CTAAAGATGATCTTTAGTCCCGGTTCGAA
TCTTTAGTCCCGGTTGATAACACCAACC
GTAATACCAACCGGGACTAAAGATCCCG
GGGACTAAAGTCCCACCCCTATATATATG

TTCAAAATTTCTTCAAAAAAGAGGGGAG
GTGATTACATACAAATCGGAGGTGCCTA
TTTGTCATACTACATTTGCACCTATGTTTT
GTAAGTTGATGAGAGAGAAAATGTGTGT

TTTGCTAAACAAGGTTTTATAAAATAGTTG
AAATAATAGAAAACAAACTAAAATGAAAAT
TATTACTTAACAAATAGTTTTTAAGAATTAT
AATAAAGATATCTTATAATTATTGTATGACT

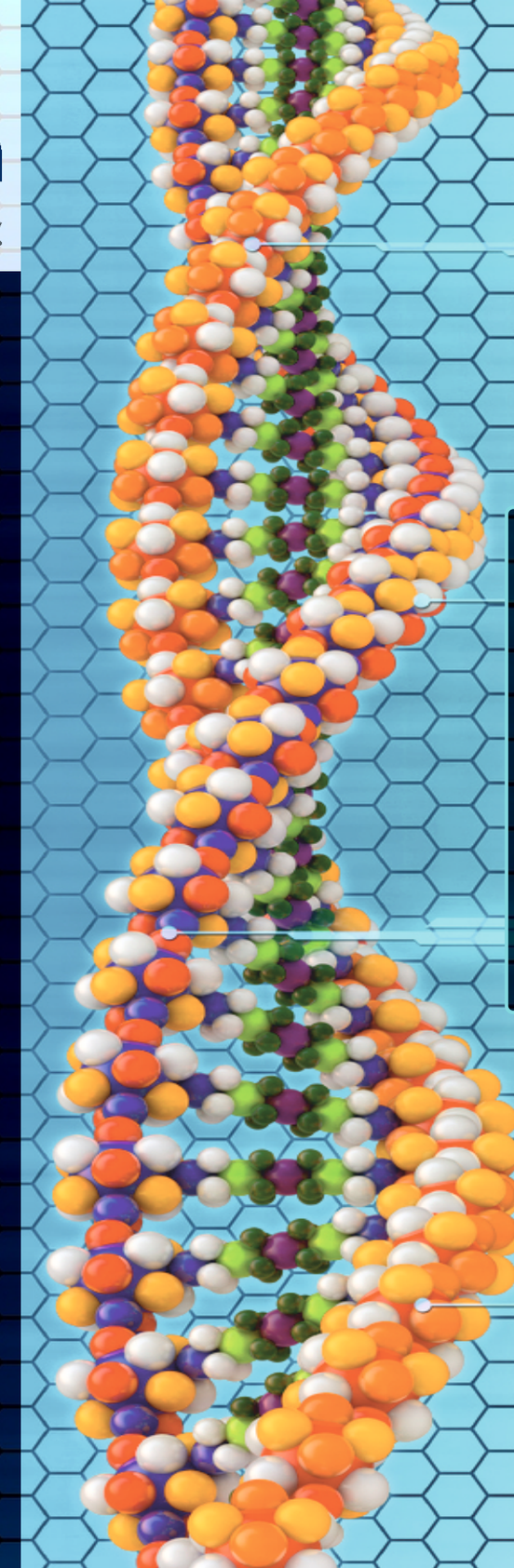
ACGGTTTTTTTGACTCATGTAGATGGATC
AGAGTTTATTGACGGCGTGCACTATTTTT
TTTTATTTGTTGTCCATGCAATAAGTGTAA
TATTCATTTCCACTTGTTTGAGTCGGGGT

ULPGC

ULPGC

PhD Thesis

Computational and statistical approaches for genotype imputation, haplotype reconstruction and analysis of genome variation



PhD Thesis

Computational and statistical approaches for genotype imputation, haplotype reconstruction and analysis of genome variation

Nathan Medina Rodríguez

Las Palmas de Gran Canaria

November, 2015

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada

Anexo I

D. PEDRO PÉREZ CARBALLO SECRETARIO DEL INSTITUTO UNIVERSITARIO DE MICROELECTRÓNICA APLICADA DE LA UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA,

CERTIFICA,

Que el Consejo de Doctores del Instituto en su sesión de fecha 23 de Noviembre de 2015 tomó el acuerdo de dar el consentimiento para su tramitación, a la tesis doctoral titulada "***Computational and statistical approaches for genotype imputation, haplotype reconstruction and analysis of genome variation***" presentada por el doctorando D. **Nathan Medina Rodríguez** y dirigida por los doctores **Ángelo Santana del Pino**, **Ana María Wägner Fahlin** y **José María Quinteiro González**.

Y para que así conste, y a efectos de lo previsto en el Artº 6 del Reglamento para la elaboración, defensa, tribunal y evaluación de tesis doctorales de la Universidad de Las Palmas de Gran Canaria, firmo la presente en Las Palmas de Gran Canaria, a 23 de Noviembre de 2015.

Anexo II

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Departamento/Instituto/Facultad:

Instituto Universitario de Microelectrónica Aplicada

Programa de doctorado:

Tecnologías de Telecomunicación

Título de la Tesis:

Computational and statistical approaches for genotype imputation, haplotype reconstruction and analysis of genome variation

Tesis Doctoral presentada por:

Nathan Medina Rodríguez

Dirigida por el Dr.:

Ángelo Santana del Pino

Dirigida por la Dra.:

Ana María Wägner Fahlin

Dirigida por el Dr.:

José María Quinteiro González

Director **Directora** **Director** **Doctorando**

Las Palmas de Gran Canaria, a 19 de Noviembre de 2015

En conformidad con los requerimientos
solicitados para la obtención del grado de

DOCTOR

POR LA UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

IUMA – Tecnologías de la Información
Dept. de Matemáticas – Grupo de Estadística

Computational and statistical approaches for genotype imputation, haplotype reconstruction and analysis of genome variation

Autor: **Nathan Medina Rodríguez**

Director: Dr. **Ángelo Santana**
Directora: Dra. **Ana Mª Wägner**
Director: Dr. **José Mª Quinteiro**

Las Palmas de Gran Canaria,
A 19 de Noviembre de 2015

A THESIS IN CONFORMITY WITH THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

IUMA – INFORMATION AND COMMUNICATION SYSTEMS

DEPARTMENT OF MATHEMATICS – GROUP OF STATISTICS

Computational and statistical approaches for genotype imputation, haplotype reconstruction and analysis of genome variation

Autor: **Nathan Medina-Rodriguez**

Supervisor: **Angelo Santana, PhD**

Supervisor: **Ana M. Wägner, MD, PhD**

Supervisor: **Jose M. Quinteiro, PhD**

Las Palmas de Gran Canaria,

November 19th, 2015

To my wife and family.

Acknowledgments

I would like to thank my PhD supervisors, who with their support, trust, help, and patience have facilitated the realization of this work.

First of all, I want to thank my mentor Angelo Santana, from whom I learned so much. He addressed my thesis tirelessly and unbreakable, guiding me in each of my steps. Thanks for your time, for your support, for your sincere friendship, for your knowledge; in short, thank you for everything.

To Ana M Wagner, for her help and for always being present when I needed her, advising me and facilitating me all necessary to conduct my research tasks.

To Jose M Quinteiro, for allowing me to join his group as a doctoral student and for his timely comments on the design of our research.

In addition, I would also like to thank the Department of Mathematics, especially to the Group of Statistics at the Universidad de Las Palmas, and to the Institute of Applied Microelectronics for facilitating me the development of my thesis and for helping with all administrative tasks.

I also extend special thanks to Dr. Torben Hansen of *The Novo Nordisk Foundation Center for Basic Metabolic Research* at the University of Copenhagen, for making me feel at home.

