 82-93).

[70] Xu, S. (2013). Genetic variation and natural selection detection. *CAS-MPG Partner Institute for Computational Biology (PICB)*.

[71] Zhang D, L. X. (2003). Hypothesis testing in semiparametric additive mixed model. *Biostatistics*, (4(1):57-74).

Appendix A

Supplementary informations

A.1 Nucleotides and nucleic acids

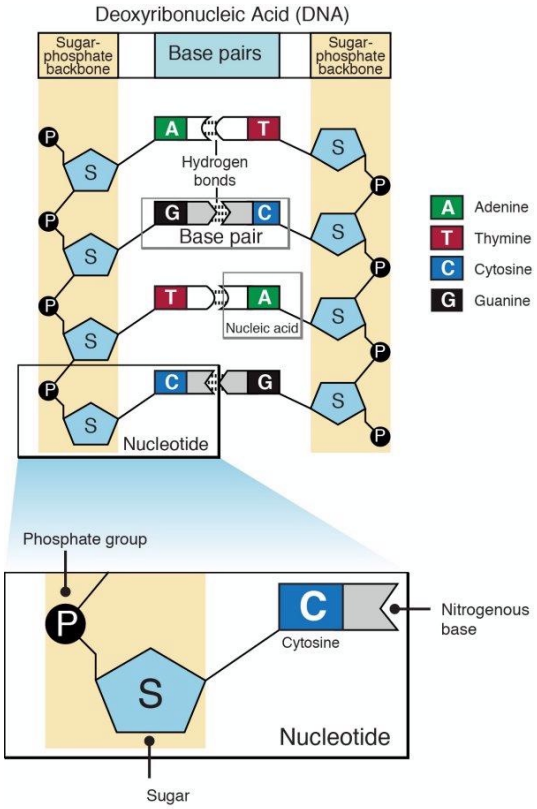


Fig. A.1 Detailed representation of DNA molecule. P stands for phosphate group, S stands for the sugar and the four nucleotides are respectively represented by the letters A, T, C, G. The figure is drawn from http://malemodelspicture.net/

A.2 Amino acids

TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys
TTC		TCC		TAC		TGC	
TTA	Leu	TCA		TAA	Stop	TGA	Stop
TTG		TCG		TAG		TGG	Trp
CTT	Leu	CCT	Pro	CAT	His	CGT	Arg
CTC		CCC		CAC		CGC	
CTA		CCA		CAA	Gln	CGA	
CTG		CCG		CAG		CGG	
ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser
ATC		ACC		AAC		AGC	
ATA		ACA		AAA	Lys	AGA	Arg
ATG	Met	ACG		AAG		AGG	
GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly
GTC		GCC		GAC		GGC	
GTA		GCA		GAA	Glu	GGA	
GTG		GCG		GAG		GGG	

Fig. A.2 The 20 common amino acids of proteins. The figure is drawn from [57].

A.3 Illustration of a chromosome and its inner structure

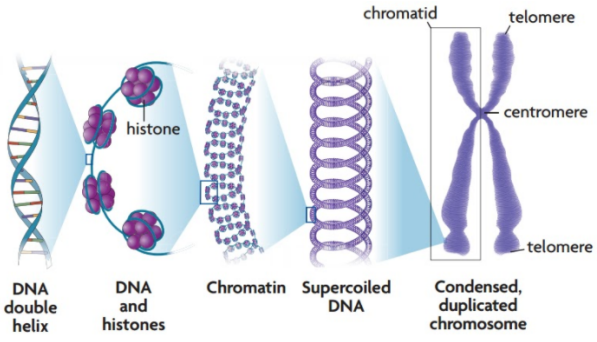


Fig. A.3 From left to right : schematic representation of DNA double helix structure, DNA and histones, chromatine, supercoiled DNA and chromosome. The figure is drawn from http://bkmn.tk/chromosome-structure-diagram/

A.4 Classical representation of a DNA molecule

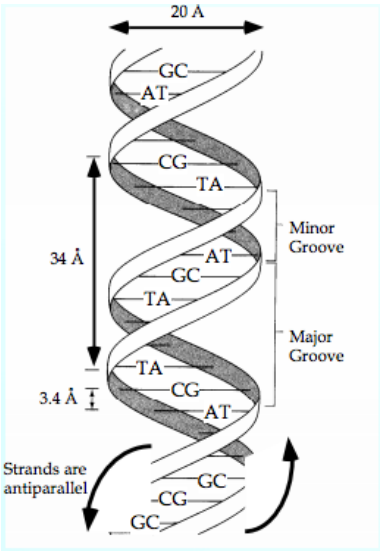


Fig. A.4 Classical representation of DNA molecule in its common form. The four nucleotides are respectively represented by the letters A, T, C, G. The figure is drawn from [24]

A.5 Polymerase chain reaction

Polymerase chain reaction is a repeated cycle of DNA denaturation, renaturation with primer oligonucleotide sequences, and replication, resulting in exponential growth in the number of copies of the DNA sequence located between the primers [9]. In order to activate PCR, 4 ingredients are needed : template DNA, primers, deoxynucleotide triphosphates (dNTPs) and DNA polymerase.

First of all, in order to duplicate DNA, the process needs an original sample of the given DNA sequence to be copied (in blue in figure A.5).

A second and crucial element needed for PCR is primers (represented in green on figure A.5). The latter pieces of DNA that are made in laboratory. Since they're custom built, primers can have any chosen sequence of nucleotides. The objective is to create two primers so that they match the segment of DNA that has to be copied. Through complementary base pairing, one primer attaches to the top strand at one end of your segment of interest, and the other primer attaches to the bottom strand at the other end [52]. Usually, primers have a length of about 20 nucleotides long so that they will target just one place in the entire genome. Primers are necessary because the DNA polymerase enzyme that catalyzes the synthesis of a new DNA strand cannot initiate strand synthesis by its own, but requires this short single stranded primer as a substrate to be elongated [16]. More formally, primers are short nucleic acid sequence, often a synthetic oligonucleotide, which binds specifically to a single strand of target nucleic acid sequence and initiates synthesis, using a suitable polymerase, of a complementary strand [64].

The third element is deoxynucleotide triphosphates. The latter is simply the usual nucleotides (adenine, guanine, cytosine, and thymine) that are linked to three phosphates.

Last but not least, DNA Polymerase. This complex of proteins has the function to copy a DNA before it divides in two. When a DNA polymerase molecule come across a primer that's base-paired with a longer piece of DNA, it attaches itself near the end of the primer and starts adding nucleotides. More formally, polymerase is an enzyme catalyzing the reaction responsible for binding nucleotides in DNA sequence (DNA replication or transcription) [52].

Polymerase chain reaction is a three steps process [64, 52] :

1. Denaturation : dissociation of complementary strands to give single-stranded DNA. Occurs typically at about 93° to 95° C for human genomic DNA.

2. Annealing : association of complementary DNA strands to form a double-stranded structure. The primers come into action and bind to the single stranded sequence.

Occurs at temperatures usually from about 50° C to 70° C, depending on the melting temperature of the expected duplex.

3. DNA synthesis : activation of DNA polymerase in order to elongate the primer with respect to the template strand. Occurs typically at about 70° C to 75° C

Theoretically, and as illustrated in figure A.5, the amount of DNA double at each round [52]. The main interest of PCR lies in the fact that it allows to create a considerable amount of DNA copy (needed for further sequencing and analysis, explained below) at low price[1].

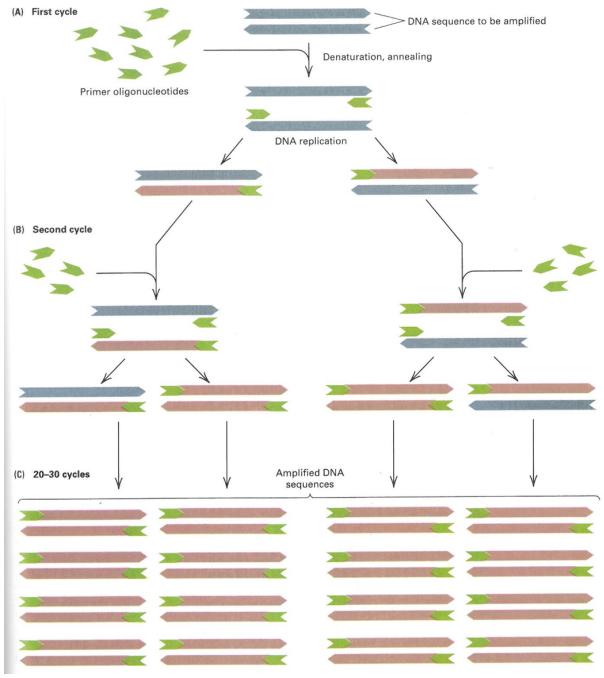


Fig. A.5 Illustration of the polymerase chain reaction cycles for the amplification of a particular DNA sequence. Primers are represented in green, original template of DNA to be duplicated in blue and copies of the latter in pink. The figure is drawn from [9]

[1]Consulted on 01/12/2016, Introduction to Genomic Technologies, Steven Salzberg, PhD & Jeff Leek, PhD `https://www.coursera.org/learn/introduction-genomics`

A.6 Next generation sequencing evolution

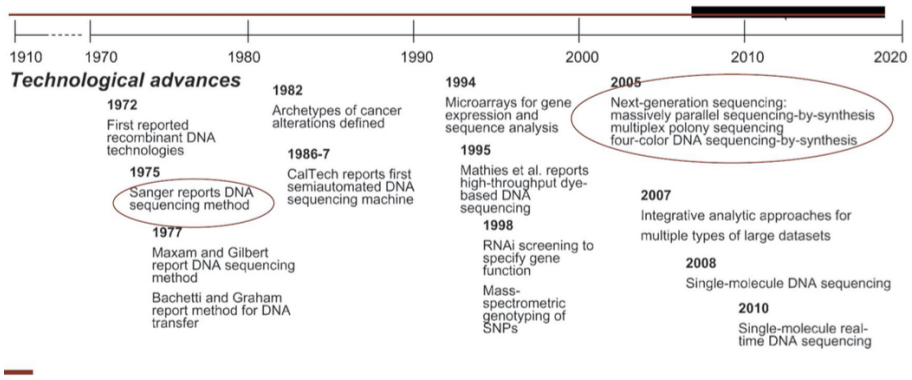


Fig. A.6 Next Generation Sequencing methods evolution time line. Figure drawn from [30]

A.7 Comparison of error in sequencing technologies

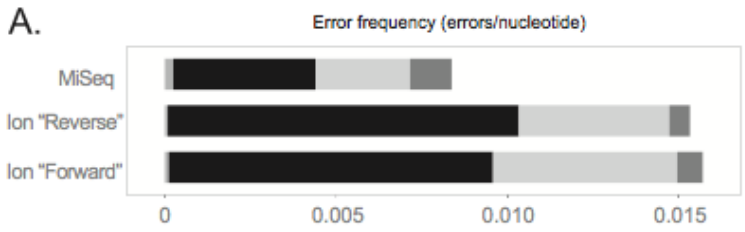


Fig. A.7 Comparison of error rates and error types for assembled paired-end MiSeq reads and unidirectional reads from the Ion Torrent PGM platform derived from a mock-community mixture of 20 bacterial DNAs. Errors for single nucleotide substitutions, homopolymer indels (homoindels), other indels outside homopolymer tracts, and compound errors (event involving two or more categories) are shown separately (figure drawn from [59])

A.8 Illustration of a BAM file

```
@HD     VN:1.0  SO:coordinate
@SQ     SN:chr20        LN:64444167
@PG     ID:TopHat       VN:2.0.14       CL:/srv/dna_tools/tophat/tophat -N 3 --read-edit-dist 5 --read-rea
lign-edit-dist 2 -i 50 -I 5000 --max-coverage-intron 5000 -M -o out /data/user446/mapping_tophat/index/chr
20 /data/user446/mapping_tophat/L6_18_GTGAAA_L007_R1_001.fastq
HWI-ST1145:74:C101DACXX:7:1102:4284:73714        16      chr20   190930  3       100M    *       0       0
       CCGTGTTTAAAGGTGGATGCGGTCACCTTCCCAGCTAGGCTTAGGGATTCTTAGTTGGCCTAGGAAATCCAGCTAGTCCTGTCTCTCAGTCCCCCCTCT
C    BBDCCDDCCDDDDCDDDDDDCDCCCDBC?DDDDDDDDDDDDDDDCCDCDDDDDDDDDDCCCCEDDDC?DDDDDDDDDDDDDDDDDDDDDBDHFFFFDC@@
    AS:i:-15        XM:i:3  XO:i:0  XG:i:0  MD:Z:55C20C13A9 NM:i:3  NH:i:2  CC:Z:=  CP:i:55352714   HI:i:0
HWI-ST1145:74:C101DACXX:7:1114:2759:41961        16      chr20   193953  50      100M    *       0       0
       TGCTGGATCATCTGGTTAGTGGCTTCTGACTCAGAGGACCTTCGTCCCCTGGGGCAGTGGACCTTCCAGTGATTCCCCTGACATAAGGGGCATGGACGA
G    DCDDDDEDDDDDDDCDDDDDDDCCCDDDCDDDDDEEC>DFFFEJJJJJIGJJJJIHGBHHGJIJJJJJJGJJJIJJJJJIHJJJJJJHHHHHFFFFFCCC
    AS:i:-16        XM:i:3  XO:i:0  XG:i:0  MD:Z:60G16T18T3 NM:i:3  NH:i:1
HWI-ST1145:74:C101DACXX:7:1204:14760:4030        16      chr20   270877  50      100M    *       0       0
       GGCTTTATTGGTAAAAAAGGAATAGCAGATTTAATCAGAAATTCCCACCTGGCCCAGCAGCACCAACCAGAAAGAAGGGAAGAAGACAGGAAAAAACCA
C    DDDDDDDDDCCDDDDDDDDDDEEEEEEEFFFEFFEGHHHHFGDJJIHJJJIJIJJJIIIIGGFJJIHIIIIJJJJJJIGHHFAHGFHJHFGGHFFFDD@BB
    AS:i:-11        XM:i:2  XO:i:0  XG:i:0  MD:Z:0A85G13    NM:i:2  NH:i:1
HWI-ST1145:74:C101DACXX:7:1210:11167:8699        0       chr20   271218  50      50M4700N50M     *       0
       0       GTGGCTCTTCCACAGGAATGTTGAGGATGACATCCATGTCTGGGGTGCACTTGGGTCTCCGAAGCAGAACATCCTCAAATATGACCTCTCG
accepted_hits.sam
```

Fig. A.8 Illustration of a VCF file, figure drawn from https://insidedna.me/tutorial

A.9 Illustration of a VCF file

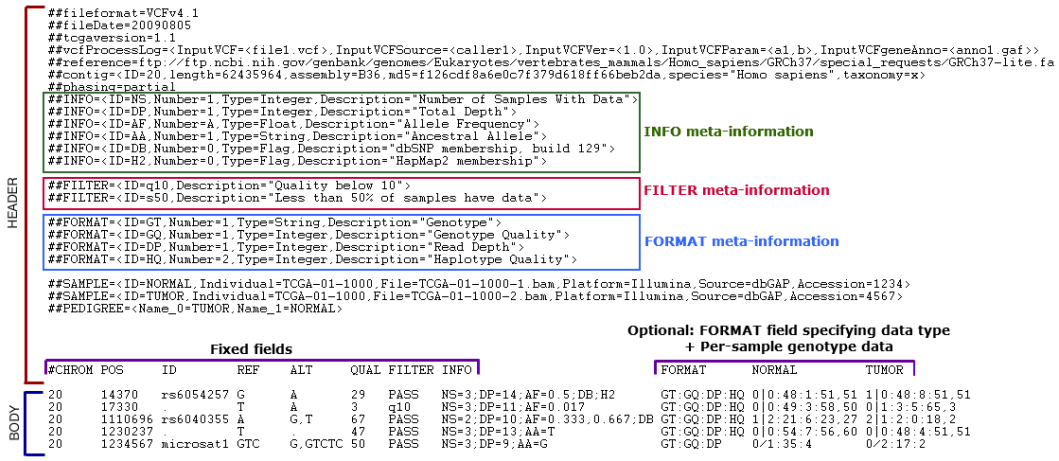


Fig. A.9 Illustration of a VCF file, figure drawn from https://wiki.nci.nih.gov/

A.10 Partial list of tools and resources to annotate DNA sequence variants

Tool/resource	Description	URL	Reference
CADD	A framework that integrates multiple annotations into one metric by contrasting variants that survived natural selection with simulated mutations	http://cadd.gs.washington.edu/	[88]
ENCODE	Annotation of potential functional elements (for example, histone tail modifications) in several cell lines	https://www.encodeproject.org/	[89]
Epigenomics Roadmap	Annotation of potential functional elements (for example, DNAse I hypersensitive sites) in many human tissues and primary cells	http://www.roadmapepigenomics.org/	[90]
FANTOM5	Annotation of transcriptional enhancers in many human tissues and primary cells through detection of bidirectional capped transcription	http://enhancer.binf.ku.dk/	[91]
GERP	Identifies constrained elements in multiple alignments by quantifying substitution deficits	http://mendel.stanford.edu/SidowLab/downloads/gerp/	[92]
HaploReg	Visualization of DNA polymorphisms along with their predicted chromatin state, their sequence conservation across mammals, and their effect on regulatory motifs	http://www.broadinstitute.org/mammals/haploreg/haploreg.php	[93]
Phen-Gen	Method that combines patients' disease symptoms and sequencing data with prior domain knowledge to identify the causative genes for rare disorders	http://phen-gen.org/about.html	[94]
PolyPhen-2	A tool that predicts the possible impact of an amino acid substitution on the structure and function of a human protein using straightforward physical and comparative considerations	http://genetics.bwh.harvard.edu/pph2/	[95]
RegulomeDB	A database that annotates SNPs with known and predicted regulatory elements in the intergenic regions of the human genome using gene expression, ENCODE and literature-mining data	http://regulomedb.org/	[96]
RVIS	This score is designed to rank genes in terms of whether they have more or less common functional genetic variation relative to the genome-wide expectation given the amount of apparently neutral variation the gene has	http://chgv.org/GenicIntolerance/	[97]
SIFT	Predicts whether an amino acid substitution affects protein function based on the degree of conservation of amino acid residues in sequence alignments derived from closely related sequences	http://sift.jcvi.org/	[98]
VEP	Determines the effect of your variants (SNPs, insertions, deletions, CNVs or structural variants) on genes, transcripts and protein sequence, as well as regulatory regions.	http://useast.ensembl.org/info/docs/tools/vep/index.html?redirect=no	[99]

CNV, copy number variant; SNP, single nucleotide polymorphism.