Thanks to my family for supporting me in every one of my journeys.

Thanks to Elizabeth for being my most faithful companion.

Finally, my sincere gratitude to all the studies' participants, since without their contribution and willingness our research would not have been possible.

Part of this work has been supported by the European Foundation for the Study of Diabetes with an *Albert Renold Travel Fellowship* grant.

Nathan Medina-Rodriguez
Las Palmas de Gran Canaria
November 2015

Agradecimientos

Me gustaría agradecer a los directores de esta Tesis Doctoral, quienes con su apoyo, confianza, ayuda y paciencia han facilitado la realización de este trabajo.

En primer lugar quiero agradecer a mi mentor el profesor Ángelo Santana, de quien tanto he aprendido, quien sin duda hace honor a su nombre ya que ha sido para mí como un "ángel", puesto que ha dirigido mi tesis de forma incansable e inquebrantable, guiándome en cada uno de mis pasos. Gracias por tu tiempo, por tu apoyo, por tu sincera amistad, por tu conocimiento... en definitiva, gracias por todo.

A la doctora Ana Mª Wägner, por su ayuda y estar siempre presente cuando la necesitara, asesorándome y proporcionándome las herramientas necesarias para llevar a cabo esta investigación.

Al profesor José Mª Quinteiro, por haberme dado la oportunidad de incorporarme a su grupo como alumno de doctorado y por sus comentarios oportunos en el diseño de nuestra investigación.

Además, me gustaría también dar las gracias al departamento de Matemáticas y en especial al grupo de Estadística de la Universidad de Las Palmas de Gran Canaria y al Instituto Universitario de Microelectrónica Aplicada por facilitar el desarrollo de mi tesis y ayudarme con todas las gestiones administrativas.

También extiendo un agradecimiento especial al Dr. Torben Hensen del *Novo Nordisk Foundation Center for Basic Metabolic Research* de la Universidad de Copenhague, por hacerme sentir como en casa.

Gracias a mi familia por apoyarme en cada una de mis travesías.

Gracias a Elizabeth por ser mi más fiel compañera de viaje.

Finalmente, mi más sincero agradecimiento a todos los participantes de los estudios, ya que sin su contribución y voluntad no habría sido posible nuestra investigación.

Parte de este trabajo ha sido apoyado por la Fundación Europea para el Estudio de la Diabetes con la beca *Albert Renold Travel Fellowship*.

Nathan Medina Rodríguez
Las Palmas de Gran Canaria
Noviembre de 2015

Abstract

The work developed in this PhD thesis has been primarily based on the research and development of various statistical and computational techniques aimed at solving some of the problems that arise in the context of the analysis of large databases that are currently used in genetics. Such work has enabled to obtain interesting results using both familial and population genetic data.

Initially, in this thesis has been performed a comprehensive review of the current statistical methods used in Genetics and Bioinformatics to assess the association between genome and disease.

Then, we used advanced computational methods to identify genetic variations associated with phenotypes of interest in *Genome-Wide Association Studies*, and we have prepared a tutorial with detailed instructions for performing such association studies, including procedures for quality control data, sequence alignment, genotype imputation, pre-phasing, and statistical analysis of association.

After the application of corresponding methods and data analysis, we have identified new associations of genetic variants associated with advanced diabetic nephropathy in patients with *Type 2 Diabetes* in the population of the Gran Canaria Island.

On the other hand, we have also developed an R package called alleHap, for simulation, imputation and deterministically reconstruction of null recombinant haplotypes from pedigrees databases by crossing genotypic information between parents and offspring.

Using alleHap package, we have identified high risk haplotypes in the *Human Leukocyte Antigen* (HLA) genomic region that are included in the *Type 1 Diabetes Genetics Consortium* database. We also conducted an analysis of maternal factors associated with the early childhood development of *Type 1 Diabetes*, and finally we compared the frequency of these risk haplotypes in subjects from the Canary Islands regarding individuals from the rest of Spain and Europe.

Resumen

El trabajo de esta tesis se ha basado fundamentalmente en la investigación y desarrollo de varias técnicas estadístico-computacionales orientadas a la resolución de algunos de los problemas que se plantean en el contexto del análisis de las grandes bases de datos que se utilizan actualmente en genética. Dicho trabajo ha permitido la obtención de interesantes resultados usando tanto con datos genéticos de origen poblacional como familiares.

Inicialmente, en esta tesis se ha realizado una revisión muy completa de los métodos estadísticos en Genética y Bioinformática utilizados en la actualidad para evaluar la asociación entre el genoma y las enfermedades.

Seguidamente, se han utilizado métodos computacionales avanzados para identificar variaciones genéticas relacionadas con fenotipos de interés en estudios de asociación genómica (GWAS) y se ha elaborado un tutorial con instrucciones detalladas para la realización de dichos estudios de asociación, incluyendo procedimientos para el control de calidad de los datos, el alineamiento de secuencias, la imputación e identificación de la fase de genotipos, y el análisis de asociación estadístico.

Tras la aplicación de los métodos correspondientes y el análisis de datos, se han identificado nuevas asociaciones de variantes genéticas asociadas a la nefropatía diabética avanzada de sujetos con Diabetes Tipo 2 en la población de la isla de Gran Canaria.

Por otro lado, también se ha desarrollado un paquete computacional en lenguaje R llamado alleHap, que es capaz de imputar alelos e identificar (reconstruir) haplotipos de manera determinista e inequívoca en bases de datos familiares mediante el cruce de la información genotípica (no recombinante) entre padres e hijos.

Finalmente, mediante la utilización del citado paquete bioinformático, se han identificado haplotipos de riesgo en la región genómica HLA (Human Leukocyte Antigen) los cuales están incluidos en la base de datos (T1DGC, Type 1 Diabetes Genetics Consortium), se ha realizado un análisis de factores maternos asociados con el desarrollo precoz y en la infancia de la diabetes tipo 1, y se han comparado la frecuencia de los haplotipos de riesgo en una muestra de sujetos de las Islas Canarias con respecto a muestras del resto de España y de Europa.

Contents

I Introduction 17

1 Introduction 19
1.1. Background 19
1.2. Motivation and Original Contributions 20
1.3. Document Structure 21

II Basic Concepts and State of the Art 23

2 Concepts of Human Genetics 25
2.1. Molecular Genetics Terminology 25
2.2. Transmission of Genetic Information 31
2.3. Human Genetic Diversity 33

3 Concepts of Statistical Genetics 39
3.1. Statistical Genetics 39
3.2. Population Studies 40
3.3. Family-based Studies 50

4 Concepts of Computational Genomics 55
4.1. Computational Genomics 55
4.2. Computational Biology vs. Bioinformatics 55
4.3. Machine Learning in Biology 56
4.4. Applied Computational Techniques 58
4.5. International Genetic Databases 63

5 State of the Art 69
5.1. Introduction 69
5.2. Genotype Imputation 70

5.3. Haplotype Reconstruction 76
5.4. Analysis of Genome Variation 83

III Approaches for Population Data 87

6 Quality Control 89
6.1. Introduction 89
6.2. Marker Quality Measures 92
6.3. Sample Quality Measures 95

7 Alignment and Phasing 103
7.1. Introduction 103
7.2. Alignment 107
7.3. Phasing 109

8 Imputation 113
8.1. Introduction 113
8.2. Imputation 115

9 GWA Testing of Imputed Data 121
9.1. Introduction 121
9.2. Association Method Description 123
9.3. Association testing of imputed data 126
9.4. Results using HapMap as Reference Panel 129
9.5. Results using 1000 Genomes as Reference Panel 130
9.6. Selection of Significant Results 133

IV Approaches for Family-based Data 141

10 alleHap Package: Description 143
10.1. Introduction 143
10.2. Theoretical Description 143
10.3. Practical Description 147

11 alleHap Package: Performance 171
11.1. Computing Times 171
11.2. Genotype Imputation Rates 175
11.3. Reconstructed Haplotypes 182