Fig. A.10 Non exhaustive list of prediction programs drawn from [2]

A.11 List of Software Packages for Rare-Variant Association

Table 3. List of Software Packages for Rare-Variant Association Tests

Name	Type	Methods Implemented	URL
EPACTS	stand alone	burden, MB test, SKAT, SKAT-O, VT	http://genome.sph.umich.edu/wiki/EPACTS
GRANVIL	stand alone	ARIEL, MZ	http://www.well.ox.ac.uk/GRANVIL
MiST	R-package	SKAT, MiST	http://cran.r-project.org/web/packages/MiST
PLINK/SEQ	stand alone	burden, C-alpha test, SKAT, SKAT-O, VT	http://atgu.mgh.harvard.edu/plinkseq
Rvtests	stand alone	burden, VT, KBAC method, SKAT	http://genome.sph.umich.edu/wiki/Rvtests
SCORE-Seq	stand alone	burden, SKAT, EREC test, VT, WSS	http://dlin.web.unc.edu/software/score-seq
SKAT	R-package	burden, SKAT, SKAT-O	http://www.hsph.harvard.edu/skat, http://cran.r-project.org/web/packages/SKAT
VAT	stand alone	aSum, burden, C-alpha test, KBAC method, RBT, VT	http://varianttools.sourceforge.net/Association/HomePage
Software Packages for Meta-analysis			
MASS	stand alone	meta-analysis: burden, SKAT, VT	http://dlin.web.unc.edu/software/mass
MetaSKAT	R-package	meta-analysis: burden, SKAT, SKAT-O	http://www.hsph.harvard.edu/skat, http://cran.r-project.org/web/packages/MetaSKAT
seqMeta	R-package	meta-analysis: burden, SKAT, SKAT-O	http://cran.r-project.org/web/packages/seqMeta/
RAREMETAL	stand alone	meta-analysis: burden, SKAT, VT	http://genome.sph.umich.edu/wiki/RAREMETAL

Abbreviations are as follows: ARIEL, accumulation of rare variants integrated and extended locus-specific; aSum, data-adaptive sum test; EPACTS, efficient and parallelizable association container toolbox; EREC, estimated regression coefficient; GRANVIL, gene- or region-based analysis of variants of intermediate and low frequency; KBAC, kernel-based adaptive cluster; MASS, meta-analysis of sequencing studies; MB, Madsen and Browning; MiST, mixed-effects score test for continuous outcomes; RBT, replication-based test; Rvtests, rare-variant tests; SKAT, sequence kernel association test; VAT, variant association tools; VT, variable threshold; and WSS, weighted-sum statistic.

Fig. A.11 List of software packages for rare variant association drawn from [62]

A.12 Portion of coding for 2wayIX kernel

```
K1_Help= function(x,y){
  # Helper function for 2 way interaction kernel
  p = length(x)
  a = x*y
  b = cumsum(a)
  return(sum(a[-1]*b[-p]))
}

# Computation of 2wayIX kernel
K = 1+Z%*%t(Z)
N1=  matrix(nrow = n, ncol = n)
for (i in 1:n){
  for (j in i:n){
    N1[j,i] = N1[i,j] = K1_Help(Z[i,], Z[j,])
  }
}
K = K+N1

return(K)
```

Fig. A.12 Portion of coding for 2wayIX kernel, the latter is taken from the function Kernel.R within the SKAT package. The code of function Kernel.R can be found at https://github.com/leeshawn/SKAT/blob/master/R/Kernel.R

A.13 Little mistake in SKAT package tutorial

```
> SKATBinary(Geno_matrix, obj, method.bin = "Hybrid", r.corr = 1)$p.value
[1] 0.1494129
> SKATBinary(Geno_matrix, obj, method = "Burden", method.bin = "Hybrid", r.corr = 0)$p.value
[1] 0.1494129
> SKATBinary(Geno_matrix, obj, method = "Burden")$p.value
[1] 0.1494129
> SKATBinary(Geno_matrix, obj, method = "davies")$p.value
[1] 0.8945253
> SKATBinary(Geno_matrix, obj, method.bin = "Hybrid", r.corr = 0)$p.value
[1] 0.8945253
> SKAT(Geno_matrix, obj, method = "davies", r.corr = 1)$p.value
[1] 0.1513422
```

Fig. A.13 Screen shot of an *R* session to prove little mistake in SKAT package tutorial

Here, the combination of each line allows to prove that, keeping all the other input parameter unchanged, the *Burden* value for parameter *method* in the *SKATBinary* function is not equivalent to *method* = *"davies"* and *r.corr=1* neither in the *SKAT* function nor in the *SKATBinary* function. The first line serves as a reference, using the default parameter value *Hybrid* and the parameter value *r.corr* = *1*. The second one shows on one hand that the *Hybrid* is still used while the *r.corr* = *0* is clearly not taken into account. This will mean (and given the name of the value parameter it seems totally logical) that the value parameter *Burden* corresponds actually to *r.corr* = *1*. This is confirmed by line 3. Line 4 could let the reader think that the *davies* value parameter is taken into account, but line 5 shows that it was not the case. Indeed, only the default value parameters were used, namely *Hybrid* and *r.corr* = *0*. Moreover, the last line shows that the actual *davies* and *r.corr* = *1* values parameters choices (that are only implemented in the *SKAT* function) give a different result.

A.14 Illustration of the small-sample adjustment method

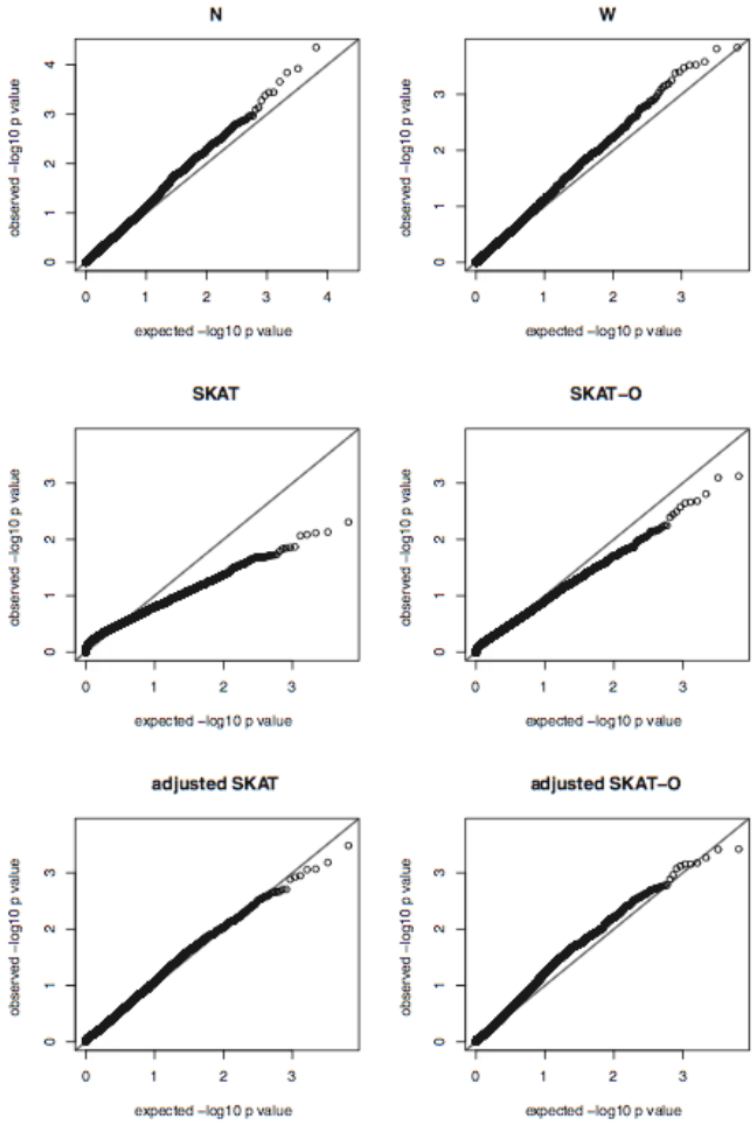


Fig. A.14 $-\log_{10}$ Q-Q plots of observed versus expected p values for the ALI exome sequence data for six methods : burden tests (linear (**N**) and weighted (**W**)), SKAT, SKATO, adjusted SKAT and adjusted SKATO. The x axis represents $-\log_{10}$ expected p values, and the y axis represents $-\log_{10}$ observed p values. A total of 6488 genes with at least four rare variants were tested for associations with ALI severity

A.15 Illustration of raw data retrieval from Highlander

	patient	chr	pos	reference	alternative	zygosity	exac_af	gonl_af	filters	read_depth	genotype_quality	gene_symbol
8	AAD-1-T	1	6529182	TTCC	T	Heterozygous	NA	NA	PASS	19	99	PLEKHG5
9	AAD-1-T	1	7863826	A	G	Heterozygous	NA	NA	PASS	42	99	PER3
11	AAD-1-T	1	8403837	C	T	Heterozygous	6.626576e-05	NA	PASS	12	99	SLC45A1
17	AAD-1-T	1	11771929	G	A	Heterozygous	8.464272e-03	NA	PASS	24	99	DRAXIN
26	AAD-1-T	1	12939546	C	CG	Heterozygous	NA	NA	PASS	209	99	PRAMEF4
32	AAD-1-T	1	16451737	C	G	Heterozygous	2.099012e-03	0.002004	PASS	17	99	EPHA2
35	AAD-1-T	1	17266389	G	A	Heterozygous	1.027479e-03	NA	PASS	12	99	CROCC
55	AAD-1-T	1	28220826	T	C	Heterozygous	6.594457e-04	NA	PASS	41	99	RPA2
58	AAD-1-T	1	32126672	C	T	Homozygous	8.682239e-03	NA	PASS	25	75	COL16A1
60	AAD-1-T	1	32280736	C	G	Heterozygous	8.390106e-06	NA	PASS	32	99	SPOCD1

	allelic_depth_proportion_alt	consensus_prediction	cadd_phred	snpeff_impact	allele_num
8	0.7368421	10	NA	MODERATE	2
9	0.5000000	2	13.46	MODERATE	2
11	0.3333333	6	29.40	MODERATE	2
17	0.3750000	5	28.40	MODERATE	2
26	0.3588517	40	NA	HIGH	2
32	0.3529412	3	13.81	MODERATE	2
35	0.5833333	25	19.37	MODERATE	2
55	0.5853659	2	13.75	MODERATE	2
58	1.0000000	2	12.90	MODERATE	2
60	0.5312500	2	13.62	MODERATE	2

Fig. A.15 Screen shot of the 10 first rows of raw data retrieval for patient for the real data application with consensus prediction greater or equal to 2 and *cadd_phred* greater or equal to 15 and a restricted group of control individuals.

A.16 Illustration of our function *creatGenoMatrix*

```
########################################## Function to creat the Genotype matrix ##########################################
# The function can construct the Genotype matrix given a specific region (could be a gene, interval position or chromosome).
# The Genotype matrix can be constructed for specified variant, or for all variant in the region of interest
#Argument : data must be a data frame : extracted from Highlander
#           position_name must a character : chr, pos or gene_symbol
#           position value
#           variant must be a data fram with chr pos ref and alt
#
creatGenoMatrix <- function(data_patientOfInterest_case, data_patientOfInterest_control, position_name,position_value,variant){
```

Fig. A.16 Screen shot of the header of our function *creatGenoMatrix* responsible for the creation of the genotype matrix.

A.17 Illustration of a possible value for our input parameter *variant*

```
> variant
  chr       pos   ref        alt
1   7   3197867     T          C
2   7  45703971     G          A
3   5 117896623 GTGTA          G
4   5  64070617     A AAATTATGAC
5   7  50611735     T          C
```

Fig. A.17 Screen shot of a possible value for input parameter *variant* of our function *creatGenoMatrix*.

A.18 Detailed description of the input parameter of *creatGenoMatrix* function

- *data_patientOfInterest_case* : a data frame containing the information about the disease of interest and obtained after desired filtering and selection (as suggested in previous section). Each row corresponds to a variant and each columns contains informations (e.g. annotations) regarding the given variant.
- *data_patientOfInterest_control* : a data frame containing the information about the controls and obtained after desired filtering and selection (as suggested in previous section). Each row corresponds to a variant and each columns contains informations (e.g. annotations) regarding the given variant.
- *position_name* : a character that precises the type of region of interest for the analysis. The parameter can take 3 different values : *chr*, *pos* or *gene_symbol*. Therefore the region of interest can be one (or more) chromosome, specific position or gene respectively. Default parameter value is gene.
- *position_value* : a character that precises the region of interest for the analysis (e.g. suppose an investigator wants to analyze variants across genes *RASA1* and *EPHB4*, then one needs to set *position_name* = *gene_symbol* and *position_value* = "RASA1" "EPHB4"). The default parameter value is any position value (e.g. if *position_name* = *gene_symbol*, then all genes will be taken into account for the analysis).

- *variant* (optional) : a data frame (as illustrated in figure (A.17)) where each row is a variant and four columns : *chr*, *pos*, *reference* and *alternative*. If this parameter has some value (meaning the investigator has given specific variants), the analysis will be run only for these variants and not any others.

A.19 A brief explanation on each test's name meaning

Trough the master thesis, the reader will often come across names (especially in the code or in the results presentation) that might seem long and complicated. Therefore, we explain briefly here what the names point to and what they imply. Actually, it is pretty simple. In order to select a test, one needs to choose between one of the two function to perform it, namely *SKAT* or *SKATBinary* and select the desired value for the following input parameters

- The kernel to use (linear, linear weighted, etc), which corresponds to parameter *kernel*
- The method use to compute the p value, which is given by parameter *method*
- The kind of test (Burden, SKAT or SKATO), or in other words the value of parameter *r.corr*

Therefore, we end up with a structured way to give names to the different tests. Let take a few examples :

Function	Kernel	p value	Kind of test	resulting name
SKAT	linear	davies	burden	p_linear_davies_burden
SKAT	weighted IBS	liu mod	SKAT	p_weighted_IBS_liumod_SKAT
SKATBinary	linear	ER	burden	pBin_linear_Burden_ER
SKATBinary	linear weighted	Hybrid	SKATO	pBin_linear_weighted_SKATO_Hybrid

The only thing one needs to pay attention to is that for the tests using *SKATBinary*, the kind of test and the p value method have to be inverted in the name. But as one can see, using this notation greatly simplifies the work and makes it easy to recognize where the p value we are dealing with comes from.

Appendix B

Results of section 5.1 : a global review and characterization of the available tests

This appendix shows the results presented in section 5.1 (A global review and characterization of the available tests). In order to facilitate the reading, note that the structure of this appendix is exactly the same as the one of section 5.1 (e.g. same sections, etc).

B.1 A global review and characterization of the available tests

B.1.1 Testing the kernels

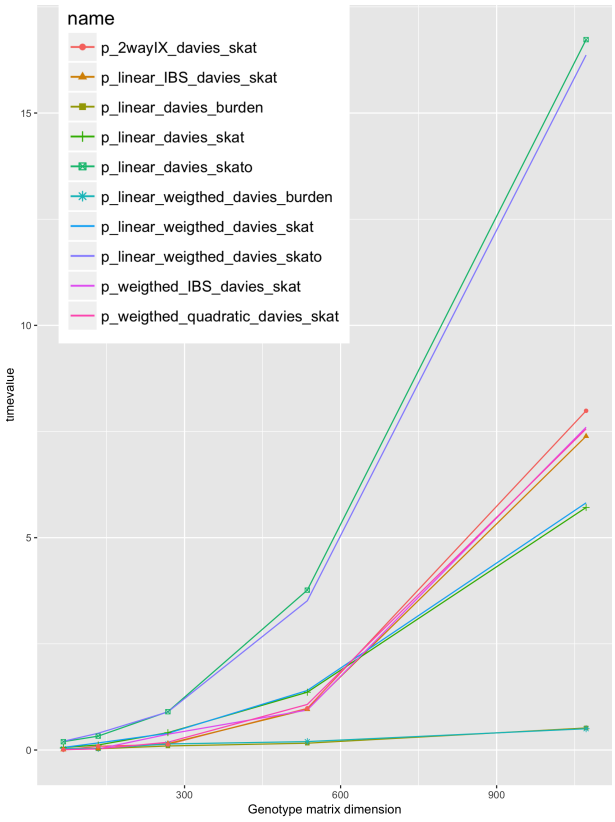


Fig. B.1 Plot of the computational time for different square genotype matrix dimension (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available kernels and the same p value method computation.

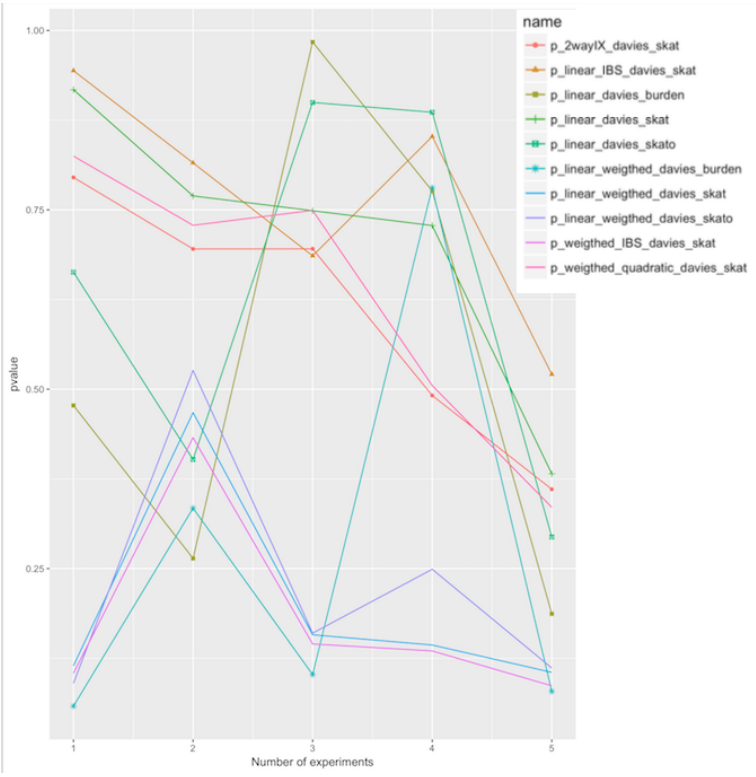


Fig. B.2 Plot of the p values for different phenotype vectors (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available kernels and the same p value method computation.

name	pvalue	pvalue	pvalue	pvalue	pvalue
p_linear_davies_burden	0,4773	0,2641	0,9838	0,7762	0,1869
p_linear_weigthed_davies_burden	0,0582	0,3342	0,1022	0,7807	0,0788
p_linear_davies_skat	0,9174	0,7694	0,7488	0,7281	0,3820
p_linear_weigthed_davies_skat	0,1146	0,4674	0,1577	0,1435	0,1053
p_weigthed_quadratic_davies_skat	0,8248	0,7284	0,7492	0,5051	0,3354
p_linear_IBS_davies_skat	0,9439	0,8154	0,6858	0,8522	0,5208
p_weigthed_IBS_davies_skat	0,1041	0,4324	0,1449	0,1352	0,0866
p_2waylX_davies_skat	0,7952	0,6955	0,6956	0,4912	0,3605
p_linear_davies_skato	0,6631	0,4020	0,8996	0,8860	0,2941
p_linear_weigthed_davies_skato	0,0902	0,5261	0,1600	0,2491	0,1113

Fig. B.3 Output value analysis : p value summary table for each tests and each experiment

name	timevalue	timevalue	timevalue	timevalue	timevalue
p_linear_davies_burden	0,0633	0,0148	0,0129	0,0144	0,0171
p_linear_weigthed_davies_burden	0,0668	0,0154	0,0144	0,0145	0,0143
p_linear_davies_skat	0,0809	0,0267	0,0236	0,0245	0,0259
p_linear_weigthed_davies_skat	0,0351	0,0297	0,0285	0,0262	0,0266
p_weigthed_quadratic_davies_skat	32,7352	30,7007	30,6405	30,3568	31,0788
p_linear_IBS_davies_skat	30,8342	31,0163	30,7666	30,5855	31,7003
p_weigthed_IBS_davies_skat	30,8156	30,8933	30,5991	30,6465	30,5907
p_2waylX_davies_skat	30,8308	30,9291	30,6593	33,0978	32,1891
p_linear_davies_skato	0,6540	0,6957	0,6855	0,7199	0,7053
p_linear_weigthed_davies_skato	0,3520	0,3807	0,3288	0,3439	0,3567

Fig. B.4 Output value analysis : time value summary table for each tests and each experiment

name	pvalue	pvalue	pvalue	pvalue	pvalue
p_linear_davies_burden	0,6784	0,5761	0,4082	0,2642	0,1053
p_linear_weigthed_davies_burden	0,2669	0,1077	0,0228	0,0014	3,21E-06
p_linear_davies_skat	0,3964	0,0789	0,0029	5,47E-06	1,61E-12
p_linear_weigthed_davies_skat	0,4638	0,0225	1,82E-05	5,81E-10	9,54E-22
p_weigthed_quadratic_davies_skat	0,3816	0,0795	0,0037	8,68E-06	2,91E-11
p_linear_IBS_davies_skat	0,5374	0,1385	0,0088	4,58E-05	4,43E-10
p_weigthed_IBS_davies_skat	0,4425	0,0459	0,0003	2,27E-07	1,27E-19
p_2waylX_davies_skat	0,4082	0,0865	0,0040	9,64E-06	3,28E-11
p_linear_davies_skato	0,5626	0,1320	0,0057	1,44E-05	1,13E-11
p_linear_weigthed_davies_skato	0,4176	0,0414	4,48E-05	4,05E-09	6,68E-21

Fig. B.5 Computational time analysis : p value summary table for each tests and each experiment

name	timevalue	timevalue	timevalue	timevalue	timevalue
p_linear_davies_burden	0,0231	0,0290	0,0956	0,1605	0,5203
p_linear_weigthed_davies_burden	0,0328	0,0380	0,1426	0,1985	0,5016
p_linear_davies_skat	0,0584	0,1184	0,4089	1,3611	5,7123
p_linear_weigthed_davies_skat	0,0606	0,1669	0,3933	1,4022	5,8218
p_weigthed_quadratic_davies_skat	0,0081	0,0273	0,1829	1,0733	7,5627
p_linear_IBS_davies_skat	0,0093	0,0381	0,1495	0,9583	7,3898
p_weigthed_IBS_davies_skat	0,0120	0,0251	0,3721	0,9418	7,6021
p_2waylX_davies_skat	0,0070	0,0878	0,1320	0,9811	7,9881
p_linear_davies_skato	0,1959	0,3271	0,9025	3,7648	16,7285
p_linear_weigthed_davies_skato	0,2058	0,3952	0,8976	3,5143	16,3649

Fig. B.6 Computational time analysis : time value summary table for each tests and each experiment

B.1.2 Testing the p value computation methods

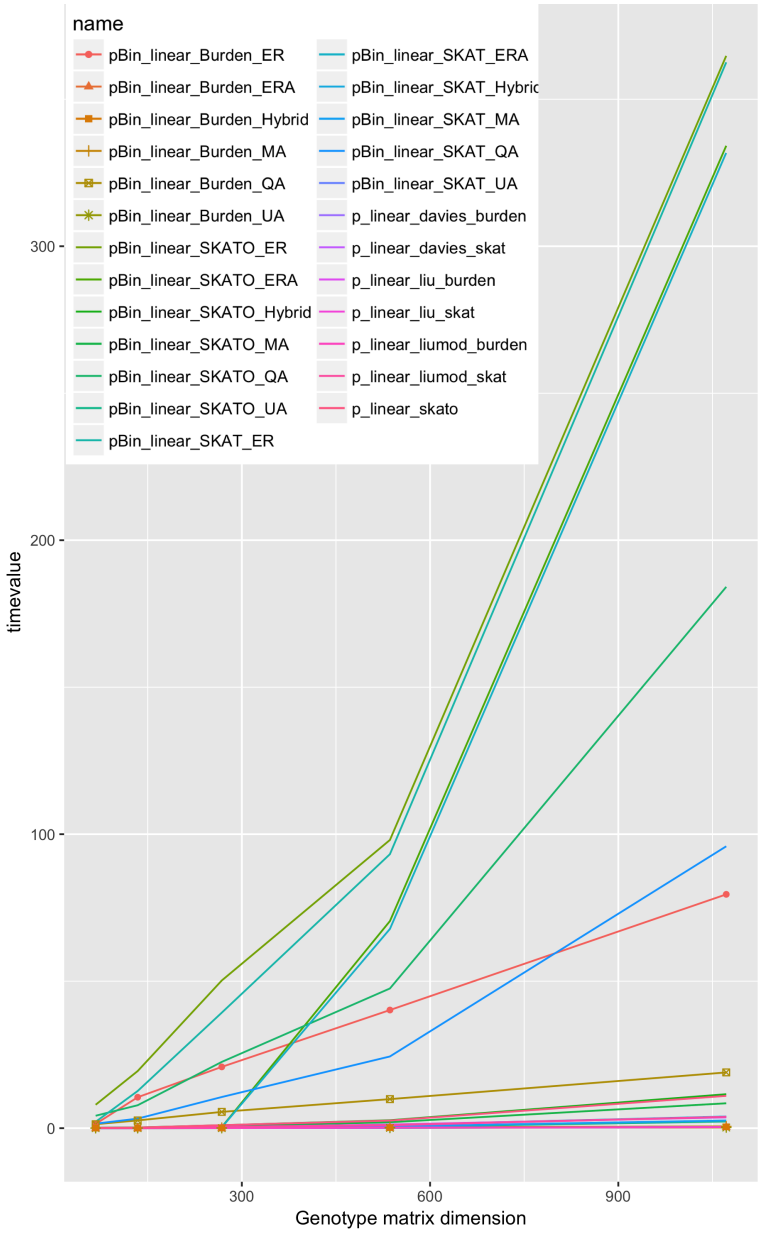


Fig. B.7 Plot of the computational time for different square genotype matrix dimension (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available p value method computation and the same kernel.

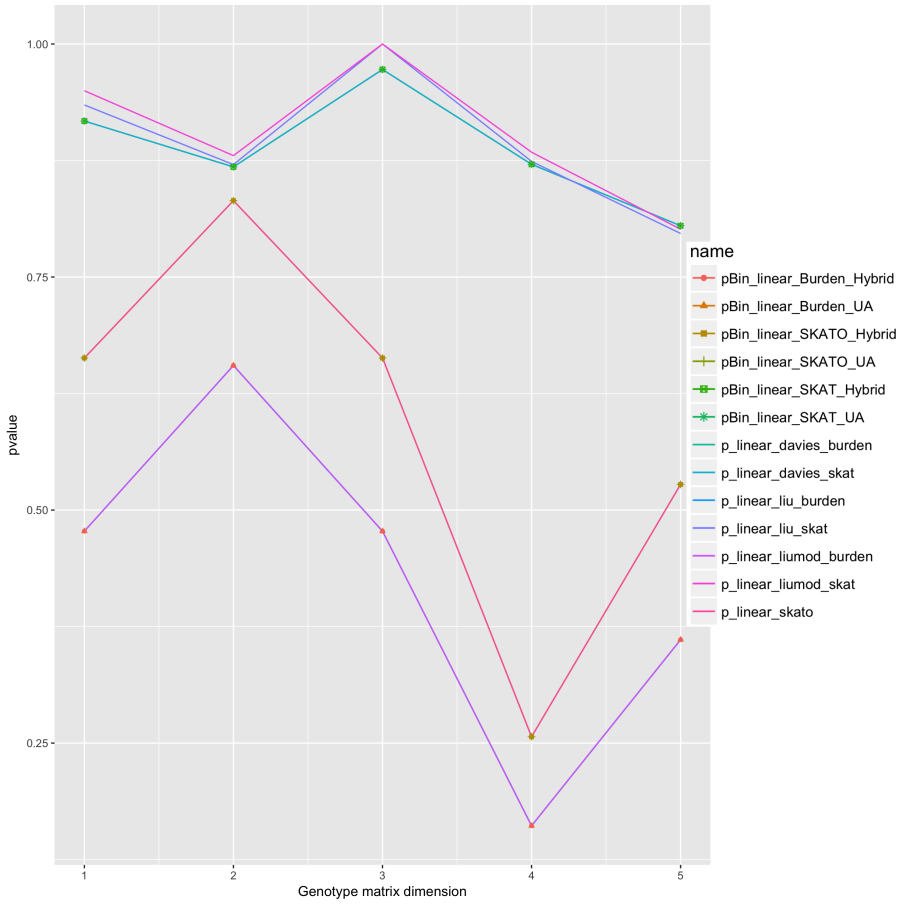


Fig. B.8 Plot of the p values for different phenotype vectors (x axis) for all kinds of tests (burden, SKAT and SKATO) using all the available p value method computation and the same kernel.

name	pvalue	pvalue	pvalue	pvalue	pvalue
p_linear_davies_burden	0,4773	0,6551	0,4773	0,1612	0,3609
p_linear_liu_burden	0,4773	0,6551	0,4773	0,1612	0,3609
p_linear_liumod_burden	0,4773	0,6551	0,4773	0,1612	0,3609
pBin_linear_Burden_UA	0,4773	0,6551	0,4773	0,1612	0,3609
pBin_linear_Burden_Hybrid	0,4773	0,6551	0,4773	0,1612	0,3609
p_linear_davies_skat	0,9174	0,8682	0,9726	0,8709	0,8050
p_linear_liu_skat	0,9346	0,8706	1,0000	0,8740	0,7967
p_linear_liumod_skat	0,9499	0,8802	1,0000	0,8839	0,8013
pBin_linear_SKAT_UA	0,9174	0,8682	0,9726	0,8709	0,8050
pBin_linear_SKAT_Hybrid	0,9174	0,8682	0,9726	0,8709	0,8050
p_linear_skato	0,6631	0,8320	0,6631	0,2568	0,5275
pBin_linear_SKATO_UA	0,6631	0,8320	0,6631	0,2568	0,5275
pBin_linear_SKATO_Hybrid	0,6631	0,8320	0,6631	0,2568	0,5275

Fig. B.9 Output value analysis : p value summary table for each test and each experiment

name	timevalue	timevalue	timevalue	timevalue	timevalue
p_linear_davies_burden	0,0173	0,0166	0,0159	0,0131	0,0128
p_linear_liu_burden	0,0153	0,0128	0,0151	0,0159	0,0178
p_linear_liumod_burden	0,0257	0,0134	0,0120	0,0137	0,0120
pBin_linear_Burden_UA	0,0294	0,0318	0,0284	0,0286	0,0290
pBin_linear_Burden_Hybrid	0,0281	0,0298	0,0266	0,0289	0,0299
p_linear_davies_skat	0,0269	0,0266	0,0818	0,0346	0,0278
p_linear_liu_skat	0,0270	0,0229	0,0229	0,0295	0,0313
p_linear_liumod_skat	0,0331	0,0248	0,0287	0,0269	0,0284
pBin_linear_SKAT_UA	0,0383	0,0433	0,0444	0,0473	0,0448
pBin_linear_SKAT_Hybrid	0,0464	0,0445	0,0422	0,0442	0,0469
p_linear_skato	0,7366	0,7347	0,7163	0,6823	0,7475
pBin_linear_SKATO_UA	0,7515	0,7361	0,7433	0,6992	0,7741
pBin_linear_SKATO_Hybrid	0,7250	0,7426	0,7523	0,7062	0,7788

Fig. B.10 Output value analysis : time value summary table for each test and each experiment

name	pvalue	pvalue	pvalue	pvalue	pvalue
p_linear_davies_burden	0,678397589	0,571424311	0,425580158	0,248019348	0,107613983
p_linear_liu_burden	0,678397589	0,571424311	0,425580158	0,248019348	0,107613983
p_linear_liumod_burden	0,678397589	0,571424311	0,425580158	0,248019348	0,107613983
pBin_linear_Burden_ER	0,726802621	0,566627717	0,429744285	0,245335377	0,106062447
pBin_linear_Burden_QA	1	0,567794727	0,418400022	0,251636596	0,103709812
pBin_linear_Burden_MA	1	0,673709744	0,428379852	0,231667045	0,102665155
pBin_linear_Burden_UA	0,684105154	0,565022271	0,41579654	0,24981631	0,103637138
pBin_linear_Burden_ERA	0,730497503	0,575078248	0,416920944	0,249541016	0,105373256
pBin_linear_Burden_Hybrid	0,678397589	0,571424311	0,425580158	0,248019348	0,107613983
p_linear_davies_skat	0,396405549	0,078753292	0,002858137	3,78E-06	8,23E-12
p_linear_liu_skat	0,396405549	0,078753292	0,002858137	3,78E-06	8,23E-12
p_linear_liumod_skat	0,396405549	0,078753292	0,002858137	3,78E-06	8,23E-12
pBin_linear_SKAT_ER	0,421964246	0,08037996	0,003117498	7,50E-06	5,00E-07
pBin_linear_SKAT_QA	0,38892022	0,078926121	0,003092884	4,01E-06	1,18E-11
pBin_linear_SKAT_MA	0,397335604	0,078861078	0,003218337	2,65E-06	4,58E-11
pBin_linear_SKAT_UA	0,383190692	0,080116623	0,003452864	8,26E-06	1,24E-11
pBin_linear_SKAT_ERA	0,42641845	0,076094336	0,003198063	5,52E-06	0
pBin_linear_SKAT_Hybrid	0,396405549	0,078753292	0,002858137	3,78E-06	8,23E-12
p_linear_skato	0,562554607	0,132101216	0,00559092	9,93E-06	5,76E-11
pBin_linear_SKATO_ER	0,611634743	0,085157957	0,003647998	9,00E-06	5,00E-07
pBin_linear_SKATO_QA	0,554942677	0,132055858	0,00599829	1,07E-05	8,26E-11
pBin_linear_SKATO_MA	0,564972637	0,131731652	0,006203671	7,21E-06	3,21E-10
pBin_linear_SKATO_UA	0,556835121	0,132692989	0,006213008	1,72E-05	8,65E-11
pBin_linear_SKATO_ERA	0,621319497	0,13898299	0,005883346	5,85E-06	0
pBin_linear_SKATO_Hybrid	0,562554607	0,132101216	0,00559092	9,93E-06	5,76E-11

Fig. B.11 Computational time analysis : p value summary table for each test and each experiment

name	timevalue	timevalue	timevalue	timevalue	timevalue
p_linear_davies_burden	0,0175	0,0310	0,0572	0,1619	0,3383
p_linear_liu_burden	0,0205	0,0279	0,0602	0,1703	0,4042
p_linear_liumod_burden	0,0254	0,0262	0,0571	0,2017	0,3333
pBin_linear_Burden_ER	1,4549	10,5335	20,8494	40,1902	79,5442
pBin_linear_Burden_QA	1,3499	2,6390	5,5422	9,8881	18,9753
pBin_linear_Burden_MA	0,0355	0,0655	0,1558	0,2763	0,6300
pBin_linear_Burden_UA	0,0123	0,0126	0,0253	0,1054	0,2211
pBin_linear_Burden_ERA	0,0189	0,0251	0,0494	0,1197	0,3478
pBin_linear_Burden_Hybrid	0,0351	0,0277	0,0687	0,2009	0,4501
p_linear_davies_skat	0,0539	0,0842	0,3480	1,2656	3,6808
p_linear_liu_skat	0,0590	0,1147	0,4551	0,9772	3,8597
p_linear_liumod_skat	0,0812	0,0801	0,3754	1,1055	3,8575
pBin_linear_SKAT_ER	2,1659	12,6531	39,2946	93,2094	362,5618
pBin_linear_SKAT_QA	1,5413	3,2505	10,6011	24,4016	95,8878
pBin_linear_SKAT_MA	0,0385	0,0810	0,2748	0,6054	2,6381
pBin_linear_SKAT_UA	0,0059	0,0115	0,0458	0,0900	0,3875
pBin_linear_SKAT_ERA	0,0158	0,0372	0,2736	67,7731	331,7183
pBin_linear_SKAT_Hybrid	0,0463	0,0871	0,3901	1,0943	3,9953
p_linear_skato	0,1378	0,2156	1,0341	2,5858	11,0127
pBin_linear_SKATO_ER	7,9687	19,4355	50,2363	98,0387	364,7519
pBin_linear_SKATO_QA	4,2311	7,8074	22,5466	47,5270	184,1369
pBin_linear_SKATO_MA	0,1181	0,1965	0,6539	2,0154	8,4893
pBin_linear_SKATO_UA	0,1747	0,2038	0,2124	0,3904	2,2746
pBin_linear_SKATO_ERA	0,0819	0,1339	0,3912	70,4376	334,1774
pBin_linear_SKATO_Hybrid	0,1065	0,2467	1,0009	2,7054	11,5689

Fig. B.12 Computational time analysis : time value summary table for each test and each experiment