12 alleHap Package: Applications **191**
12.1. alleHap into T1DGC database 191
12.2. Comparison of the distribution of risk haplotypes between the Canary Islands and the rest of Spain 201
12.3. Comparison of the distribution of risk haplotypes between Spain and the rest of Europe 202

v Conclusion **205**

13 Main conclusions **207**

vi Appendix **211**

A GWA Tutorial **213**
A.1. Quality Control 214
A.2. Pre-processing 218
A.3. Imputation 220
A.4. GWA Analysis 222
A.5. Data Representation 224

B alleHap Manual **227**
B.1. Input Format 227
B.2. Data Simulation 228
B.3. Workflow 231

vii Resumen en Español **243**

14 Introducción **245**

15 Objetivos **249**

16 Planteamiento y Metodología **251**
16.1. Planteamiento 251
16.2. Metodología 251

17 Resultados **275**
17.1. Resultados del análisis de bases de datos poblacionales . 275
17.2. Resultados del análisis de bases de datos familiares . . . 287

18 Conclusiones **301**

Bibliografía **305**

List of Figures

2.1. From cell to gene 25
2.2. Karyogram of human chromosomes 27
2.3. DNA structural diagram 28
2.4. Alleles in chromosomes 30
2.5. Meiosis two stage scheme 32
2.6. Meiotic recombination 33
2.7. DNA Structural Variation 34
2.8. SNP diagram 35
2.9. MHC-HLA Complex 36
2.10. HLA Alleles Nomenclature 38

3.1. Hardy-Weinberg proportions for two alleles 42
3.2. Linkage and Linkage Disequilibrium 47
3.3. Indirect Association 50

4.1. Machine Learning Topics 57
4.2. HMM Architecture 59
4.3. HMM for Haplotypic Data 60
4.4. HMM example 61
4.5. Overview of the EM algorithm 62
4.6. SNPs, haplotypes and tag SNPs. 64
4.7. Pedigree structures into the T1DGC 67

5.1. Genotype imputation overview 70
5.2. Association example using imputed data 72
5.3. IMPUTE2 standard imputation scenario 74
5.4. Imputation scenarios 78
5.5. Phasing Methods Comparison 79
5.6. Haplotype correction example using DuoHMM 80

5.7. GWA published reports 83
5.8. Published Genome-Wide Associations 84

6.1. Flowchart of the Quality Control process 90
6.2. Genotyping Efficiency 93
6.3. QQ plot of HW control p-values 95
6.4. Sample missingness 97
6.5. Example of relatedness networks 99
6.6. Relatedness among samples 99
6.7. IBD Histograms 99
6.8. Heterozygosity Histograms before SQC 100
6.9. Heterozygosity Histograms after SQC 101

7.1. SHAPEIT method example 111

8.1. Standard Imputation Scenario 119

9.1. Manhattan plot using HapMap 129
9.2. QQ plot using HapMap 129
9.3. Manhattan plot using 1000G 131
9.4. QQ plot using 1000G 131
9.5. Manhattan plot using 1000G 132
9.6. QQ plot using 1000G 132
9.7. First region of Typed SNPs 136
9.8. First region of Typed/Imputed SNPs 136
9.9. Second region of Typed SNPs 137
9.10. Second region of Typed/Imputed SNPs 137
9.11. Third region of Typed SNPs 138
9.12. Third region of Typed/Imputed SNPs 138

10.1. Alleles, haplotypes and pedigree scheme 144
10.2. Package Description 148

11.1. Computing Times per Families 172
11.2. Computing Times per Markers 173
11.3. Computing Times per Alleles 174
11.4. Initial vs. Final Imputation rates 176
11.5. Imputation rates vs. missing genotypes -1/4- 176
11.6. Imputation rates vs. missing genotypes -2/4- 177
11.7. Imputation rates vs. missing genotypes -3/4- 178
11.8. Imputation rates vs. missing genotypes -4/4- 179

11.9. Imputation rates vs. number of alleles per marker 181
11.10Imputation rates vs. number of markers 182
11.11Reconstructed haplotypes vs. missing genotypes -2/4- . . . 183
11.12Reconstructed haplotypes vs. missing genotypes -3/4- . . . 184
11.13Reconstructed haplotypes vs. missing genotypes -4/4- . . . 185
11.14Reconstructed haplotypes vs. number of alleles per marker 186
11.15Reconstructed haplotypes vs. number of markers 187
11.16Reconstructed vs. number of markers 188

17.1. Eficiencia de Genotipado 277
17.2. Gráfico cuantil-cuantil de los p-valores de sujetos controles 278
17.3. Tasa de pérdidas por individuo 279
17.4. Ejemplo de redes de parentesco 281
17.5. Parentesco entre los sujetos del estudio 281
17.6. Histogramas de heterocigosidad antes del control de calidad de muestras 281
17.7. Histogramas de heterocigosidad después del control de calidad de muestras 282
17.8. Gráfico Manhattan usando 1000G y MINIMAC3 283
17.9. Gráfico Cuantil-Cuantil usando 1000G y MINIMAC3 . . 284
17.10Gráfico Manhattan usando 1000G y IMPUTE2 284
17.11Gráfico Cuantil-Cuantil usando 1000G e IMPUTE2 . . . 285
17.12Primera región con SNPs significativos 286
17.13Segunda región con SNPs significativos 286
17.14Tercera región con SNPs significativos 287

List of Tables

3.1. Genotype counting 43
3.2. LD allele distributions 48
3.3. *Transmission/Disequilibrium Test* (TDT) allele counting . . 51

4.1. 1000 Genomes Samples 66

5.1. Imputation methods comparison 76
5.2. Most used programs for genotype imputation and haplotype phasing. 81
5.3. R packages related to haplotype reconstruction. 81
5.4. Other programs related to haplotype reconstruction. 82
5.5. GWA imputation-based methods: error rates 85
5.6. GWA imputation-based association methods: accuracy . . 86

9.1. SNPTEST2 additional columns 122
9.2. SNPTEST2 additional fields 123
9.3. Association between genotype and disease 124
9.4. Association between genotype and disease after imputation 127
9.5. Most significant SNPs from *additive* test (without adjusting by any co-variable). 134
9.6. Most significant SNPs from *additive* test (adjusted by the *retinopathy* co-variable). 135

10.1. Biallelic configurations in a parent-offspring pedigree . . . 145
10.2. Haplotype configurations for scenario 2 160
10.3. Cases description of Scenario 4 164
10.4. Detailed case arrangement for Scenario 4 167

12.1. Distribution of DR3-DQ2 and DR4-DQ8 high risk haplotypes -1/2- 194

12.2. Distribution of DR3-DQ2 and DR4-DQ8 high risk haplotypes -2/2- 195
12.3. Estimation of linear model for child's onset age -1/7- . . . 197
12.4. Estimation of linear model for child's onset age -2/7- . . . 198
12.5. Estimation of linear model for child's onset age -3/7- . . . 198
12.6. Estimation of linear model for the child's onset age -4/7- . 199
12.7. Estimation of linear model for the child's onset age -5/7- . 200
12.8. Estimation of linear model for the child's onset age -6/7- . 200
12.9. Estimation of linear model for the child's onset age -7/7- . 201
12.10Frequencies of the DRB-DQA-DQB haplotypes in the Canary Islands and Peninsular Spain 202
12.11Frequencies of the DRB-DQA-DQB haplotypes in Spain vs. rest of European countries 203

B.1. Example of a Family in .ped file format 228

16.1. Configuraciones haplotípicas del escenario 2 263
16.2. Descripción de los casos para el escenario 4. 267
16.3. Detalle de los casos del escenario 4 271