B.1.3 A global tests characterization

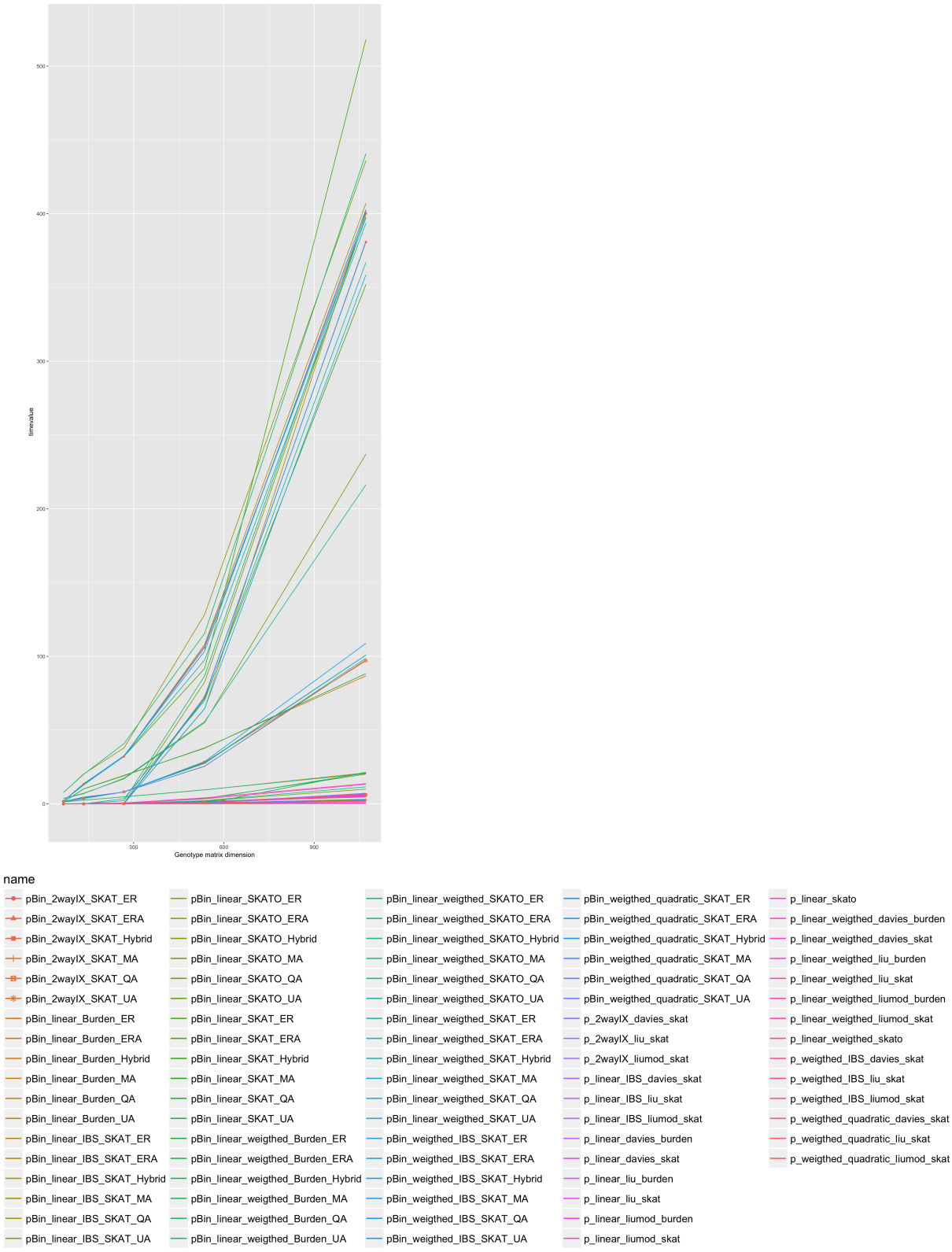
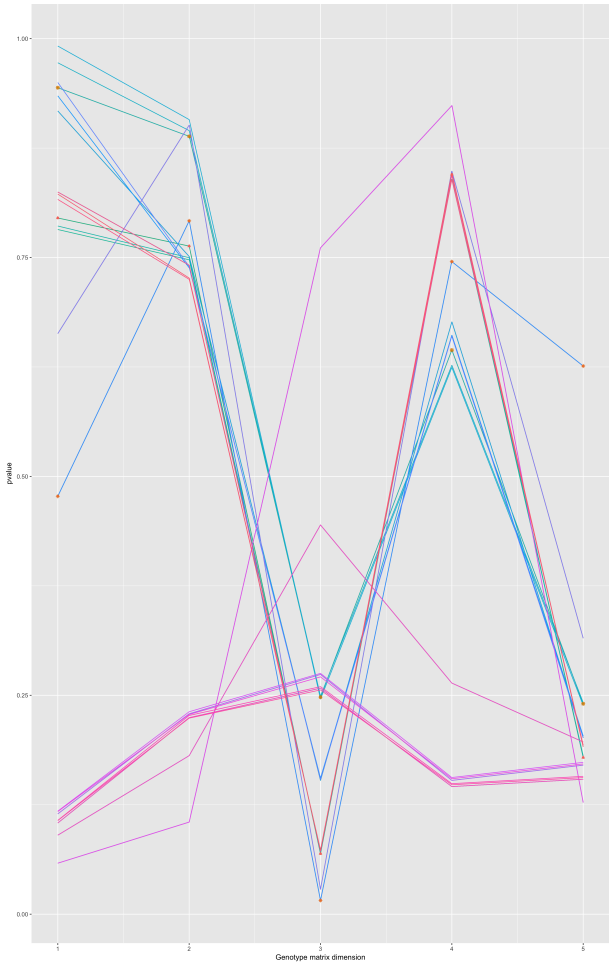


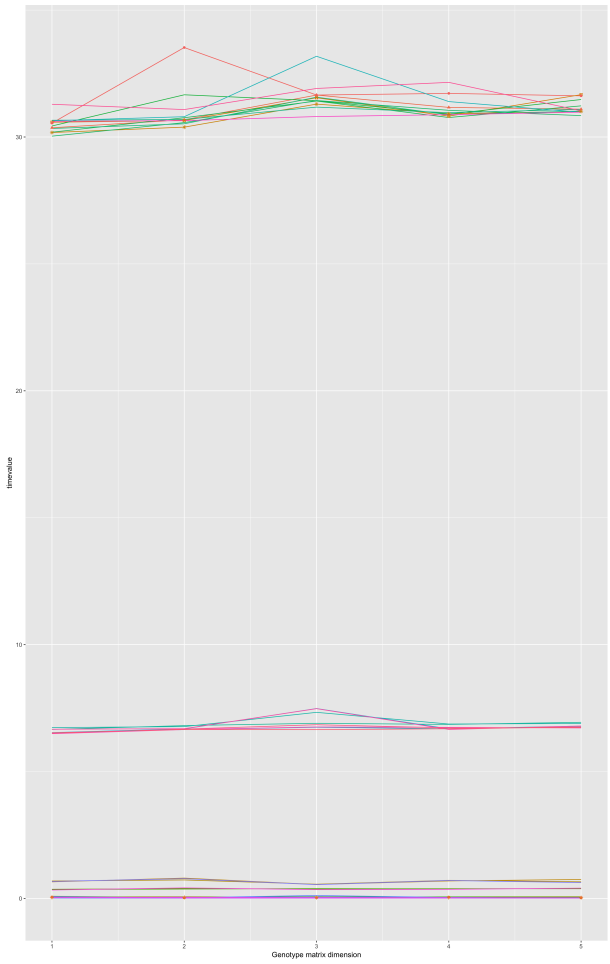
Fig. B.13 Plot of computational time (with y axis going from 0 to 500 seconds) for different square genotype matrix dimensions (x axis) for all kinds of tests (burden, SKAT and SKATO) implementing all the available combinations of kernels and p value method computations in order to highlight the huge diversity of tests.



name

- pBin_2wayIX_SKAT_Hybrid
- pBin_2wayIX_SKAT_UA
- pBin_linear_Burden_Hybrid
- pBin_linear_Burden_UA
- pBin_linear_IBS_SKAT_Hybrid
- pBin_linear_IBS_SKAT_UA
- pBin_linear_SKATO_Hybrid
- pBin_linear_SKATO_UA
- pBin_linear_SKAT_Hybrid
- pBin_linear_SKAT_UA
- pBin_linear_weigthed_Burden_Hybrid
- pBin_linear_weigthed_Burden_UA
- pBin_linear_weigthed_SKATO_Hybrid
- pBin_linear_weigthed_SKATO_UA
- pBin_linear_weigthed_SKAT_Hybrid
- pBin_linear_weigthed_SKAT_UA
- pBin_weigthed_IBS_SKAT_Hybrid
- pBin_weigthed_IBS_SKAT_UA
- pBin_weigthed_quadratic_SKAT_Hybrid
- pBin_weigthed_quadratic_SKAT_UA
- p_2wayIX_davies_skat
- p_2wayIX_liu_skat
- p_2wayIX_liumod_skat
- p_linear_IBS_davies_skat
- p_linear_IBS_liu_skat
- p_linear_IBS_liumod_skat
- p_linear_davies_burden
- p_linear_davies_skat
- p_linear_liu_burden
- p_linear_liu_skat
- p_linear_liumod_burden
- p_linear_liumod_skat
- p_linear_skato
- p_linear_weigthed_davies_burden
- p_linear_weigthed_davies_skat
- p_linear_weigthed_liu_burden
- p_linear_weigthed_liu_skat
- p_linear_weigthed_liumod_burden
- p_linear_weigthed_liumod_skat
- p_linear_weigthed_skato
- p_weigthed_IBS_davies_skat
- p_weigthed_IBS_liu_skat
- p_weigthed_IBS_liumod_skat
- p_weigthed_quadratic_davies_skat
- p_weigthed_quadratic_liu_skat
- p_weigthed_quadratic_liumod_skat

Fig. B.14 Plot of the p values (on the left), and computational time (on the right, with y axis going from 0 to 30 seconds) for different case-controls (x axis) for all kinds of tests (burden, SKAT and SKATO) implementing all the functional combinations of kernels and p value method computations in order to highlight the huge diversity of tests.



name

- pBin_2wayIX_SKAT_Hybrid
- pBin_2wayIX_SKAT_UA
- pBin_linear_Burden_Hybrid
- pBin_linear_Burden_UA
- pBin_linear_IBS_SKAT_Hybrid
- pBin_linear_IBS_SKAT_UA
- pBin_linear_SKATO_Hybrid
- pBin_linear_SKATO_UA
- pBin_linear_SKAT_Hybrid
- pBin_linear_SKAT_UA
- pBin_linear_weigthed_Burden_Hybrid
- pBin_linear_weigthed_Burden_UA
- pBin_linear_weigthed_SKATO_Hybrid
- pBin_linear_weigthed_SKATO_UA
- pBin_linear_weigthed_SKAT_Hybrid
- pBin_linear_weigthed_SKAT_UA
- pBin_weigthed_IBS_SKAT_Hybrid
- pBin_weigthed_IBS_SKAT_UA
- pBin_weigthed_quadratic_SKAT_Hybrid
- pBin_weigthed_quadratic_SKAT_UA
- p_2wayIX_davies_skat
- p_2wayIX_liu_skat
- p_2wayIX_liumod_skat
- p_linear_IBS_davies_skat
- p_linear_IBS_liu_skat
- p_linear_IBS_liumod_skat
- p_linear_davies_burden
- p_linear_davies_skat
- p_linear_liu_burden
- p_linear_liu_skat
- p_linear_liumod_burden
- p_linear_liumod_skat
- p_linear_skato
- p_linear_weigthed_davies_burden
- p_linear_weigthed_davies_skat
- p_linear_weigthed_liu_burden
- p_linear_weigthed_liu_skat
- p_linear_weigthed_liumod_burden
- p_linear_weigthed_liumod_skat
- p_linear_weigthed_skato
- p_weigthed_IBS_davies_skat
- p_weigthed_IBS_liu_skat
- p_weigthed_IBS_liumod_skat
- p_weigthed_quadratic_davies_skat
- p_weigthed_quadratic_liu_skat
- p_weigthed_quadratic_liumod_skat

Fig. B.15 Plot of the p values (on the left), and computational time (on the right, with y axis going from 0 to 30 seconds) for different case-control (x axis) for all kind of tests (burden, SKAT and SKATO) implementing all the functional combinations of kernels and p value method computations in order to highlight the huge diversity of tests.

name	pvalue	pvalue	pvalue	pvalue	pvalue
p_linear_davies_burden	0,4773	0,7918	0,0157	0,7453	0,6261
p_linear_weigthed_davies_burden	0,0582	0,1052	0,7611	0,9235	0,1275
p_linear_liu_burden	0,4773	0,7918	0,0157	0,7453	0,6261
p_linear_weigthed_liu_burden	0,0582	0,1052	0,7611	0,9235	0,1275
p_linear_liumod_burden	0,4773	0,7918	0,0157	0,7453	0,6261
p_linear_weigthed_liumod_burden	0,0582	0,1052	0,7611	0,9235	0,1275
pBin_linear_Burden_UA	0,4773	0,7918	0,0157	0,7453	0,6261
pBin_linear_weigthed_Burden_UA	0,0582	0,1052	0,7611	0,9235	0,1275
pBin_linear_Burden_Hybrid	0,4773	0,7918	0,0157	0,7453	0,6261
pBin_linear_weigthed_Burden_Hybrid	0,0582	0,1052	0,7611	0,9235	0,1275
p_linear_davies_skat	0,9174	0,7512	0,1528	0,6764	0,2017
p_linear_weigthed_davies_skat	0,1146	0,2288	0,2738	0,1528	0,1702
p_weigthed_quadratic_davies_skat	0,8248	0,7413	0,0718	0,8437	0,1913
p_linear_IBS_davies_skat	0,9439	0,8884	0,2479	0,6442	0,2403
p_weigthed_IBS_davies_skat	0,1041	0,2244	0,2581	0,1457	0,1541
p_2wayIX_davies_skat	0,7952	0,7631	0,0691	0,8448	0,1789
p_linear_liu_skat	0,9346	0,7382	0,1554	0,6612	0,2035
p_linear_weigthed_liu_skat	0,1177	0,2312	0,2751	0,1561	0,1734
p_weigthed_quadratic_liu_skat	0,8164	0,7251	0,0734	0,8388	0,1931
p_linear_IBS_liu_skat	0,9724	0,8950	0,2491	0,6269	0,2417
p_weigthed_IBS_liu_skat	0,1072	0,2270	0,2599	0,1491	0,1574
p_2wayIX_liu_skat	0,7820	0,7474	0,0706	0,8393	0,1808
p_linear_liumod_skat	0,9499	0,7398	0,1540	0,6599	0,2012
p_linear_weigthed_liumod_skat	0,1172	0,2281	0,2711	0,1547	0,1716
p_weigthed_quadratic_liumod_skat	0,8224	0,7265	0,0733	0,8463	0,1913
p_linear_IBS_liumod_skat	0,9915	0,9075	0,2457	0,6242	0,2385
p_weigthed_IBS_liumod_skat	0,1069	0,2239	0,2561	0,1478	0,1560
p_2wayIX_liumod_skat	0,7860	0,7498	0,0705	0,8467	0,1792
pBin_linear_SKAT_UA	0,9174	0,7512	0,1528	0,6764	0,2017
pBin_linear_weigthed_SKAT_UA	0,1146	0,2288	0,2738	0,1528	0,1702
pBin_weigthed_quadratic_SKAT_UA	0,8248	0,7413	0,0718	0,8437	0,1913
pBin_linear_IBS_SKAT_UA	0,9439	0,8884	0,2479	0,6442	0,2403
pBin_weigthed_IBS_SKAT_UA	0,1041	0,2244	0,2581	0,1457	0,1541
pBin_2wayIX_SKAT_UA	0,7952	0,7631	0,0691	0,8448	0,1789
pBin_linear_SKAT_Hybrid	0,9174	0,7512	0,1528	0,6764	0,2017
pBin_linear_weigthed_SKAT_Hybrid	0,1146	0,2288	0,2738	0,1528	0,1702
pBin_weigthed_quadratic_SKAT_Hybrid	0,8248	0,7413	0,0718	0,8437	0,1913
pBin_linear_IBS_SKAT_Hybrid	0,9439	0,8884	0,2479	0,6442	0,2403
pBin_weigthed_IBS_SKAT_Hybrid	0,1041	0,2244	0,2581	0,1457	0,1541
pBin_2wayIX_SKAT_Hybrid	0,7952	0,7631	0,0691	0,8448	0,1789
p_linear_skato	0,6631	0,9012	0,0285	0,8486	0,3152
p_linear_weigthed_skato	0,0902	0,1810	0,4446	0,2639	0,1967
pBin_linear_SKATO_UA	0,6631	0,9012	0,0285	0,8486	0,3152
pBin_linear_weigthed_SKATO_UA	0,0902	0,1810	0,4446	0,2639	0,1967
pBin_linear_SKATO_Hybrid	0,6631	0,9012	0,0285	0,8486	0,3152
pBin_linear_weigthed_SKATO_Hybrid	0,0902	0,1810	0,4446	0,2639	0,1967

Recommended tests
Functional and fast in category of value
Functional but slow in catgory of value

Fig. B.16 Output value analysis : p value summary table for each test and each experiment

name	timevalue	timevalue	timevalue	timevalue	timevalue
p_linear_davies_burden	0,0168	0,0119	0,0123	0,0458	0,0149
p_linear_weigthed_davies_burden	0,0258	0,0171	0,0151	0,0175	0,0296
p_linear_liu_burden	0,0172	0,0135	0,0129	0,0138	0,0121
p_linear_weigthed_liu_burden	0,0280	0,0163	0,0149	0,0130	0,0166
p_linear_liumod_burden	0,0298	0,0135	0,0133	0,0142	0,0143
p_linear_weigthed_liumod_burden	0,0305	0,0193	0,0155	0,0185	0,0144
pBin_linear_Burden_UA	0,0434	0,0404	0,0294	0,0330	0,0297
pBin_linear_weigthed_Burden_UA	0,0457	0,0280	0,0339	0,0356	0,0389
pBin_linear_Burden_Hybrid	0,0486	0,0288	0,0352	0,0475	0,0287
pBin_linear_weigthed_Burden_Hybrid	0,0886	0,0344	0,0305	0,0331	0,0321
p_linear_davies_skat	0,0739	0,0261	0,0293	0,0292	0,0265
p_linear_weigthed_davies_skat	0,0596	0,0373	0,0546	0,0281	0,0290
p_weigthed_quadratic_davies_skat	31,2904	31,0809	31,9128	32,1529	31,0323
p_linear_IBS_davies_skat	30,6406	30,7997	33,1810	31,3955	30,9705
p_weigthed_IBS_davies_skat	30,6198	30,6437	30,8058	30,8799	30,9783
p_2wayIX_davies_skat	30,6599	30,6914	31,1777	30,9585	31,0892
p_linear_liu_skat	0,0283	0,0258	0,0249	0,0329	0,0275
p_linear_weigthed_liu_skat	0,0262	0,0517	0,0252	0,0360	0,0273
p_weigthed_quadratic_liu_skat	6,5094	6,6722	6,8453	6,7201	6,7289
p_linear_IBS_liu_skat	6,5316	6,6534	6,7457	6,6860	6,7925
p_weigthed_IBS_liu_skat	6,4914	6,6542	6,7576	6,7411	6,7185
p_2wayIX_liu_skat	6,6609	6,8107	6,9011	6,8565	6,9412
p_linear_liumod_skat	0,0257	0,0274	0,1196	0,0396	0,0264
p_linear_weigthed_liumod_skat	0,0280	0,0262	0,0260	0,0249	0,0277
p_weigthed_quadratic_liumod_skat	6,5263	6,6579	6,6500	6,6857	6,7532
p_linear_IBS_liumod_skat	6,5340	6,6821	7,4838	6,6820	6,7867
p_weigthed_IBS_liumod_skat	6,6675	6,6903	7,4783	6,6584	6,7967
p_2wayIX_liumod_skat	6,7323	6,7706	7,3328	6,8721	6,8962
pBin_linear_SKAT_UA	0,0579	0,0692	0,0393	0,0627	0,0681
pBin_linear_weigthed_SKAT_UA	0,0410	0,0447	0,0730	0,0408	0,0443
pBin_weigthed_quadratic_SKAT_UA	30,3358	30,5201	31,5693	30,8737	31,0245
pBin_linear_IBS_SKAT_UA	30,1764	30,3888	31,2964	30,9166	31,0094
pBin_weigthed_IBS_SKAT_UA	30,1890	30,7595	31,4163	30,7677	31,2272
pBin_2wayIX_SKAT_UA	30,3639	30,6791	31,6529	31,1659	31,0962
pBin_linear_SKAT_Hybrid	0,0405	0,0397	0,0391	0,0382	0,0420
pBin_linear_weigthed_SKAT_Hybrid	0,0406	0,0406	0,0612	0,0603	0,0471
pBin_weigthed_quadratic_SKAT_Hybric	30,0330	30,5671	31,4388	31,0517	30,8425
pBin_linear_IBS_SKAT_Hybrid	30,5820	30,6485	31,5455	30,8219	31,6684
pBin_weigthed_IBS_SKAT_Hybrid	30,4341	31,6628	31,4313	30,9012	31,4760
pBin_2wayIX_SKAT_Hybrid	30,5469	33,5258	31,6532	31,7190	31,6216
p_linear_skato	0,6710	0,7801	0,5446	0,7087	0,6249
p_linear_weigthed_skato	0,3330	0,4171	0,3491	0,3594	0,4066
pBin_linear_SKATO_UA	0,6880	0,7295	0,5612	0,7142	0,6559
pBin_linear_weigthed_SKATO_UA	0,3520	0,3719	0,3876	0,3769	0,3931
pBin_linear_SKATO_Hybrid	0,6544	0,8026	0,5490	0,6904	0,7482
pBin_linear_weigthed_SKATO_Hybrid	0,3623	0,3899	0,3772	0,3842	0,3836

Recommended tests
Functional and fast in category of value
Functional but slow in catgory of value

Fig. B.17 Output value analysis : time value summary table for each tests and each experiment

name	pvalue	pvalue	pvalue	pvalue	pvalue
p_linear_davies_burden	6,784E-01	5,714E-01	4,256E-01	2,480E-01	1,076E-01
p_linear_weigthed_davies_burden	2,669E-01	1,115E-01	2,153E-02	1,338E-03	3,057E-06
p_linear_liu_burden	6,784E-01	5,714E-01	4,256E-01	2,480E-01	1,076E-01
p_linear_weigthed_liu_burden	2,669E-01	1,115E-01	2,153E-02	1,338E-03	3,057E-06
p_linear_liumod_burden	6,784E-01	5,714E-01	4,256E-01	2,480E-01	1,076E-01
p_linear_weigthed_liumod_burden	2,669E-01	1,115E-01	2,153E-02	1,338E-03	3,057E-06
pBin_linear_Burden_ER	7,268E-01	5,666E-01	4,297E-01	2,453E-01	1,061E-01
pBin_linear_weigthed_Burden_ER	2,779E-01	1,111E-01	2,278E-02	1,212E-03	3,500E-06
pBin_linear_Burden_QA	1,000E+00	5,678E-01	4,184E-01	2,516E-01	1,037E-01
pBin_linear_weigthed_Burden_QA	2,661E-01	1,077E-01	2,221E-02	1,232E-03	5,592E-06
pBin_linear_Burden_MA	1,000E+00	6,737E-01	4,284E-01	2,317E-01	1,027E-01
pBin_linear_weigthed_Burden_MA	2,784E-01	1,080E-01	2,152E-02	1,210E-03	5,006E-06
pBin_linear_Burden_UA	6,841E-01	5,650E-01	4,158E-01	2,498E-01	1,036E-01
pBin_linear_weigthed_Burden_UA	2,555E-01	1,078E-01	2,296E-02	1,300E-03	5,410E-06
pBin_linear_Burden_ERA	7,305E-01	5,751E-01	4,169E-01	2,495E-01	1,054E-01
pBin_linear_weigthed_Burden_ERA	2,809E-01	1,158E-01	2,342E-02	1,293E-03	7,059E-06
pBin_linear_Burden_Hybrid	6,784E-01	5,714E-01	4,256E-01	2,480E-01	1,076E-01
pBin_linear_weigthed_Burden_Hybrid	2,669E-01	1,115E-01	2,153E-02	1,338E-03	3,057E-06
p_linear_davies_skat	3,964E-01	7,875E-02	2,858E-03	3,784E-06	8,226E-12
p_linear_weigthed_davies_skat	4,638E-01	2,451E-02	2,426E-05	3,879E-10	1,892E-22
p_weigthed_quadratic_davies_skat	3,816E-01	7,945E-02	3,656E-03	8,681E-06	2,907E-11
p_linear_IBS_davies_skat	5,374E-01	1,385E-01	8,757E-03	4,576E-05	4,428E-10
p_weigthed_IBS_davies_skat	4,425E-01	4,589E-02	2,947E-04	2,267E-07	1,268E-19
p_2waylX_davies_skat	4,082E-01	8,646E-02	4,030E-03	9,635E-06	3,280E-11
p_linear_liu_skat	3,964E-01	7,875E-02	2,858E-03	3,784E-06	8,226E-12
p_linear_weigthed_liu_skat	4,638E-01	2,451E-02	2,426E-05	3,879E-10	1,892E-22
p_weigthed_quadratic_liu_skat	3,781E-01	8,113E-02	3,533E-03	6,316E-06	1,880E-11
p_linear_IBS_liu_skat	5,221E-01	1,420E-01	8,686E-03	2,666E-05	2,040E-10
p_weigthed_IBS_liu_skat	4,373E-01	4,751E-02	2,002E-04	1,143E-09	1,144E-20
p_2waylX_liu_skat	4,039E-01	8,814E-02	3,911E-03	7,101E-06	2,136E-11
p_linear_liumod_skat	3,964E-01	7,875E-02	2,858E-03	3,784E-06	8,226E-12
p_linear_weigthed_liumod_skat	4,638E-01	2,451E-02	2,426E-05	3,879E-10	1,892E-22
p_weigthed_quadratic_liumod_skat	3,747E-01	8,099E-02	3,689E-03	7,438E-06	2,907E-11
p_linear_IBS_liumod_skat	5,166E-01	1,404E-01	9,189E-03	3,487E-05	4,428E-10
p_weigthed_IBS_liumod_skat	4,315E-01	4,831E-02	2,537E-04	2,767E-09	1,268E-19
p_2waylX_liumod_skat	4,007E-01	8,793E-02	4,071E-03	8,316E-06	3,280E-11
pBin_linear_SKAT_ER	4,220E-01	8,038E-02	3,117E-03	7,500E-06	5,000E-07
pBin_linear_weigthed_SKAT_ER	4,379E-01	2,248E-02	4,850E-05	5,000E-07	5,000E-07
pBin_weigthed_quadratic_SKAT_ER	4,220E-01	8,038E-02	3,117E-03	7,500E-06	5,000E-07
pBin_linear_IBS_SKAT_ER	4,220E-01	8,038E-02	3,117E-03	7,500E-06	5,000E-07
pBin_weigthed_IBS_SKAT_ER	4,220E-01	8,038E-02	3,117E-03	7,500E-06	5,000E-07
pBin_2waylX_SKAT_ER	4,220E-01	8,038E-02	3,117E-03	7,500E-06	5,000E-07

Recommended tests
Functional and fast in category of value
Functional but slow in catgory of value
Not functional everytime but fast
Not functional everytime and slow

Fig. B.18 Computational time analysis : p value summary table for each test and each experiment (part 1)

name	pvalue	pvalue	pvalue	pvalue	pvalue
pBin_linear_SKAT_QA	3,889E-01	7,893E-02	3,093E-03	4,007E-06	1,180E-11
pBin_linear_weigthed_SKAT_QA	4,668E-01	2,249E-02	4,384E-05	1,649E-10	2,497E-21
pBin_weigthed_quadratic_SKAT_QA	3,889E-01	7,893E-02	3,093E-03	4,007E-06	1,180E-11
pBin_linear_IBS_SKAT_QA	3,889E-01	7,893E-02	3,093E-03	4,007E-06	1,180E-11
pBin_weigthed_IBS_SKAT_QA	3,889E-01	7,893E-02	3,093E-03	4,007E-06	1,180E-11
pBin_2wayIX_SKAT_QA	3,889E-01	7,893E-02	3,093E-03	4,007E-06	1,180E-11
pBin_linear_SKAT_MA	3,973E-01	7,886E-02	3,218E-03	2,647E-06	4,580E-11
pBin_linear_weigthed_SKAT_MA	4,620E-01	2,493E-02	4,913E-05	1,218E-10	4,536E-23
pBin_weigthed_quadratic_SKAT_MA	3,973E-01	7,886E-02	3,218E-03	2,647E-06	4,580E-11
pBin_linear_IBS_SKAT_MA	3,973E-01	7,886E-02	3,218E-03	2,647E-06	4,580E-11
pBin_weigthed_IBS_SKAT_MA	3,973E-01	7,886E-02	3,218E-03	2,647E-06	4,580E-11
pBin_2wayIX_SKAT_MA	3,973E-01	7,886E-02	3,218E-03	2,647E-06	4,580E-11
pBin_linear_SKAT_UA	3,832E-01	8,012E-02	3,453E-03	8,255E-06	1,235E-11
pBin_linear_weigthed_SKAT_UA	4,264E-01	4,098E-02	2,239E-04	1,837E-07	2,509E-20
pBin_weigthed_quadratic_SKAT_UA	3,816E-01	7,945E-02	3,656E-03	8,681E-06	2,907E-11
pBin_linear_IBS_SKAT_UA	5,374E-01	1,385E-01	8,757E-03	4,576E-05	4,428E-10
pBin_weigthed_IBS_SKAT_UA	4,425E-01	4,589E-02	2,947E-04	2,267E-07	1,268E-19
pBin_2wayIX_SKAT_UA	4,082E-01	8,646E-02	4,030E-03	9,635E-06	3,280E-11
pBin_linear_SKAT_ERA	4,264E-01	7,609E-02	3,198E-03	5,518E-06	0,000E+00
pBin_linear_weigthed_SKAT_ERA	4,312E-01	2,128E-02	6,196E-05	0,000E+00	0,000E+00
pBin_weigthed_quadratic_SKAT_ERA	4,264E-01	7,609E-02	3,198E-03	5,518E-06	0,000E+00
pBin_linear_IBS_SKAT_ERA	4,264E-01	7,609E-02	3,198E-03	5,518E-06	0,000E+00
pBin_weigthed_IBS_SKAT_ERA	4,264E-01	7,609E-02	3,198E-03	5,518E-06	0,000E+00
pBin_2wayIX_SKAT_ERA	4,264E-01	7,609E-02	3,198E-03	5,518E-06	0,000E+00
pBin_linear_SKAT_Hybrid	3,964E-01	7,875E-02	2,858E-03	3,784E-06	8,226E-12
pBin_linear_weigthed_SKAT_Hybrid	4,638E-01	2,451E-02	2,426E-05	3,879E-10	1,892E-22
pBin_weigthed_quadratic_SKAT_Hybri	3,816E-01	7,945E-02	3,656E-03	8,681E-06	2,907E-11
pBin_linear_IBS_SKAT_Hybrid	5,374E-01	1,385E-01	8,757E-03	4,576E-05	4,428E-10
pBin_weigthed_IBS_SKAT_Hybrid	4,425E-01	4,589E-02	2,947E-04	2,267E-07	1,268E-19
pBin_2wayIX_SKAT_Hybrid	4,082E-01	8,646E-02	4,030E-03	9,635E-06	3,280E-11
p_linear_skato	5,626E-01	1,321E-01	5,591E-03	9,928E-06	5,758E-11
p_linear_weigthed_skato	4,176E-01	4,467E-02	5,937E-05	2,716E-09	1,324E-21
pBin_linear_SKATO_ER	6,116E-01	8,516E-02	3,648E-03	9,000E-06	5,000E-07
pBin_linear_weigthed_SKATO_ER	4,579E-01	2,402E-02	6,150E-05	5,000E-07	5,000E-07
pBin_linear_SKATO_QA	5,549E-01	1,321E-01	5,998E-03	1,067E-05	8,259E-11
pBin_linear_weigthed_SKATO_QA	4,165E-01	4,099E-02	1,071E-04	1,155E-09	1,748E-20
pBin_linear_SKATO_MA	5,650E-01	1,317E-01	6,204E-03	7,211E-06	3,206E-10
pBin_linear_weigthed_SKATO_MA	4,333E-01	4,541E-02	1,138E-04	3,358E-10	3,175E-22
pBin_linear_SKATO_UA	5,568E-01	1,327E-01	6,213E-03	1,719E-05	8,648E-11
pBin_linear_weigthed_SKATO_UA	4,182E-01	7,048E-02	4,493E-04	7,379E-07	1,757E-19
pBin_linear_SKATO_ERA	6,213E-01	1,390E-01	5,883E-03	5,854E-06	0,000E+00
pBin_linear_weigthed_SKATO_ERA	4,497E-01	4,688E-02	1,538E-04	0,000E+00	0,000E+00
pBin_linear_SKATO_Hybrid	5,626E-01	1,321E-01	5,591E-03	9,928E-06	5,758E-11
pBin_linear_weigthed_SKATO_Hybrid	4,176E-01	4,467E-02	5,937E-05	2,716E-09	1,324E-21

Recommended tests
Functional and fast in category of value
Functional but slow in catgory of value
Not functional everytime but fast
Not functional everytime and slow

Fig. B.19 Computational time analysis : p value summary table for each test and each experiment (part 2)

name	timevalue	timevalue	timevalue	timevalue	timevalue
p_linear_davies_burden	0,0161	0,0285	0,0530	0,2243	0,7683
p_linear_weigthed_davies_burden	0,0169	0,0298	0,0578	0,1372	0,6271
p_linear_liu_burden	0,0165	0,0315	0,0526	0,1336	0,6482
p_linear_weigthed_liu_burden	0,0167	0,0285	0,0842	0,1916	0,5667
p_linear_liumod_burden	0,0159	0,0274	0,0652	0,1303	0,4482
p_linear_weigthed_liumod_burden	0,0168	0,0248	0,1178	0,1401	0,4460
pBin_linear_Burden_ER	1,2644	10,1280	19,1017	38,0249	86,8185
pBin_linear_weigthed_Burden_ER	1,2061	10,2056	19,3368	37,7813	88,3558
pBin_linear_Burden_QA	1,3639	2,5752	4,8123	9,5207	21,1034
pBin_linear_weigthed_Burden_QA	1,4149	2,5678	4,8317	9,5751	20,6071
pBin_linear_Burden_MA	0,0454	0,0621	0,1277	0,2607	0,7076
pBin_linear_weigthed_Burden_MA	0,0398	0,0577	0,1266	0,2987	0,7662
pBin_linear_Burden_UA	0,0066	0,0122	0,0257	0,0677	0,3555
pBin_linear_weigthed_Burden_UA	0,0099	0,0144	0,0251	0,0671	0,2991
pBin_linear_Burden_ERA	0,0149	0,0250	0,0937	0,1122	0,4278
pBin_linear_weigthed_Burden_ERA	0,0148	0,0263	0,0479	0,1591	21,4738
pBin_linear_Burden_Hybrid	0,0231	0,0253	0,0636	0,1969	0,8614
pBin_linear_weigthed_Burden_Hybrid	0,0246	0,0259	0,0649	0,2431	0,8775
p_linear_davies_skat	0,0584	0,0790	0,2591	0,9582	6,1967
p_linear_weigthed_davies_skat	0,0445	0,0812	0,3159	0,9606	6,0109
p_weigthed_quadratic_davies_skat	0,0059	0,0172	0,1011	0,6882	6,5505
p_linear_IBS_davies_skat	0,0071	0,0173	0,1010	0,7594	6,8245
p_weigthed_IBS_davies_skat	0,0079	0,0183	0,1026	0,6963	6,7765
p_2wayIX_davies_skat	0,0064	0,0732	0,1072	0,7310	7,2640
p_linear_liu_skat	0,0401	0,0846	0,2579	0,9626	5,4592
p_linear_weigthed_liu_skat	0,0374	0,0817	0,3070	1,0197	5,0790
p_weigthed_quadratic_liu_skat	0,0068	0,0088	0,0342	0,2020	1,7783
p_linear_IBS_liu_skat	0,0049	0,0093	0,0361	0,2082	1,6961
p_weigthed_IBS_liu_skat	0,0034	0,0090	0,0354	0,2138	1,7327
p_2wayIX_liu_skat	0,0033	0,0097	0,0387	0,2412	2,0070
p_linear_liumod_skat	0,0395	0,0782	0,2609	0,9944	5,1700
p_linear_weigthed_liumod_skat	0,0411	0,0839	0,2907	0,9485	5,8051
p_weigthed_quadratic_liumod_skat	0,0053	0,0096	0,0348	0,2005	2,2032
p_linear_IBS_liumod_skat	0,0072	0,0092	0,0399	0,2102	2,0592
p_weigthed_IBS_liumod_skat	0,0047	0,0092	0,0373	0,2427	1,8311
p_2wayIX_liumod_skat	0,0062	0,0126	0,0391	0,2354	2,0098
pBin_linear_SKAT_ER	1,9731	12,8350	32,2002	91,8215	517,9163
pBin_linear_weigthed_SKAT_ER	1,9168	13,0477	32,1596	97,6182	393,6553
pBin_weigthed_quadratic_SKAT_ER	1,9788	13,4881	32,6830	103,6014	402,8528
pBin_linear_IBS_SKAT_ER	1,9402	13,7708	32,1837	108,0867	407,0277
pBin_weigthed_IBS_SKAT_ER	1,9301	12,9760	32,2320	107,0035	397,0885
pBin_2wayIX_SKAT_ER	1,8616	13,4120	32,1095	105,9680	400,1610

Recommended tests
Functional and fast in category of value
Functional but slow in catgory of value
Not functional everytime but fast
Not functional everytime and slow