17.1. Distribución de los haplotipos de riesgo -1/2- 291
17.2. Distribución de los haplotipos de riesgo -2/2- 292
17.3. Estimación del modelo lineal para la edad de debut -1/7- . 294
17.4. Estimación del modelo lineal para la edad de debut -2/7- . 295
17.5. Estimación del modelo lineal para la edad de debut -3/7- . 295
17.6. Estimación del modelo lineal para la edad de debut -4/7- . 296
17.7. Estimación del modelo lineal para la edad de debut -5/7- . 297
17.8. Estimación del modelo lineal para la edad de debut -6/7- . 297
17.9. Estimación del modelo lineal para la edad de debut -7/7- . 298
17.10Frecuencias de los haplotipos DRB-DQA-DQB de las islas Canarias con respecto a los del resto de España. 299
17.11Frecuencias de los haplotipos DRB-DQA-DQB en España con respecto a los del resto de países Europeos. 300

List of Acronyms

AAID Associated Autoimmune Disease 196

AFR African Ancestry 65

ASP Affected Sib-Pair 67

AMR Americas Ancestry 65

CHMM Compact Hidden Markov Model 110

CNV Copy Number Variant 71

DNA Deoxyribonucleic Acid 26

EAS East Asian Ancestry 65

EBI European Bioinformatics Institute 83

EUR European Ancestry 65

EM Expectation-Maximization 61

FBAT Family-Based Association Test 50

GEM Generalized Expectation Maximization 61

GWA Genome-Wide Association 21

GWAS Genome-Wide Association Studies 19

HLA Human Leukocyte Antigen vi

HMM Hidden Markov Model 58

HQ High Quality 76
HWE Hardy-Weinberg Equilibrium 41
IBD Identity-by-Descent 98
ID Identifier 227
k-NN k-Nearest Neighbors 57
LD Linkage Disequilibrium 46
MCMC Markov Chain Monte Carlo 115
MAF Minor Allele Frequency 85
MAP Maximum a Posteriori 61
MHC Major Histocompatibility Complex 19
MLE Maximum-Likelihood Estimation 62
NA Not Available 146
NHGRI National Human Genome Research Institute 83
NUMT Nuclear Mitochondrial Insertion 66
PDT Pedigree Disequilibrium Test 52
PED Pedigree 227
OR Odds Ratio 44
QC Quality control 89
QQ Quantile-Quantile 94
RNA Ribonucleic Acid 28
SD Standard Deviation 101
SNP Simple Nucleotide Polymorphism 35
SAS South Asian Ancestry 65
SQC Sample Quality Control 99

S-TDT Sib Transmission/Disequilibrium Test 52

SVM Support Vector Machine 57

T1D Type 1 Diabetes 66

T2D Type 2 Diabetes 20

T1DGC Type 1 Diabetes Genetics Consortium 20

TDT Transmission/Disequilibrium Test 10

VQC Variant Quality Control 99

PART

I

Introduction

Chapter 1

Introduction

1.1. Background

Genotype imputation and haplotype reconstruction have achieved an important role in *Genome-Wide Association Studies* (GWAS) during recent years. Estimation methods are frequently used to infer either missing genotypes as well as haplotypes from databases containing related or unrelated subjects. The majority of these analyzes have been developed using several statistical methods [1] which can impute genotypes as well as perform haplotype phasing (also known as haplotype estimation) of the corresponding genomic regions.

Currently, algorithms do not carry out genotype imputation or haplotype reconstruction using deterministic techniques on pedigree databases, despite the fact that computational inference by probabilistic models may cause some incorrect results. These methods are usually focused on population data. In the case of pedigree data, families typically are comprised by duos (parent-child) or trios (parents-child) [2], whereas those studies focused on more than two offspring (for each line of descent) are uncommon.

On the other hand, certain genomic regions are very stable against recombination but at the same time, they may be highly polymorphic. For this reason, in some well-studied regions, such as HLA loci [3] in the extended human *Major Histocompatibility Complex* (MHC) [4], an alphanumeric nomenclature is needed to facilitate later analysis. At this juncture, the available typing techniques usually are not able to determine the allele phase and, therefore, the constitution of the appropri-

ate haplotypes is not possible. Although some computational methods have been evaluated for the reconstruction of haplotypes [5], none of them is capable to perform haplotype phasing or genotype imputation of missing data without using reference panels.

Finally, although there is a growing number of bioinformatic/biostatistical tools for processing genomic databases, the documentation related to each of them is often somewhat confusing. Therefore, a need exists for clarification and simplification in the documentation relating to processes that include quality control, imputation of missing values and statistical analysis of association in genetic/genomic databases.

1.2. Motivation and Original Contributions

The motivation for the realization of this work came from the necessity of organizing and applying different biostatistical and bioinformatics methods to solve several problems posed by diverse research groups in the field of endocrinology at the *Complejo Hospitalario Universitario Insular Materno Infantil* of Las Palmas de Gran Canaria. These problems were related to diabetes genetics and used different kind of data. On one hand, family-type genetic data (genetic information of parents and children in a number of families) were selected and, in other, population data came from a case-control study. The data analysis process required knowledge not only of statistical and computing methods but also of the basics of human genetics and the recent developments in methodologies for treating genetic data. For that reason, this document includes an informative part that intends to summarize these concepts and ideas, previous to the development of our original contributions, which can be synthesized as:

1) **Identification of genetic variants** associated with advanced diabetic nephropathy in a *Type 2 Diabetes* (T2D) population from the Gran Canaria Island.

2) **GWAS Tutorial**: *Quality Control, Imputation, Analysis of population data.*

3) **Identification of haplotype associations** in the international *Type 1 Diabetes Genetics Consortium* (T1DGC) pedigree database.

4) Development of the **R package** alleHap.

5) **alleHap Manual**: *Allele Imputation and Haplotype Reconstruction from Pedigree Databases*.

Together with previous milestones, the author of this PhD thesis was also co-author of several publications in peer-reviewed journals as well as international proceedings/conferences.

1.3. Document Structure

This document has been structured in 6 parts, each one containing, at least, three chapters. A brief description of each one is listed as follows:

- **Part I** will explain the motivation for this PhD dissertation, the original contributions of its author and the document structure.
- Since this thesis covers diverse research areas, **Part II** intends to establish the state of the art, as well as to specify and clarify some basic concepts that may be useful for those who are not familiarized with genetics, biostatistics, and/or bioinformatics.
- **Part III** will cover all necessary topics to develop the identification of genetic variants associated with a phenotype (disease) in a population. To achieve this purpose, quality control, alignment, haplotype phasing, genotype imputation and association analysis of genomic data were implemented.
- **Part IV** will comprise a description of the alleHap package, an analysis of its performance and the study of its application in the T1DGC database.
- **Part V** will present the main conclusions of previous chapters.
- **Part VI** will consist of two tutorials/manuals, one for the management of *Genome-Wide Association* (GWA) data and other for proper utilization of the alleHap package.
- As completion of this PhD dissertation, **Part VII** will include a summary of this thesis in Spanish, containing proposed goals, methodology, original contributions and final conclusions.

PART

II

Basic Concepts and State of the Art

Chapter 2

Concepts of Human Genetics

2.1. Molecular Genetics Terminology

Some basic terms used in human genetics are important to define before going further in this dissertation. From **cell** to **gene**, essential concepts of molecular genetics are explained and clarified in this chapter. Some of these concepts are represented in Figure 2.1.

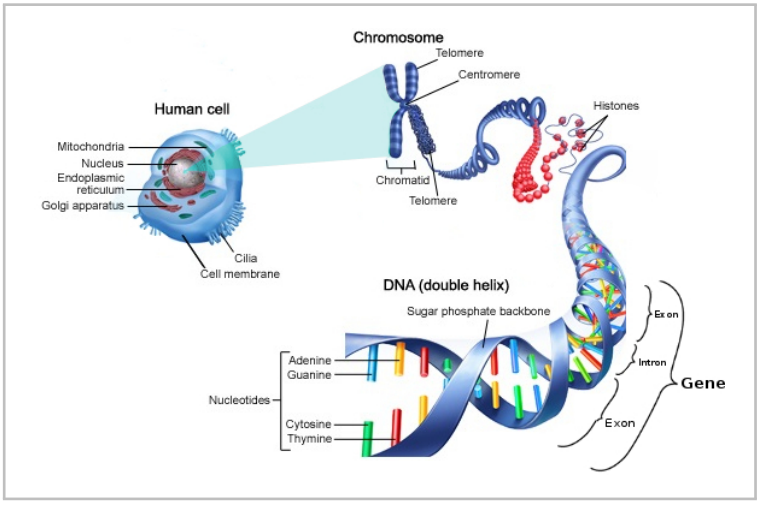


Figure 2.1: Cell, Chromosome, DNA and Gene representation, adapted from [6].

2.1.1. Cell

The **cell** is the basic morphological and functional unit of all known living organisms. Organisms can be classified as unicellular (consisting of a single cell; including most bacteria) or multicellular (including plants and animals). While the number of cells in plants and animals varies from species to species, humans contain about 100 trillion (10^{14}) cells [7]. Most plant and animal cells are visible only under the microscope, with dimensions between 1 and 100 micrometers [8].

Cells contain ***organelles*** (subcellular differentiated structures that perform specific functions) which work together to fulfil tasks needed for maintenance and growing. Among all of them stands the ***nucleus***, that contains the genetic material in the form of *Deoxyribonucleic Acid* (DNA) inside each chromosome. The nucleus is placed in the cytoplasm, the substance that also houses the rest of the organelles. Not all cells have nuclei, so these which have it are called **eukaryotes** and those which do not are named **prokaryotes** (their DNA reside in the cytoplasm).

2.1.2. Chromosome

All human cells (except red blood cells) have 23 **chromosomes**: 22 ***autosomes*** and one sex chromosome *(X or Y)*. Cells that contain one set of chromosomes, such as sperm or unfertilized egg cells, are said to be **haploid**. Fertilized eggs and most body cells derived from them are said to be **diploid** [9]. Chromosomes are structures comprised by double helix substructures of DNA that contain the information necessary for the cell operation and reproduction. The members of a chromosome pair are referred to as ***homologous***. Homologous chromosomes are matched up and arranged by size, from largest *(chromosome 1)* to smallest *(chromosomes 21 and 22)*, followed by the sex chromosomes to form a display called a **karyogram** [9] *(see Figure 2.2)* or **karyotype** *(if the chromosomes description is more detailed)*.

Chromosomes are often represented joined to each other at the middle, looking something like an H, as depicted in Figure 2.2. A chromosome is composed of two ***chromatids*** where each one typically is joined to the other copy by a single point called ***centromere*** [11]. The centromere also divides the chromosome into two arms, designated p (French pern) for the shorter of the two and q (French queue) for the longer. Within each arm, staining produces characteristic bands, which

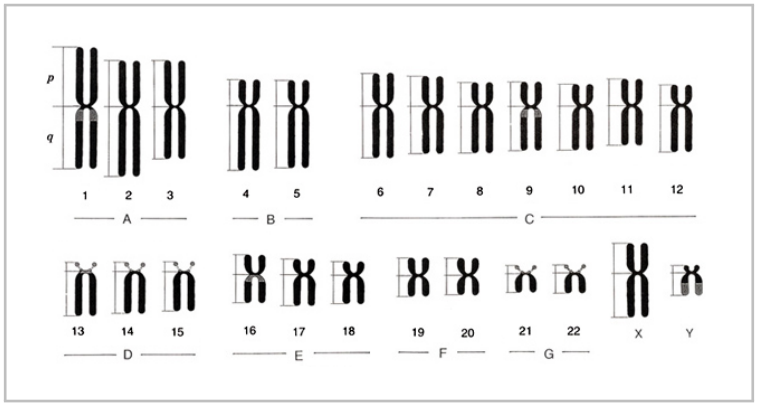


Figure 2.2: *The rule of karyotyping is to arrange 22 autosomes following the size and sex chromosomes, X and Y, at the end. Chromosomes are classified into seven groups, A to G, by the length and centromere position.* Karyogram of human chromosomes, adapted from [10].

are grouped into regions and numbered, counting outward from the centromere [9]. For example, the band referred to as 6p14 is the fourth band of the first region of the short arm of chromosome 6. The position of the centromere is very useful, since the rule of karyotyping is to classify all chromosomes by the length and centromere position into seven groups [10], from A to G (see Figure 2.2).

2.1.3. DNA

DNA serves as the primary and permanent storage of information in most organisms [12]. DNA is composed of a long chain of **nucleotides** (or *bases*), grouped in segments called **genes** that are able to encode sequences of amino acids, the building blocks of **proteins**.

The DNA nucleotides are composed of a ***phosphate group***, a ***deoxyribose sugar***, and one of four ***nitrogen bases***: adenine (**A**), guanine (**G**), cytosine (**C**), and thymine (**T**) [13]. Using biochemical nomenclature, the sugars are joined by phosphate groups that form phosphodiester bonds between the *third* and *fifth* carbon atoms of adjacent sugar ring [14]. These asymmetric bonds mean a strand of DNA has a direction. So, one direction of a single strand is called the 5' direction, and the other is the 3' direction since such direction in one strand is opposite to their direction in the other strand, the strands are *antiparallel* [12].

In a DNA double helix, each type of *nucleobase* on one strand bonds with just one type of *nucleobase* on the other strand. This is called complementary base pairing [15]. Here, purines form hydrogen bonds to pyrimidines (i.e. **A** bonds only to **T**, and **C** bonds only to **G**). This arrangement of two nucleotides binding together across the double helix is called a ***base pair*** [16]. As the sequence of one strand can be easily inferred from that of the other strand, sequences are usually specified by writing only a single strand (by convention, in the 5' to 3' direction). Therefore, the two DNA strands are ***complementary***. Figure 2.3 depicts an example of one strand (5'-TGACAGTCAGT-3') and its complementary (3'-ACTGTCAGTCA-5') one.

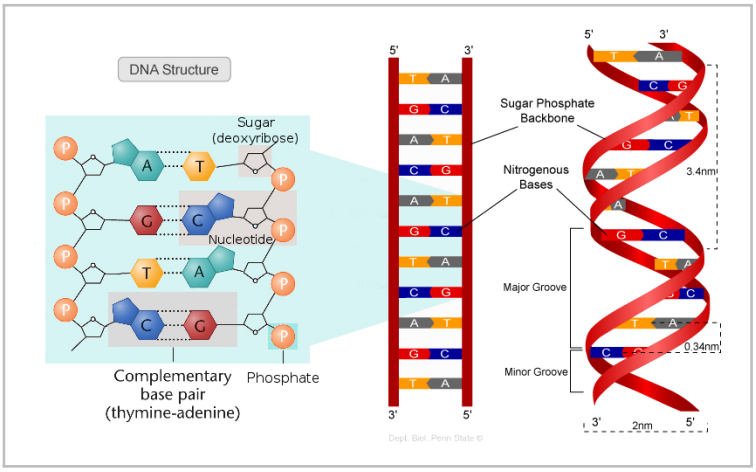


Figure 2.3: Structural diagram of DNA, adapted from [17].

It is also important to note that **human genome** is made up of 3×10^9 base-pairs, distributed over the 23 pairs of chromosomes, each chromosome being on average, 125 million base-pairs in length [9], and containing around 20000-25000 *protein-coding* genes [18].

2.1.4. Gene

A **gene** may be defined as a linear sequence of nucleotides along a segment of DNA that provides the coded instructions for synthesis of *Ribonucleic Acid* (RNA), which, when translated into protein, leads to the expression of hereditary character [19].

In the human genome, large stretches of DNA are transcribed but do not code for proteins. These regions are called introns (see right side of Figure 2.1) and make up to around 95 percent of the genome [20]. The nucleotide sequence of the human genome is now known to a reasonable degree of accuracy but the large amount of non-coding DNA is not. Some of this non-coding DNA controls gene expression, but the purpose of much of it is not yet understood. The process by which DNA is copied to RNA is called **transcription** and that by which RNA is used to produce proteins is called **translation** [21].