Fig. B.20 Computational time analysis : time value summary table for each test and each experiment (part 1)

name	timevalue	timevalue	timevalue	timevalue	timevalue
pBin_linear_SKAT_QA	1,5819	3,5608	8,1603	27,8369	100,9668
pBin_linear_weigthed_SKAT_QA	1,5762	4,2069	8,2205	27,4684	100,8938
pBin_weigthed_quadratic_SKAT_QA	1,6460	3,6012	8,1738	25,4463	98,8437
pBin_linear_IBS_SKAT_QA	1,5822	4,6449	8,0883	25,6420	97,7072
pBin_weigthed_IBS_SKAT_QA	1,5225	4,4509	8,1487	28,7194	108,8556
pBin_2wayIX_SKAT_QA	1,5584	3,8348	8,1261	28,2402	97,0359
pBin_linear_SKAT_MA	0,0413	0,0904	0,2029	0,7957	2,6265
pBin_linear_weigthed_SKAT_MA	0,0394	0,1430	0,2057	0,6992	2,6351
pBin_weigthed_quadratic_SKAT_MA	0,0420	0,1297	0,2045	0,8129	2,6471
pBin_linear_IBS_SKAT_MA	0,0416	0,1011	0,2082	0,7176	2,6466
pBin_weigthed_IBS_SKAT_MA	0,0410	0,0917	0,2086	0,7464	2,6314
pBin_2wayIX_SKAT_MA	0,0470	0,0909	0,2121	0,8489	2,6374
pBin_linear_SKAT_UA	0,0077	0,0140	0,0885	0,1389	0,4002
pBin_linear_weigthed_SKAT_UA	0,0066	0,0132	0,0280	0,1312	0,4010
pBin_weigthed_quadratic_SKAT_UA	0,0096	0,0264	0,1129	0,9452	5,4693
pBin_linear_IBS_SKAT_UA	0,0086	0,0278	0,1156	1,0607	5,6155
pBin_weigthed_IBS_SKAT_UA	0,0094	0,0268	0,1176	0,9193	5,5973
pBin_2wayIX_SKAT_UA	0,0099	0,0306	0,1203	0,8967	5,7822
pBin_linear_SKAT_ERA	0,0171	0,0416	0,1940	71,3451	352,0439
pBin_linear_weigthed_SKAT_ERA	0,0173	0,0401	3,5677	70,1351	367,0143
pBin_weigthed_quadratic_SKAT_ERA	0,0184	0,0428	0,1949	72,2827	381,3133
pBin_linear_IBS_SKAT_ERA	0,0209	0,0412	0,1917	64,5580	402,0263
pBin_weigthed_IBS_SKAT_ERA	0,0170	0,0412	0,1958	64,7220	358,6390
pBin_2wayIX_SKAT_ERA	0,0195	0,0397	0,2221	72,6794	380,8437
pBin_linear_SKAT_Hybrid	0,0383	0,2097	0,2755	1,6576	4,6478
pBin_linear_weigthed_SKAT_Hybrid	0,0356	0,0912	0,2771	1,8715	4,6667
pBin_weigthed_quadratic_SKAT_Hybrid	0,0098	0,0257	0,1132	1,0557	6,8898
pBin_linear_IBS_SKAT_Hybrid	0,0100	0,0261	0,1193	1,1093	6,2716
pBin_weigthed_IBS_SKAT_Hybrid	0,0094	0,0327	0,1178	1,0327	6,1831
pBin_2wayIX_SKAT_Hybrid	0,0101	0,0274	0,1191	1,3769	6,4607
p_linear_skato	0,1283	0,2341	0,7064	4,1494	13,7816
p_linear_weigthed_skato	0,1115	0,2290	0,6851	3,4207	13,2993
pBin_linear_SKATO_ER	7,7137	20,3038	37,9310	128,4142	435,9695
pBin_linear_weigthed_SKATO_ER	7,7911	20,0472	40,8670	115,5829	440,6764
pBin_linear_SKATO_QA	3,7414	7,2423	17,0288	55,0900	237,0908
pBin_linear_weigthed_SKATO_QA	3,5803	6,9799	17,4303	55,7945	216,3640
pBin_linear_SKATO_MA	0,1074	0,2375	0,4795	2,0046	10,0501
pBin_linear_weigthed_SKATO_MA	0,1107	0,1885	0,4796	2,2537	11,6067
pBin_linear_SKATO_UA	0,1736	0,1966	0,1624	0,6288	3,1619
pBin_linear_weigthed_SKATO_UA	0,0989	0,1298	0,1044	0,4728	2,7775
pBin_linear_SKATO_ERA	0,0830	0,1325	0,3835	82,5862	398,2389
pBin_linear_weigthed_SKATO_ERA	0,0834	0,1325	2,2125	86,7931	400,6309
pBin_linear_SKATO_Hybrid	0,1094	0,2230	0,6612	3,7196	20,2171
pBin_linear_weigthed_SKATO_Hybrid	0,1037	0,2895	0,6610	3,3856	20,9061

Recommended tests
Functional and fast in category of value
Functional but slow in catgory of value
Not functional everytime but fast
Not functional everytime and slow

Fig. B.21 Computational time analysis : time value summary table for each test and each experiment (part 1)

B.1.4 Summary and recommendations

	Category 1	Category 2
Burden tests	p_linear_davies_burden	p_linear_weigthed_davies_burden
	p_linear_liu_burden	p_linear_weigthed_liu_burden
	p_linear_liumod_burden	p_linear_weigthed_liumod_burden
	pBin_linear_Burden_UA	pBin_linear_weigthed_Burden_UA
	pBin_linear_Burden_Hybrid	pBin_linear_weigthed_Burden_Hybrid
	Category 3	**Category 4**
Variance-component tests	p_linear_weigthed_davies_skat	p_weigthed_quadratic_davies_skat
	p_weigthed_IBS_davies_skat	p_2wayIX_davies_skat
	p_linear_weigthed_liu_skat	p_weigthed_quadratic_liu_skat
	p_weigthed_IBS_liu_skat	p_2wayIX_liu_skat
	p_linear_weigthed_liumod_skat	p_weigthed_quadratic_liumod_skat
	p_weigthed_IBS_liumod_skat	p_2wayIX_liumod_skat
	pBin_linear_weigthed_SKAT_UA	pBin_weigthed_quadratic_SKAT_UA
	pBin_weigthed_IBS_SKAT_UA	pBin_2wayIX_SKAT_UA
	pBin_linear_weigthed_SKAT_Hybrid	pBin_weigthed_quadratic_SKAT_Hybrid
	pBin_weigthed_IBS_SKAT_Hybrid	pBin_2wayIX_SKAT_Hybrid
	Category 5	**Category 6**
	p_linear_davies_skat	p_linear_IBS_davies_skat
	p_linear_liu_skat	p_linear_IBS_liu_skat
	p_linear_liumod_skat	p_linear_IBS_liumod_skat
	pBin_linear_SKAT_UA	pBin_linear_IBS_SKAT_UA
	pBin_linear_SKAT_Hybrid	pBin_linear_IBS_SKAT_Hybrid
	Category 7	**Category 8**
SKATO	p_linear_skato	p_linear_weigthed_skato
	pBin_linear_SKATO_UA	pBin_linear_weigthed_SKATO_UA
	pBin_linear_SKATO_Hybrid	pBin_linear_weigthed_SKATO_Hybrid

Fig. B.22 Summary of the 8 categories of test across the 3 class of tests. Within each category there is a recommended test (in green), and some functional and computationally fast (in blue), and functional but time consuming tests (in yellow).

Recommended group of tests for the continuous case

1. **Category 1** : p_linear_liumod_burden
2. **Category 2** : p_linear_weighted_liumod_burden
3. **Category 3** : p_linear_weighted_liu_skat
4. **Category 4** : p_weighted_quadratic_liu_skat
5. **Category 5** : p_linear_liumod_skat

6. **Category 6** : p_linear_IBS_liu_skat
7. **Category 7** : p_linear_skato
8. **Category 8** : p_linear_weighted_skato

Recommended group of tests for the discrete case

1. **Category 1** : pBin_linear_Burden_UA
2. **Category 2** : pBin_linear_weighted_Burden_UA
3. **Category 3** : pBin_linear_weighted_SKAT_UA
4. **Category 4** : p_weighted_quadratic_liu_skat
5. **Category 5** : pBin_linear_SKAT_UA
6. **Category 6** : p_linear_IBS_liu_skat
7. **Category 7** : pBin_linear_SKATO_UA
8. **Category 8** : pBin_linear_weighted_SKATO_UA

Group of tests used for our following analysis

1. **Category 1** : p_linear_davies_burden
2. **Category 2** : p_linear_weighted_liu_burden
3. **Category 3** : p_linear_weighted_liumod_skat
4. **Category 4** : p_weighted_quadratic_liumod_skat
5. **Category 5** : p_linear_liu_skat
6. **Category 6** : p_linear_IBS_liu_skat
7. **Category 7** : p_linear_skato
8. **Category 8** : p_linear_weighted_skato

B.1.5 A validation case : the *CM / CLP* association study

To run this analysis, investigators at GEHU laboratory (mainly Mustapha Amyere in charge of this particular research) gave us a precise patient list (namely the CM-AVM, 60 patients) and suggested to use the Cleft lip and or palate (CLP) patient as a control group (we took 100 controls). When testing for the *CM* analysis, we tried different filtering methods in order to focus on different variants and therefore observing different results. Here, we only present the tables that summarize the p values output for each test and each gene. More results about this analysis can be found in the drive. In the first experiment (illustrated in figure *B*.23 and *B*.24), we used a series of filters[1] (recommended by the investigators)

- *read_depth* > 5
- *allelic_depth_proportion_alt* > 0.1
- *allele_num* < 3
- *genotype_quality* > 80
- *filters* $=$ PASS
- *instr(platform, HISEQ)*
- *sample_type* $=$ Blood
- *exac_af* < 0.01 or *exac_af* < 0.05 (depending on the experiment)
- *gonl_af* < 0.05 or *gonl_af* < 0.05 (depending on the experiment)

For experiment 3 (results illustrated in figure *B*.25), we decided to remove all filters in order to have access to the full information within Highlander. And finally (results presented on figure *B*.25) we only used *exac_af* < 0.05 and *gonl_af* < 0.05

[1] For detailed information about the meaning of each filter one can refer to [55, 18]

P value	GNA11	GNAQ	KRIT1	PIK3CA	RASA1	DLL4	EFNB2	EPHB4	HEY2	JAG1	JAG2	NOTCH1	NOTCH4
p_linear_davies_burden	0,02429886	0,00941243	1	1	0,63834559	1	0,53107707	0,80208971	0,84532994	1	0,91534246	0,07876118	0,25316923
p_linear_weigthed_liu_burden	0,03159881	0,29190392	1	0,76008309	0,84142372	1	0,7724067	0,1911092	0,26476777	0,59962383	1	0,16337471	0,2407303
p_linear_weigthed_liumod_skat	0,03280891	0,15353727	0,24492928	0,308032	0,45072243	0,59154343	0,42708015	0,002903	0,15335625	0,21734497	0,40506807	0,45327884	0,37763519
p_weigthed_quadratic_liumod_skat	0,03984383	0,00073144	0,29711691	0,46749962	0,72215603	0,64707056	0,32653612	0,66052102	0,16049165	0,54434795	0,5735151	0,34795489	0,52514499
p_linear_liu_skat	0,03397649	0,00119483	0,25059983	0,45644003	0,64871881	0,59116641	0,29822493	0,47591815	0,12726851	0,47533286	0,57331651	0,15959017	0,56829496
p_linear_IBS_liu_skat	0,01416166	0,00076981	0,28662124	0,44389767	0,67993148	0,62371441	0,32540554	0,37057938	0,09633605	0,43732912	0,55889402	0,15719977	0,42700562
p_linear_weigthed_skato	0,04474119	0,24596089	0,40133528	0,48742218	0,63817471	0,80276313	0,61961759	0,0063813	0,25260349	0,36878295	0,61247607	0,27397871	0,39220175
p_linear_skato	0,03511423	0,0022106	0,40069774	0,6553752	0,81844756	0,79072756	0,38447318	0,60563437	0,19955776	0,69245308	0,78702193	0,13996196	0,41490014

Fig. B.23 Table of the p values obtained for each test and each gene. Here the filters are activated along with $exac\_af < 0.05$ criteria and $gonl\_af < 0.05$.

P value	GNA11	GNAQ	KRIT1	PIK3CA	RASA1	DLL4	EFNB2	EPHB4	HEY2	JAG1	JAG2	NOTCH1	NOTCH4
p_linear_davies_burden	0,38221288	0,01537437	1	0,21928752	0,20745303	1	0,33658777	0,71209805	1	1	0,62377606	0,14942144	0,18463992
p_linear_weigthed_liu_burden	0,55267682	0,62500188	0,6876229	0,20172749	0,24523592	1	1	0,07466587	0,26351762	0,47320537	1	0,17688622	0,16273141
p_linear_weigthed_liumod_skat	0,10859436	0,2535025	0,21814151	0,11396895	0,27007496	0,39392602	0,46291506	0,00103212	0,15369318	0,07021939	0,62885773	0,74293671	0,06243885
p_weigthed_quadratic_liumod_skat	0,06720779	0,00091534	0,27431324	0,07265253	0,47001171	0,47817634	0,32348029	0,63708082	0,16049165	0,41887425	0,72728534	0,87781225	0,30217909
p_linear_liu_skat	0,07335049	0,00130255	0,23005355	0,12848299	0,32147321	0,3891476	0,29791663	0,4589855	0,1240725	0,38855502	0,69176798	0,74534383	0,29692654
p_linear_IBS_liu_skat	0,0671458	0,00085927	0,26125466	0,10173943	0,40013283	0,46349484	0,3227096	0,3598335	0,09633605	0,34408232	0,6824734	0,78363192	0,15034067
p_linear_weigthed_skato	0,1790352	0,39128053	0,36099739	0,18649095	0,34289787	0,57858457	0,65589619	0,00233599	0,25231261	0,12568903	0,83167136	0,30570018	0,11023869
p_linear_skato	0,1222507	0,00268391	0,36983426	0,21400218	0,30080043	0,55902447	0,37676858	0,57574995	0,19499923	0,59987503	0,82546828	0,26410227	0,30533972

Fig. B.24 Table of the p values obtained for each test and each gene. Here the filters are activated along with $exac\_af < 0.01$ and $gonl\_af < 0.01$.

P value	GNA11	GNAQ	KRIT1	PIK3CA	RASA1	DLL4	EFNB2	EPHB4	HEY2	JAG1	JAG2	NOTCH1	NOTCH4
p_linear_davies_burden	0,19778522	0,14045342	0,38012176	0,61927417	0,01036924	0,59018457	0,00823388	0,21302795	0,40648402	0,35190266	0,31022418	0,02493966	0,51819941
p_linear_weigthed_liu_burden	0,02851367	0,3787007	0,2548249	0,3200074	0,56723065	0,74643499	0,64251134	0,43862505	0,54748444	0,68686853	1	1	0,91161061
p_linear_weigthed_liumod_skat	0,13044284	0,0620965	0,515992	0,10198922	0,47309268	0,28176345	0,6729261	0,07817085	0,11277715	0,10779156	0,53927521	0,14810384	0,43106039
p_weigthed_quadratic_liumod_skat	0,02971382	0,08592541	0,01887154	0,38387765	0,00165974	0,05570772	0,15330318	0,53155577	0,16269787	0,19079885	0,12589846	0,01603076	0,43091789
p_linear_liu_skat	0,01933955	0,02886888	0,01299935	0,70751737	0,00769357	0,05208134	0,14328634	0,31765356	0,12960368	0,08683909	0,03884703	0,0155978	0,19522448
p_linear_IBS_liu_skat	0,02299065	0,01205135	0,02435907	0,81151602	0,01528659	0,10097445	0,16387989	0,36160146	0,06789337	0,05196247	0,05112148	0,02181345	0,26150185
p_linear_weigthed_skato	0,0499848	0,10120335	0,42061511	0,17018677	0,64133831	0,44604967	0,84292134	0,13885542	0,18686803	0,18587929	0,73363259	0,25692214	0,64095282
p_linear_skato	0,03601862	0,05087082	0,02313863	0,73362547	0,00934012	0,09185379	0,01722883	0,32465283	0,21208878	0,16035909	0,06158012	0,0187172	0,28450153

Fig. B.25 Table of the p values obtained for each test and each gene. Here without any filters.

P value	GNA11	GNAQ	KRIT1	PIK3CA	RASA1	DLL4	EFNB2	EPHB4	HEY2	JAG1	JAG2	NOTCH1	NOTCH4
p_linear_davies_burden	0,10696683	0,09055896	0,74441685	0,05374838	0,02593748	1	0,10541176	0,74160743	0,77429845	0,30977749	0,27562854	0,10360298	0,80525756
p_linear_weigthed_liu_burden	0,1554593	0,48725582	0,81106626	0,04453394	0,62116891	1	0,33810419	0,36566743	0,18640811	0,45980695	0,48291908	0,12032799	0,71799094
p_linear_weigthed_liumod_skat	0,10020376	0,11005674	0,45854581	0,03752315	0,45825225	0,648791	0,56941221	0,03890041	0,08550837	0,11269307	0,70362501	0,79978187	0,11972548
p_weigthed_quadratic_liumod_skat	0,08458751	0,23245267	0,64139703	0,02988717	0,00088862	0,82893196	0,35229961	0,3121817	0,33902633	0,16700019	0,59545789	0,93783418	0,33414744
p_linear_liu_skat	0,08168345	0,10787385	0,48646065	0,05312798	0,00238455	1	0,29535906	0,30545387	0,44804699	0,11988749	0,4728492	0,80111043	0,32317319
p_linear_IBS_liu_skat	0,08488659	0,0217937	0,49689182	0,03738503	0,00290287	0,82004711	0,32793052	0,26823714	0,20515664	0,04983217	0,53430587	0,84374778	0,3936877
p_linear_weigthed_skato	0,17167951	0,18322305	0,68141512	0,05893372	0,64679962	0,82990394	0,52476501	0,07416088	0,15089578	0,19887452	0,68589043	0,21175574	0,2098092
p_linear_skato	0,13069374	0,13018933	0,67866289	0,07777686	0,00465994	1	0,16622354	0,47181944	0,663298	0,21541437	0,42677127	0,18739116	0,52495222

Fig. B.26 Table of the p values obtained for each test and each gene. Here the only filtering options are $exac\_af < 0.05$ and $gonl\_af < 0.05$.

Appendix C

Results of section 5.3 : a real data application

This appendix shows the results presented in section 5.3.

For this analysis, we used the following set up :

- Number of cases : 114
- Number of controls in the basic configuration : 135
- Number of controls in the restricted configuration : 67
- Genetic region to be tested : a entire gene
- Number of region to be tested : 93
- different filtering combination to be tested : 3

The first set of filters used is as follow

1. $read\_depth > 5$
2. $filters =$ PASS
3. $exac\_af \leq 0.01$
4. $gonl\_af \leq 0.01$

The second set of filters reuse all the previous filters and add the following ones

1. $consensus\_prediction \geq 2$ OR $cadd\_phred \geq 15$

2. *snpeff_impact* IN (High, Moderate)

Let's note that the OR and IN operator in filter 2 and 3 above are simply MySQL command in order to perform specific filters.

Finally, the last set of filters is similar to the second one (also reuses all the filters present in set 1 and 2), but slightly modifies the first criteria of set 2 as follows

1. *consensus_prediction* ≥ 4 OR *cadd_phred* ≥ 20

Here two group of controls where used. The first one is made of more individuals, but having slightly different diseases, while the second group of controls is more homogeneous.