2.1.5. Genotype vs. Phenotype

An individual's **genotype** is the totality of that person's hereditary material, whereas an individual's phenotype is his/her appearance [22], i.e. their observable characteristics or traits. Not all organisms with the same genotype look or act the same way because appearance and behavior are modified by environmental and developmental conditions. Likewise, not all organisms that look alike necessarily have the same genotype [23].

A **phenotype** results from the expression of an organism's genes as well as the influence of environmental factors and the interactions between the two. When two or more clearly different phenotypes exist in the same population of a species, the species is called **polymorph** [24].

However, the terms **genotype** and **phenotype** are usually used for a particular locus or set of loci, and for a particular trait or set of traits [22].

2.1.6. Locus

A **locus** *(plural loci)* is the specific location of a gene, DNA sequence, or position on a chromosome [25]. The locus content is defined by its base sequence. The different base sequences that may be at the same locus are called ***variants***.

2.1.6.1. Marker

A particular genetic **marker** is a DNA locus located in the DNA, in which there is, at least, one base-pair difference between at least two individuals. A locus could be used as a marker, only if the locus can be detected and it has a known location in the genome.

The usefulness of a marker lies in how informative it can be. Its level of information depends on the number of heterozygous subjects in the population.

2.1.7. Allele

An **allele** is one of a number of alternative forms of the same gene or same genetic locus [26]. The information within a particular gene is not always the same between one organism and another. Different alleles can result in different observable phenotypic traits, such as different pigmentation. However, most genetic variations result in little or no detectable variation [27]. Per example, if the eye color were characterized in the human genome by only one locus and two possible colors: brown and blue, a pair of alleles would be necessary to determine the eye colour (see Figure 2.4).

2.1.7.1. Heterozygosity and homozygosity

In many cases, genotypic interactions between the two alleles at a locus can be described as ***dominant*** or ***recessive***, according to which of the two homozygous phenotypes the heterozygote most resembles. Where the **heterozygote** is indistinguishable from one of the **homozygotes**, the allele involved is said to be dominant to the other, which is said to be recessive to the former [28]. The degree and pattern of dominance vary among loci. This type of interaction was first formally described by Gregor Mendel. However, many traits defy this simple categorization, and it is necessary to characterize the phenotypes by using more complex inheritance models [29].

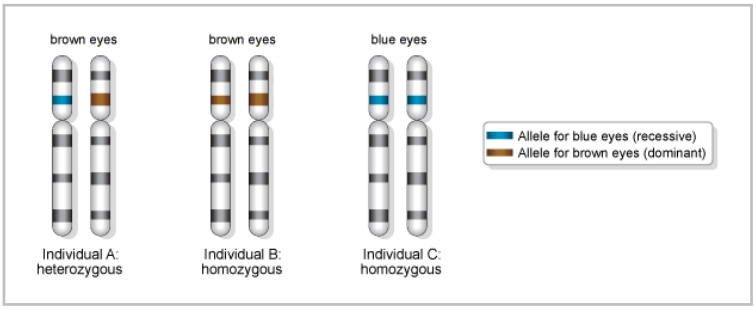


Figure 2.4: Chromosomes show matching genes with alleles displayed in different colours, adapted from [30].

The term ***wild type*** allele was historically regarded as dominant, common and normal, in contrast to *mutant* alleles regarded as recessive, rare and frequently deleterious. It was formerly thought that most individuals were homozygous for the *wild type* allele at most gene loci, and that any alternative *mutant* allele was found in homozygous form in a small minority of *affected* individuals, often as genetic diseases, and more frequently in heterozygous form in *carriers* for the mutant allele. It is now noted that most or all gene loci are highly polymorphic, with multiple alleles, whose frequencies vary from population to population, and that many genetic variations are hidden in the form of alleles that do not produce obvious phenotypic differences [29].

2.1.8. Haploid genotype: Haplotype

A **haplotype** is a group of genes within an organism that was inherited together from a single parent. This term is derived from *haploid* (which describes cells with only one set of chromosomes) and from *genotype* (which refers to the genetic composition of an organism). This group of genes is often inherited together because they are physically close to each other on the same paternal chromosome (*genetic linkage*). Furthermore, the term *haplotype* can also refer to the inheritance of a group of variations at single positions in a DNA sequence among individuals [31].

By examining haplotypes, those patterns of genetic variation which are associated with health and disease states *(in **case-control** studies)* can be identified. For instance, if a haplotype is associated with a certain disease, then the associated cluster of DNA sequences can be examined to identify the related gene/s that may be causing such disease [31].

2.2. Transmission of Genetic Information

The first step in the transmission of genetic information (from parent to offspring) in human cells is called **meiosis**. Meiosis is one form of cell reproduction in which a diploid (2n) cell undergoes two successive divisions, thus generating four haploid cells (n). Namely, from a diploid cell *(containing 23 pairs of chromosomes)*, four haploid cells are built *(each one containing 23 chromosomes)*. The process is divided into two stages: meiosis I and meiosis II (see Figure 2.5).

Meiosis I results in two daughter cells, each having 23 duplicated chromosomes. In meiosis II the two ***chromatids*** of each chromosome

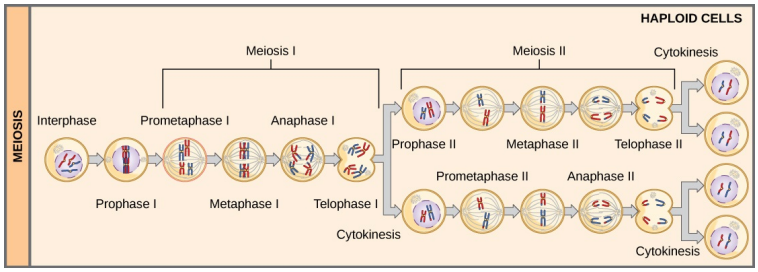


Figure 2.5: *Meiosis includes two nuclear divisions. The four daughter cells resulting from meiosis are haploid and genetically distinct. The daughter cells resulting from mitosis are diploid and identical to the parent cell.* Meiosis two stage scheme, adapted from [32].

are separated, and the cell division results in two haploid cells. If this were the full description of meiosis, each of the 22 autosomes in a gamete would be an exact copy of one of the two parental homologous chromatids. In fact, a process called **meiotic recombination** mixes the genetic material of the homologous chromatids during meiosis, so that each chromosome present in the gamete has contributions from both parents [32].

2.2.1. Meiotic recombination

In meiosis I, homologous chromatids pair up and form physical connections called ***chiasmata*** (singular, *chiasma*). Chiasmata are essential for correct chromosome alignment and segregation and thus are thought to perform a role in meiosis I. Each chromosome arm normally forms at least one chiasma [9].

Meiotic recombination or crossing-over occurs at the chiasmata. Specific enzymes break the DNA strands and repair the break in a way that swaps material from one chromosome with material from another (see Figure 2.6). The most important consequence of meiotic recombination is that gametes receive contributions from both homologs of a chromosome pair (thus from both grandparents) [9]. Two alleles that are linked (on the same chromosome) in the parent may or may not be linked in the offspring. As a result, a single person could theoretically produce an almost infinite number of genetically different gametes.