C.1 Summary and context of current investigation

C.1.1 Abstract

Human vascular anomalies occur in 4-5 % of the population and can contribute to pain, disfigurement, and death. We performed a forward genetic screen in mice to identify causative mutations. Mice expressing Sleeping Beauty transposase in the endothelium developed a broad range of vascular lesions in multiple organs. Sequencing of genomic DNA from the lesions identified recurrent transposon insertions in 101 loci. Rasa1, Skint5, Lrch1 and Nf1 were the most mutated genes. Candidate genes populated pathways that included regulation of the cytoskeleton, focal adhesions, cancer signaling, and the hippo pathway. Exome sequencing of human vascular malformations and tumors identified damaging variants in genes predicted by the mouse genetic screen, including combinations of mutated genes that were simultaneously found in individual samples. Thus, the screen provided an important subset of individual genes, as well as gene combinations that are likely causative of sporadic human vascular disease. Furthermore, our results highlight a network of interrelated pathways that control vascular normalcy.

C.1.2 Confidential excerpt from paper

In the present study, transposition was restricted to the endothelial compartment by activating transposase expression. Two screens, Onc2 and Onc3, were conducted with distinct transposon constructs. The transposons contained a viral promoter/enhancer and splicing signals that, when spliced into the mRNA of a given gene, could cause overexpression or premature truncation of causative proteins. The transposon construct also includes polyA sequences capable of truncating gene products. In this manner, both gain- and loss-of-function mutations could occur.

Overall, the screens yielded vascular lesions from which genomic DNA was isolated to identify causative mutations. Unique sequences in the transposons allowed for precise mapping of transposon-induced mutations as determined using linker-mediated PCR and Illumina next-generation sequencing. Mapping of transposon orientation and distribution to the mouse genome revealed that the majority of mutations were loss-of-function in both screens, reflecting the outcome of known vascular anomaly inherited mutations. The Onc2 screen yielded 28 gCIS within 25 lesions and the Onc3 screen identified a total of 80 gCIS in 85 lesions.

Pathway analysis indicated that many of the genes identified in the screen were regulators of focal adhesions and the actin cytoskeleton.

Additional analysis revealed that some genes were frequently mutated in combination with others. Other genes were concurrently mutated more than 70 % of the time. Similar relationships were found in the T2/onc2 screen. Additional evaluation of the relationships between the gCIS co-occurring in lesions revealed some mutually exclusive gene networks. In this manner, we identified seven networks.

To ascertain whether the genes identified in the mouse screens could contribute to human vascular anomalies, we analyzed exome sequencing data from patient tissues and blood, for rare germline and somatic mutations in the genes identified in the mouse screen cohort. Importantly, we found an impressive number of variants in the human samples that corresponded to the list of identified gCIS isolated in our screen. Importantly, several variants were found in combination. These data are of relevance as it highlights potential epistasis or combinatorial inputs that are required to impose changes in cell function necessary for the emergence of vascular anomalies. In fact, placing the genes emanating from our screen in the context of known information from the literature, we found critical groups of regulatory pathways that emerge as nodes to maintain vascular normalcy. Placing genes identified in the mouse screen in context with the literature facilitates the building of networks of interacting genes that are likely to play critical roles in the maintenance of vascular integrity, hierarchy, and quiescence. Much like the Knudson multiple hit hypothesis, it is likely that not a single gene, but rather a combination of genes affecting several pathways, are necessary to impose sufficient level of destabilization for the emergence of vascular lesions.

The next sections will present the main results via summary tables and graphs in order to facilitate the data interpretations.

C.2 Computational analysis

Test	Average time	Total time
p_linear_davies_burden	0,0493	4,5807
p_linear_weigthed_liu_burden	0,0517	4,8074
p_linear_weigthed_liumod_skat	0,1935	17,9926
p_weigthed_quadratic_liumod_skat	0,0243	2,2630
p_linear_liu_skat	0,1909	17,7515
p_linear_IBS_liu_skat	0,0244	2,2714
p_linear_weigthed_skato	0,6977	64,8855
p_linear_skato	0,7084	65,8830

Fig. C.1 Summary of the average and total computational time (computed on the 93 tests) required for each test in the basic experiment (first set of filters)

C.3 P value analysis

	Minimum pvalue	Maximum pvalue	Standard deviation
p_linear_davies_burden	0,000082300	1,0000	0,27594
p_linear_weigthed_liu_burden	0,000082300	0,8730	0,26714
p_linear_weigthed_liumod_skat	0,000000011	0,9973	0,30558
p_weigthed_quadratic_liumod_skat	0,000000075	0,8973	0,25584
p_linear_liu_skat	0,000000191	1,0000	0,32000
p_linear_IBS_liu_skat	0,000000611	1,0000	0,27981
p_linear_weigthed_skato	0,000000075	0,9040	0,28485
p_linear_skato	0,000001040	1,0000	0,27974

Fig. C.2 Summary of the minimum, maximum and standard deviation based on the 93 p value computed by each test on each gene for the basic experiment (using first set of filters)

C.4 Comparing the ranking between the different class of tests

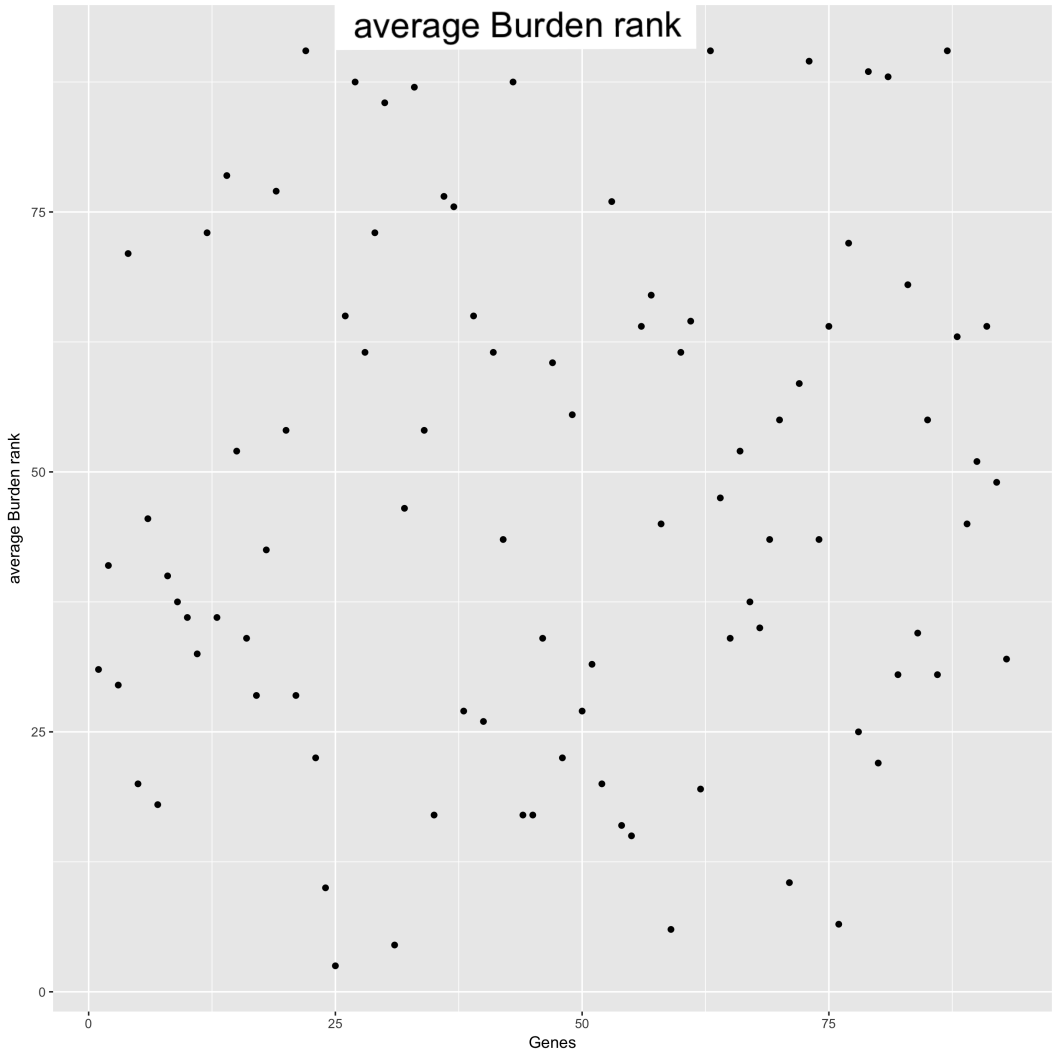


Fig. C.3 Plot of the average ranking (y axis) based on the burden tests for each gene (x axis) for the basic experiment (using first set of filters)

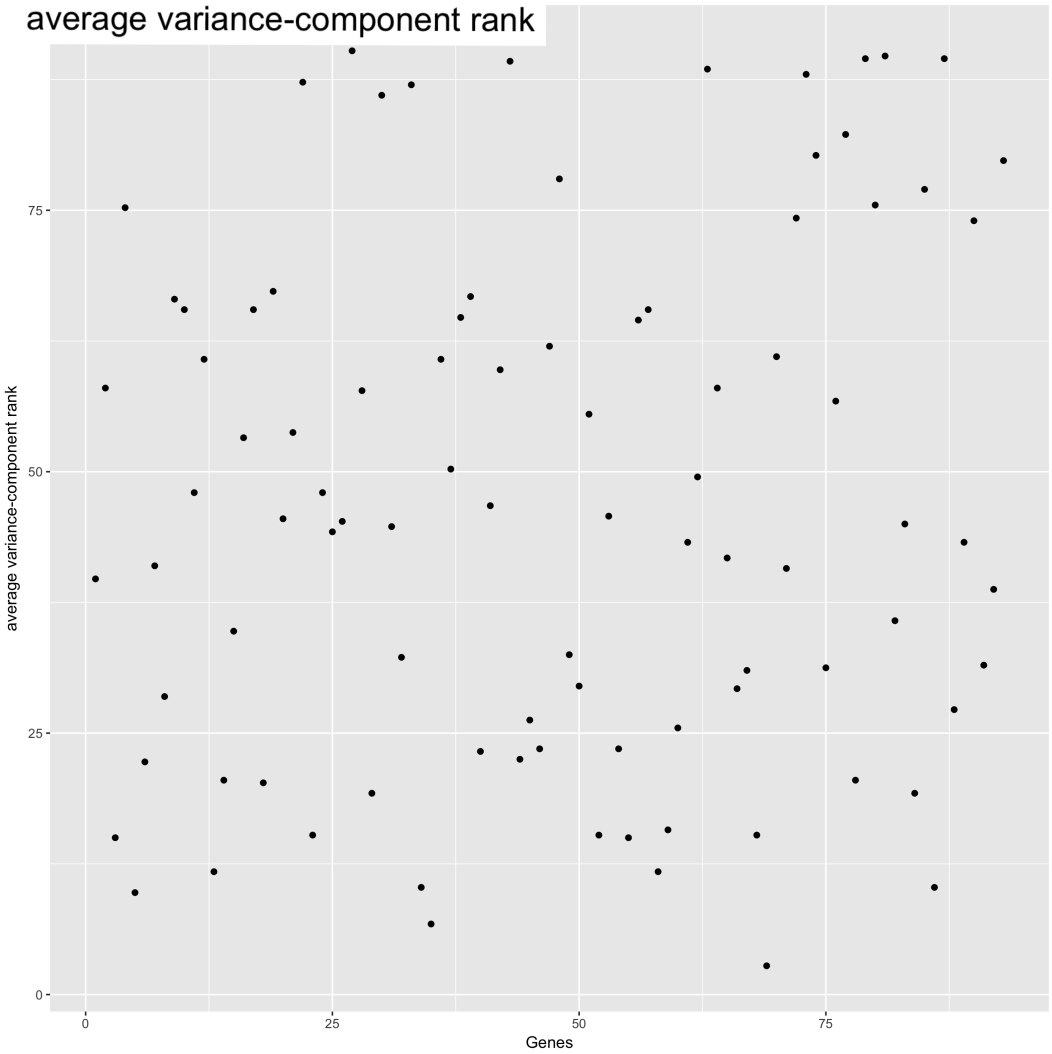


Fig. C.4 Plot of the average ranking (y axis) based on the SKAT tests for each gene (x axis) for the basic experiment (using first set of filters)

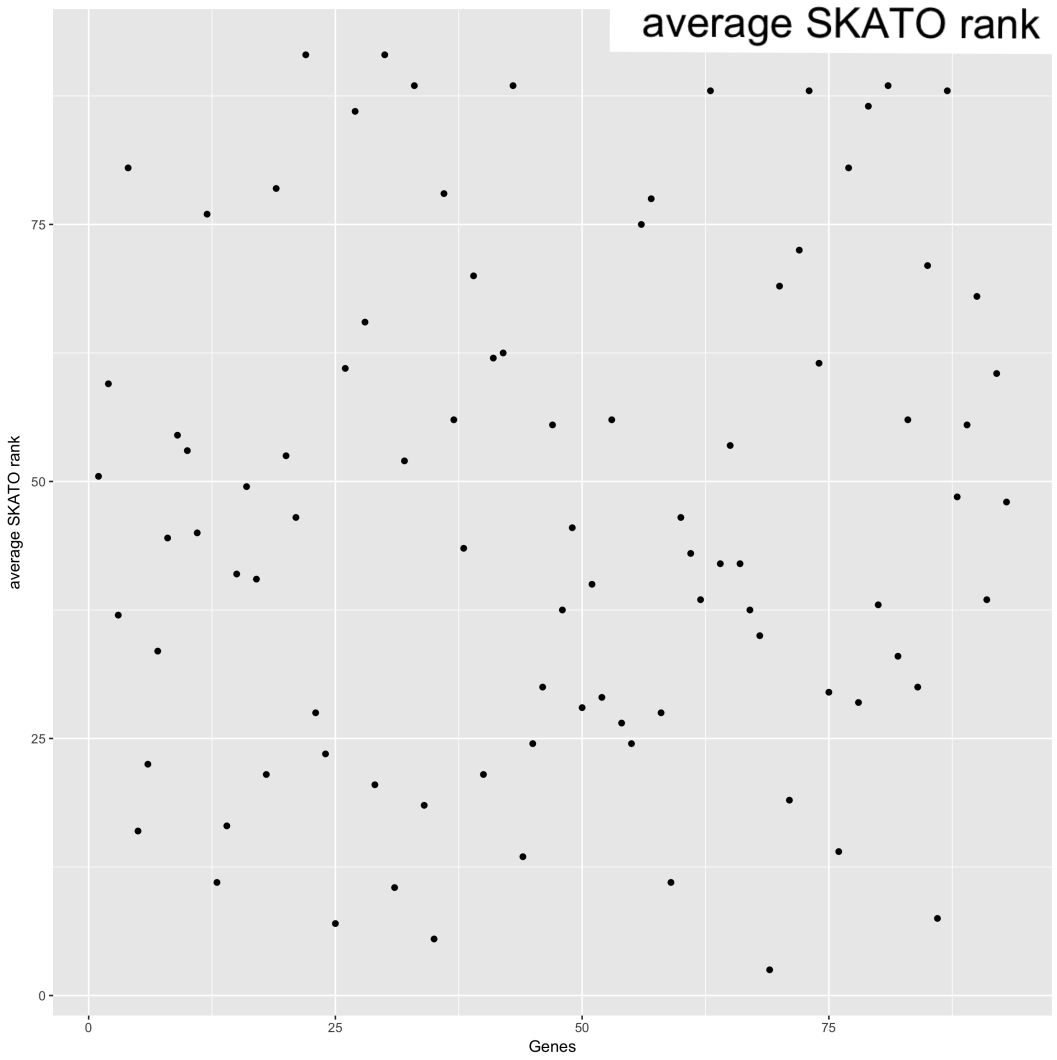


Fig. C.5 Plot of the average ranking (y axis) based on the SKATO tests for each gene (x axis) for the basic experiment (using first set of filters)

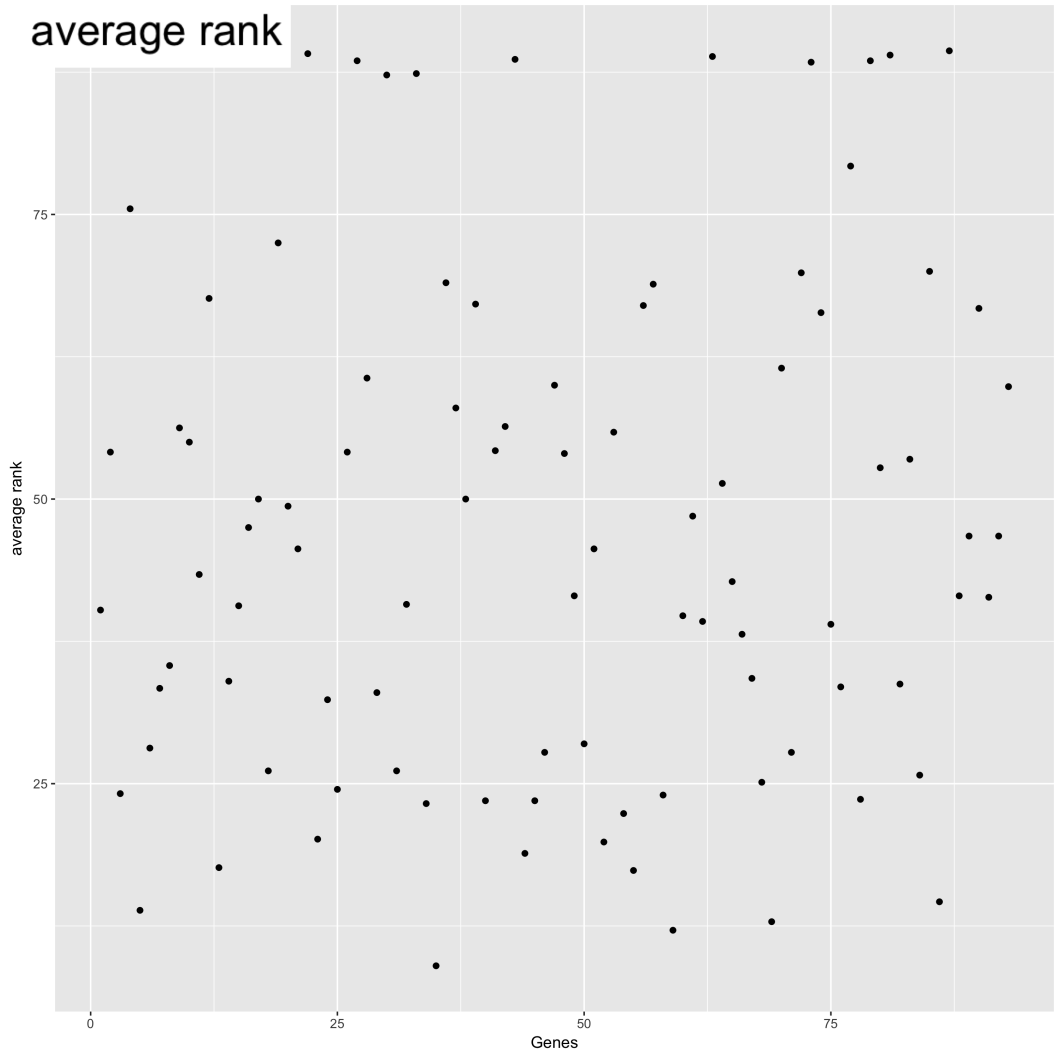


Fig. C.6 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the basic experiment (using first set of filters)

C.5 Comparing the ranking between the different experiments

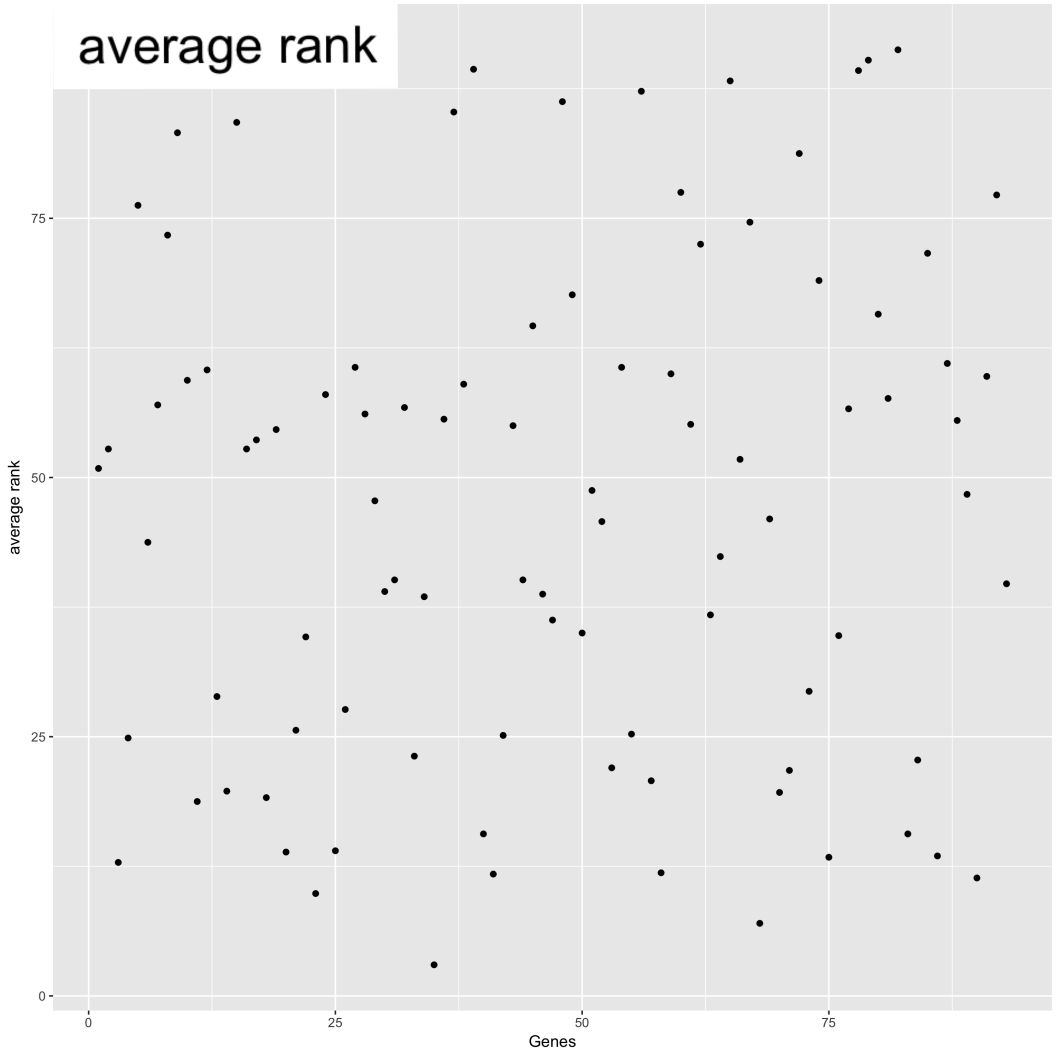


Fig. C.7 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the basic experiment (using second set of filters)

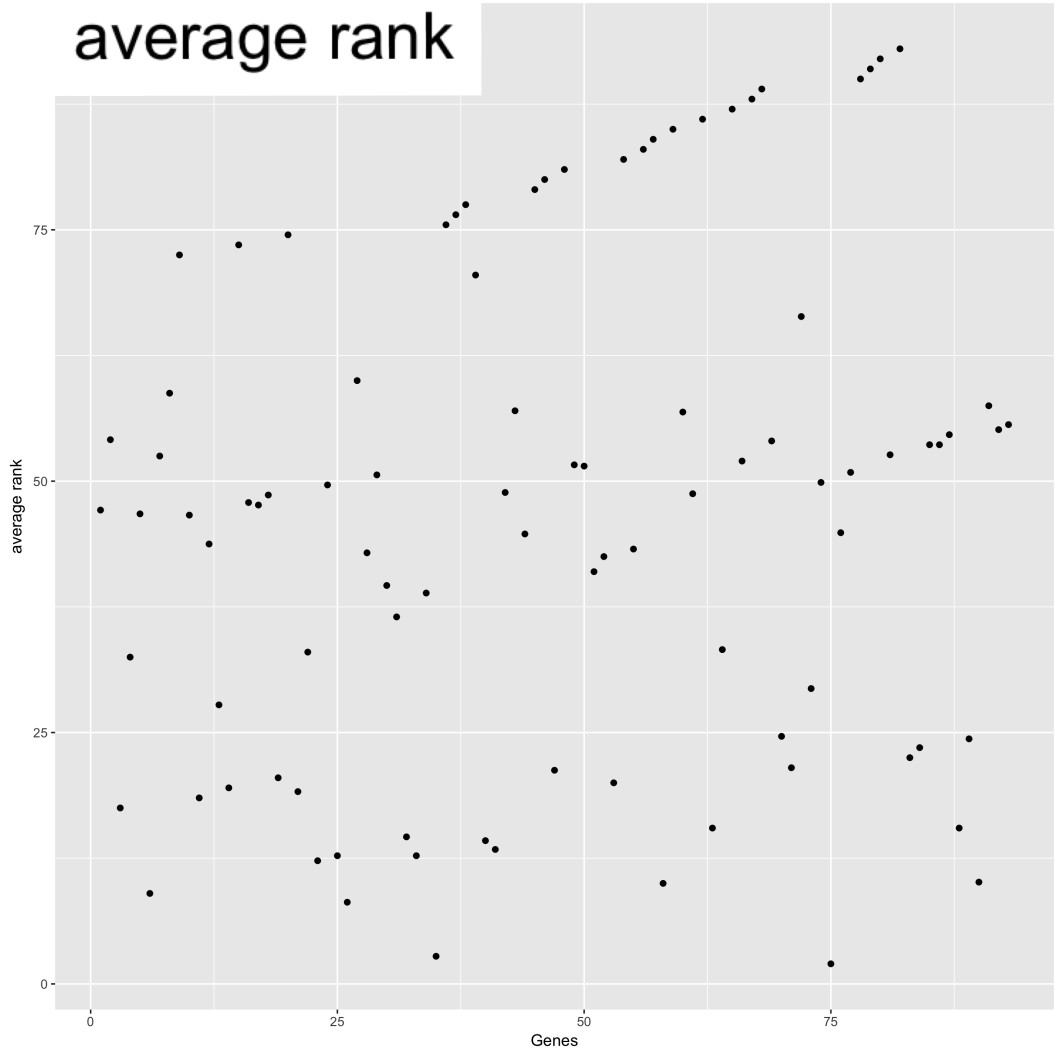


Fig. C.8 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the basic experiment (using third set of filters)

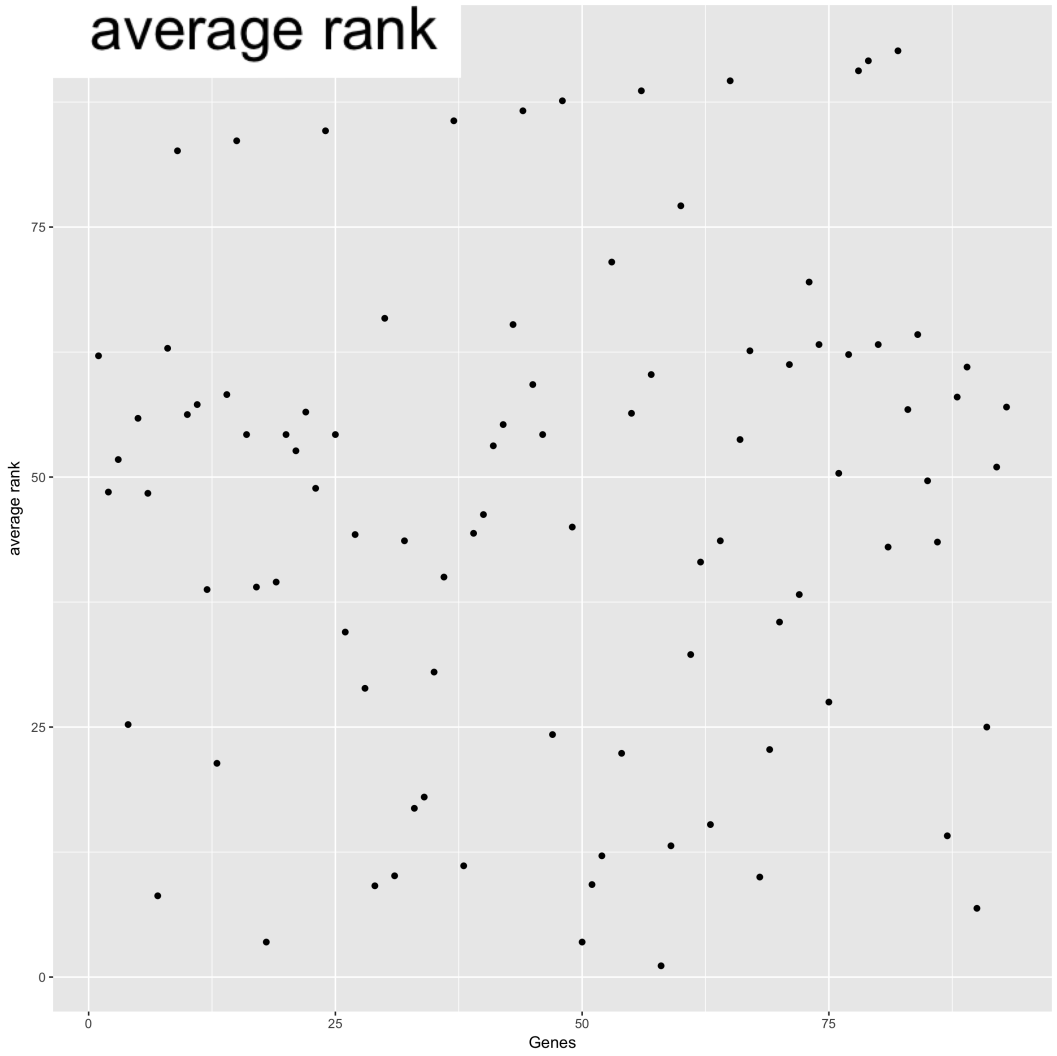


Fig. C.9 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the restricted experiment (using second set of filters)

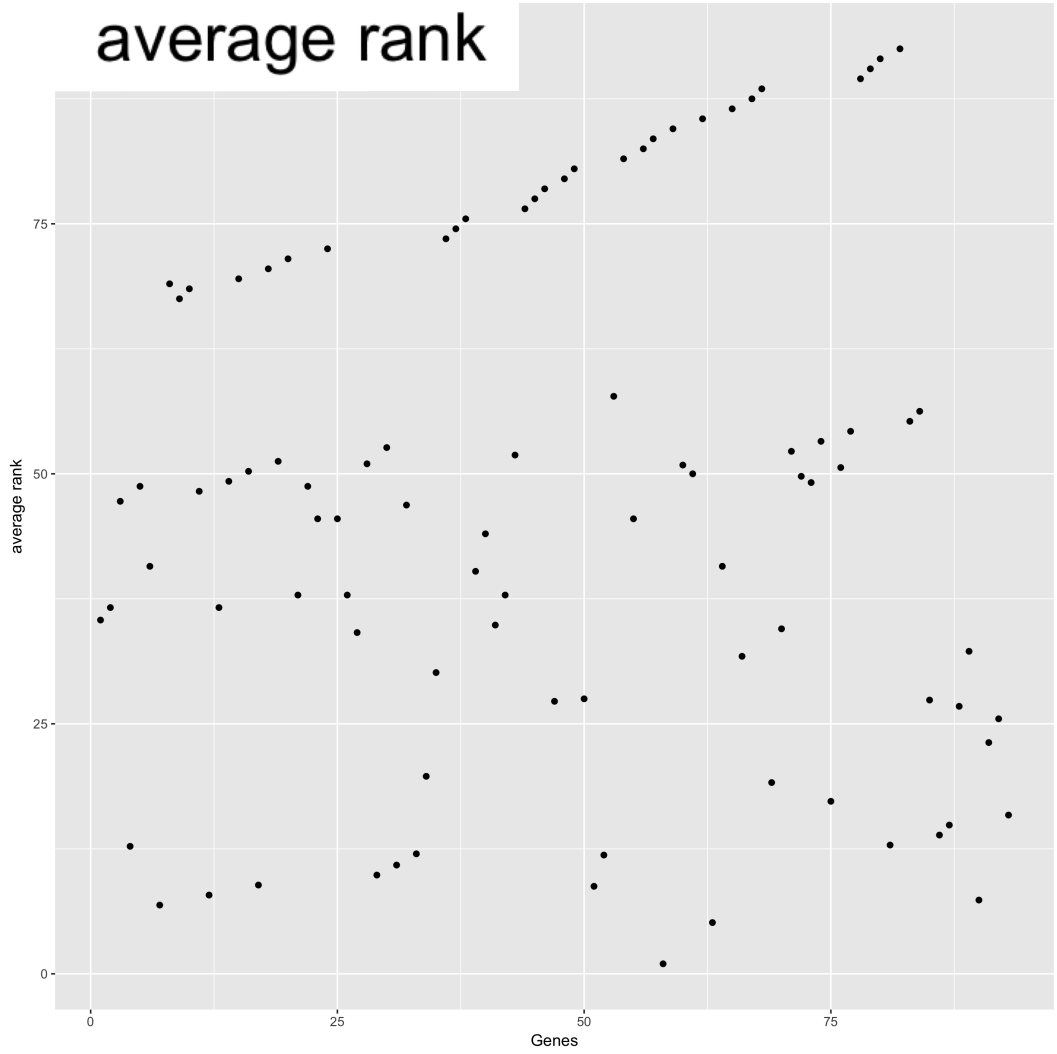


Fig. C.10 Plot of the average ranking (y axis) based on all the tests for each gene (x axis) for the restricted experiment (using third set of filters)

C.6 Summary of all experiments

Gene with pvalue < 5 %	basic	basic_cp2_Cadd15	basic_cp4_Cadd20	restricted_cp2_Cadd15	restricted_cp_Cadd20	Total Genes < 5 % per test
p_linear_davies_burden	20	6	5	2	1	34
p_linear_weigthed_liu_burden	10	7	5	2	1	25
p_linear_weigthed_liumod_skat	16	3	3	3	3	28
p_weigthed_quadratic_liumod_skat	21	1	0	1	0	23
p_linear_liu_skat	33	2	2	3	3	43
p_linear_IBS_liu_skat	29	1	0	1	0	31
p_linear_weigthed_skato	21	5	4	2	3	35
p_linear_skato	36	4	3	2	3	48
Total Genes < 5 % per experiment	186	29	22	16	14	
Total time for experiment in second	587,57	258,52	9,16	199,69	3,54	1058,48

Fig. C.11 Table showing the number of genes that have passed a tes with a pvalue $< 5\%$, per tests and also per experiment. Last row shows the total time taken per experiment.

C.7 Pair enrichment testing

Here, the presence of a gene in a sample is assumed to be independent of the presence of another gene. First of all let's compute, given the data present in *onc3_tumor_v_gene*, the expected number of observed pair of genes (given our hypothesis). As an illustration, let's take *Rasa*1 and *Skint*5. We observed 26 times *Rasa*1 across all 83 samples, and 14 time *Skint*5. Given that information, we know that the probability of finding *Rasa*1 in a sample is given by

$$P(Rasa1) = \frac{26}{83}$$

and probability of finding *Skint*5 in a sample is given by

$$P(Skint5) = \frac{14}{83}$$

. Therefore, the probability of finding both genes should be given by

$$P(Rasa1 \cap Skint5) = P(Rasa1) \cdot P(Skint5) = 0,0528$$

. Using this information, we know that the expected number of pair of *Rasa*1 and *Skint*5 should be $0,0528 \cdot 83 = 4,385$.

Now that we have the expected population (by population I mean the number of times the pair of genes is present in one sample) and the observed population, we can compute the enrichment. Therefore, we can use several methods :

1. *Prop test* : Pearson's chi-square test for 2-sample test for equality of proportions with continuity correction
2. *Chisq test* : Pearson's Chi-squared test with Yate's continuity correction
3. *fisher test* : Fisher's Exact Test for Count Data
4. *Binom test* : Exact binomial test

Excepted for the *Chisq test*, we apply an unilateral test while we use a bilateral test for the *Chisq test*. Therefore we obtain 4 p values for each combination of genes (here we have 3081 possible pair of genes). In order to filter the obtained data, we decided to accept the result if the p value is such that $p\_value < 0.05$.

In order to classify the different pairs, we compute their ranking in each test and then compute the average rank.

C.7.1 Excel file description

1. *Sheet 1* : basic summary for each gene, with number of times observed across all samples, maximum number of times observed with another gene and proportion that the gene appears with the same one particular gene.
2. *Observed_number* is a 79×79 symmetric matrix containing the observed count for each pair of genes (off diagonal element) across all samples.
3. *expected_number* : expected number of times we are supposed to observe a given pair of genes (computation explained in previous section).
4. *pvalue_Prop_test_matrix* : upper triangular matrix containing the p value obtained from the Pearson's chi-square test.
5. *pvalue_binom_test_matrix* : upper triangular matrix containing the p value obtained from the exact binomial test.
6. *pvalue_chisq_test_matrix* : upper triangular matrix containing the p value obtained from the Pearson's Chi-squared test with Yate's continuity correction test.
7. *pvalue_fisher_test_matrix* : upper triangular matrix containing the p value obtained from Fisher's Exact test.
8. *count_matrix* : upper triangular matrix showing number of test passed with $p\_value < 0.05$ for each pair of genes.

9. *average_rank_matrix* : average rank across all 4 tests. Cells colored in red are the pair that also passed all 4 tests with $p\_value < 0.05$.
10. *rank_matrix_binom_test* : rank obtained for the binomial test.
11. *rank_matrix_chisq_test* : rank obtained for the Pearson's Chi-squared test with Yate's continuity correction test.
12. *rank_matrix_fisher_test* : rank obtained for the Fisher's Exact test.
13. *rank_matrix_Prop_test* : rank obtained for the Pearson's chi-square test.