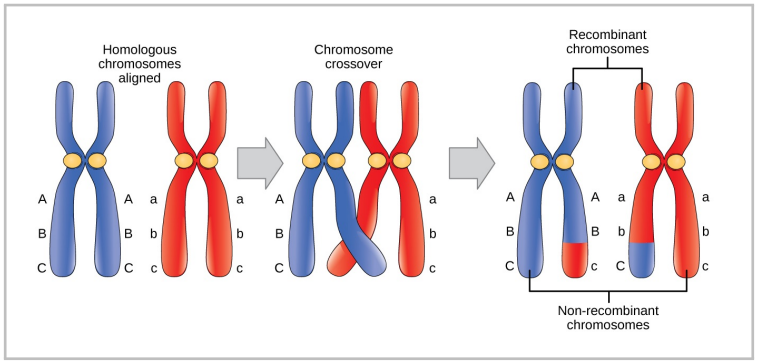


Figure 2.6: *Effects of crossing over: the blue chromosome came from the individual's father and the red chromosome came from the individual's mother.* Meiotic recombination, adapted from [32].

2.3. Human Genetic Diversity

Human genetic diversity is considerably lower than other species, including our nearest evolutionary relative, the chimpanzee. Genetic diversity is a function of a population's "age", i.e. the amount of time during which mutations accumulate to generate diversity and its size. Our genetic homogeneity implies that anatomically modern humans arose relatively recently *(200000 years ago)* and that our population size was quite small at one time *(10000 breeding individuals)* [33].

According to *Baker* [34], the human genetic diversity can be estimated as 0.1-0.5%. Taking into account that humans have approximately 3 billion (3×10^9) base pairs in a haploid cell, it can be said that any pair of humans differs by approximately 3 to 15 million base pairs. These differences contain much useful information about the evolutionary history of our species [33].

Genetic variations mainly include **mutations** and **polymorphisms**, described in subsection 2.3.1. These DNA variations can be single base pair changes, deletions, insertions, inversions, translocations, changes in the number of copies of a given DNA sequence, or even duplications of whole chromosome sections, see Figure 2.7.

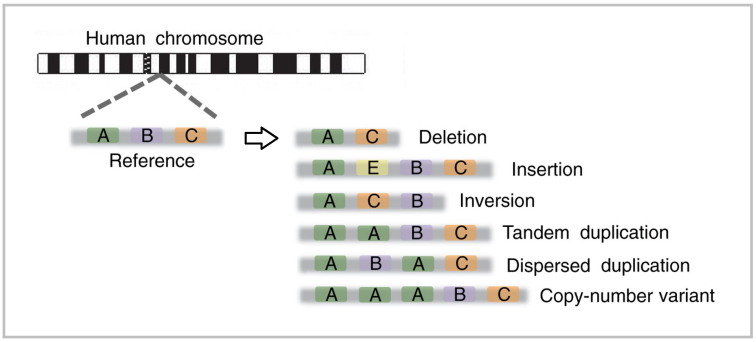


Figure 2.7: *Genome structural variation encompasses polymorphic rearrangements 50 base pairs to hundreds of kilobases in size and affects about 0.5 percent of the genome of a given individual.* Genetic Variations, adapted from [34].

2.3.1. Polymorphism vs. Mutation

DNA sequence variations are sometimes described as mutations and sometimes as polymorphisms. It is important to clarify what is the difference between these terms and how they are applied to the human genome.

The term **polymorphism** (a term that comes from the Greek words *poly*: "many" and *morphe*: "form") is generally restricted to those variations that are relatively common (present in more than 1 percent of individuals) and that usually do not have highly deleterious consequences. Highly deleterious rare variants are often referred to as mutations (present in less than 1 percent of the population) [35]. Therefore, to be classified as a polymorphism the least common allele must have a frequency of 1% or more in the population. If the frequency is lower than this, the allele is regarded as a **mutation**.

The above definitions cannot be applied rigorously. Thus, within a population, a variant can be characterized by its lower allelic frequency (minor allele frequency) which it is simply the lowest frequency corresponding to the two alleles of the at a given locus. Given the variations among human populations, the least frequent allele of a certain locus within a population may be the most prevalent in another, namely a mutation in one population can become a polymorphism in another if it confers an advantage and increases in frequency. A good example is the allele of sickle-cell disease. In Caucasian populations, this is a rare

sequence variant of the beta-globin gene that causes a severely debilitating blood disorder. In certain parts of Africa, however, the same allele is polymorphic because it confers resistance to the blood-borne parasite that causes malaria [36].

2.3.2. SNP: Single Nucleotide Polymorphism

The most common type of variation in the human genome is the *Simple Nucleotide Polymorphism* (SNP), which is a change in one base pair at a particular location of the genome. SNPs can be divided into two types, depending on the base substitution [37]:

- ***Transitions***, the substitution of one **purine** for another (A ↔ G) or one **pyrimidine** for another (C ↔ T), are the most common type of SNP.
- ***Transversions***, in which a purine is replaced by a pyrimidine, or vice versa, are less common.

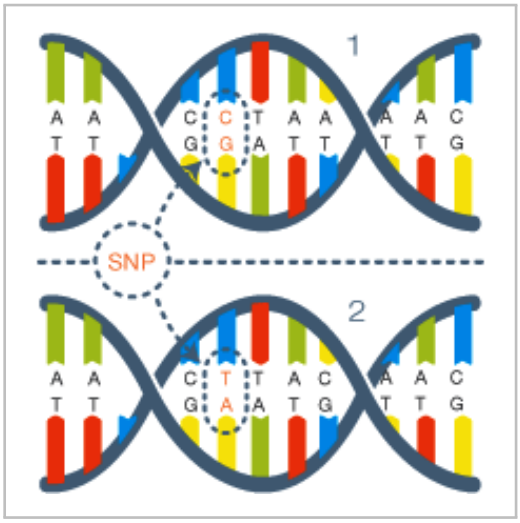


Figure 2.8: *DNA molecule 1 differs from DNA molecule 2 at the same location.* SNP representation, adapted from [38].

SNPs are found more frequently in those DNA regions containing genes and may serve as biomarkers to identify what genes are associated with specific diseases. When SNPs occur within the gene, or in a regulatory region near a gene, they can significantly affect gene function [39].

The frequency of a particular SNP tends to remain stable in the population. Unlike the other, rarer kinds of variations, many SNPs occur in genes and in the surrounding regions of the genome that control their expression. The effect of a single polymorphism in a gene may not be large –*perhaps influencing the activity of the encoded protein in a subtle way*– but even subtle effects can influence susceptibility to common diseases [36].

2.3.3. Highly Polymorphic Regions: MHC - HLA Complex

Certain DNA regions of the human genome have a high polymorphism rate. From all these stands the MHC that contains the most diverse genes known in vertebrates. These highly polymorphic genes encode cell surface receptors that play a central role in distinguishing self/own from foreign proteins. The polymorphisms of MHC genes has been maintained by natural selection over long periods of evolutionary time [40].

The MHC is also known as HLA in humans [41]. This region encompasses 7.6×10^6 bases on chromosome 6p21 and is the most gene dense region within the human genome encoding 252 loci [42] including several key immune response genes [43]. The region can be subdivided into Class I, Class II and Class III regions (see Figure 2.9).

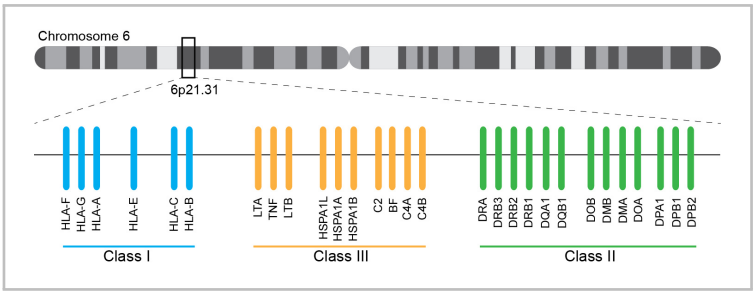


Figure 2.9: *MHC complex (HLA region) in Human Chromosome 6.* MHC-HLA Complex [44].

The enormous polymorphism of MHC alleles is difficult to explain because natural selection should eliminate all but the most disease- protective allele. For example, a particular human MHC allele confers resistance to malaria in Africa. This process of directional selection should lead to all of the susceptible alleles becoming extinct, leaving only the most resistant allele in the population (fixation). Therefore, there must be some other evolutionary pressure that maintains MHC diversity [40].

Specific HLA alleles are associated with susceptibility and resistance to autoimmune and infectious diseases. The disparity between donor and recipient HLA-A, B, C, DR, DQA, DQB and DPA and DPB alleles impacts the outcome of both bone marrow and solid organ transplantation [44].

2.3.3.1. HLA Allele nomenclature

Early in their study, it was recognized that the genes encoding the HLA molecules were highly polymorphic and that there was a need for a systematic nomenclature. The HLA complex contains more than 220 genes of diverse function. Many of the genes encode proteins of the immune system [45]. The naming of new HLA genes and allele sequences and their quality control is the responsibility of the WHO Nomenclature Committee for Factors of the HLA System.

In 2010, a new HLA nomenclature system was adopted. The main drive for the change was that the old system could no longer accommodate the increasing number of HLA alleles that were being described, due to the fact that HLA complex is the most polymorphic region of the entire human genome with close to 9000 different HLA alleles characterized thus far [44]. The list continues to expand rapidly as increasing numbers of new alleles continue to be identified. The list containing the most updated alleles is in IMGT/HLA database.

As is depicted in Figure 2.10, have been adopted colons ':' as separators between pairs of digits. HLA-A*02010102L therefore became HLA-A*02:01:01:02L. The pairs of digits separated by colons are known as Fields. The first and second digits of the old nomenclature form the 1^{st} Field of the new nomenclature. The third and fourth digits of the old nomenclature form the 2^{nd} Field of the new nomenclature. To help reduce confusion in adopting the new nomenclature, the leading '0' in alleles 1-9 of each allele group was kept [46].

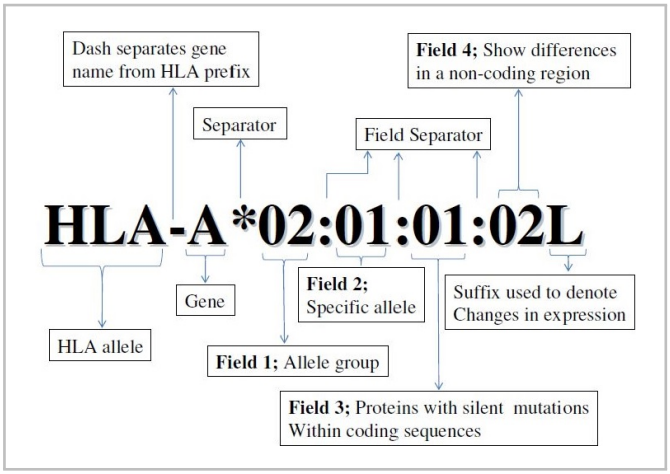


Figure 2.10: *Each HLA allele name has a unique number corresponding to up to four sets of digits separated by colons. The length of the allele designation is dependent on the sequence of the allele and that of its nearest relative.* HLA Alleles Nomenclature [46].

Chapter 3

Concepts of Statistical Genetics

3.1. Statistical Genetics

Statistical genetics has experienced exponential growth during last years, with an increasing number of people involved in this area. It can be affirmed that principles of statistical genetics synthesize previous statistical models and methods applied to genome study. Statistical methodologies behind certain computational software packages may help researchers to choose the optimal tool to analyze their data and also interpret their studies better [47].

Primarily, contemporary statistical genetics focuses on the development and implementation of data analysis methodologies that may facilitate the identification of genes and genetic variations that influence phenotypic expression and disease susceptibility [48]. As it is possible that a single SNP or a set of a few SNPs contribute significantly to a disease, the fact that there are millions of SNPs makes it particularly difficult to detect their real causal association with a specific disease. Furthermore, most conditions for current public health research, are complex and multifactorial, since they usually are composed of many genes and environmental factors, as well as their corresponding expression or interactions [49].

There are two general designs in *genetic association studies*: ***family-based*** designs that use pedigrees and ***population-based*** studies that employ unrelated individuals. The recruitment of unrelated individuals is easier than the recruitment of families, but they are subject to bias in the presence of *population stratification* (individuals with a significantly

different genetic ancestry and phenotype from the rest of the study). As a compromise between *linkage studies* (which usually involve large families where the disease affects individuals in several generations) and population-based association studies, family-based association designs can have similar power as population-based designs and are more robust in the presence of population stratification [50].

3.2. Population Studies

Throughout the history of population genetics, statistical models have played a significant role in explaining the effects of genetic diversity of organisms. Such models today are crucial in the development of statistical tools for analyzing molecular biology data.

Regardless of assumptions about the genetic model of a trait, or technologies used to assess genetic variation, no genetic study will have meaningful results without a thoughtful approach to characterize the phenotype of interest. When embarking on a genetic study, the initial focus should be on identifying precisely what genetic variation influences [51].

The following subsections will provide a brief introduction to some concepts of population genetics.

3.2.1. Case-Control and Quantitative Designs

There are two primary classes of phenotypes: categorical (often binary case/control) or quantitative. From the statistical point of view, quantitative traits are preferred because they improve power to detect a genetic effect, and often have a more interpretable outcome [51].

An example of quantitative trait is cholesterol levels, which are strong predictors of heart disease. The analysis of such levels is very useful for clinical practice since they are precise and ubiquitous measurements that are easy to obtain [51]. Genetic variants that influence these levels have a clear interpretation (e.g. a unitary level change per allele), having, therefore, an easily measurable effect on the quantitative trait.

Other disease traits do not have well-established quantitative measures. In these circumstances, individuals are usually classified as either affected or unaffected (case or control), i.e. a binary categorical variable [51]. Although quantitative outcomes are preferred, they are not always required for a successful study.

3.2.2. Population Association Analysis

In recent years, GWAS have become the most used tool for the identification of loci associated with complex traits. Through this method, association between a trait of interest and genetic polymorphisms is studied using individuals' samples typed for millions of SNPs [52], allowing for the discovering of numerous statistical associations between genomic variants and quantitative traits or complex diseases.

When genotypes are collected, and a well-defined phenotype has been selected for a population study, the statistical analysis of genetic data can begin [51]. Such analysis can be performed either with series of single-locus tests (examining each SNP independently for association with phenotype) or by multi-loci tests (considering interactions among different genetic variants throughout the genome).

3.2.2.1. Single-locus Analysis

For both quantitative and dichotomous trait analysis, there are numerous ways to handle genotype data for association tests. The choice of the model can have implications for the statistical power of the test, as the degrees of freedom may change depending on genotype classes (i.e.: homozygous/heterozygous, dominant/recessive, etc.). Allelic association tests, therefore, examine the association between one allele of the SNP and a single or multiple phenotypes.

Statistical analysis of possible genotype independence at a ***single locus*** is simple because the absolute frequencies observed in both of them will be presented in a double entry table. In such a table, independence will be studied by an independent χ^2 test (see next paragraph) using Yates correction for continuity (or Yates' χ^2 test) [53] if the expected cell frequencies are less than 5.

3.2.2.1.1. Hardy-Weinberg Equilibrium

Hardy-Weinberg Equilibrium **(HWE)** (also known as Hardy-Weinberg principle) states that allele and genotype frequencies in a population will remain constant from generation to generation in the absence of other evolutionary influences [54]. These influences include mate choice, mutation, selection, genetic drift, gene flow and meiotic drive. HWE denotes independence of alleles at a single site between two homologous chromosomes [55].

Under HWE assumptions, genotype frequencies can be estimated from allele frequencies. When the ratios of homozygous and heterozygous genotypes significantly differ from the prediction (under such assumptions), this can indicate genotyping errors, batch effects, population stratification, non-random mating or inbreeding [56]. The departure from HWE is an indicator that a marker should be discarded from the study since it would not be useful to test the association.

A simple example for visualizing the allele proportions of a SNP (with genotypes AA, Aa, and aa) under HWE is shown in Figure 3.1.

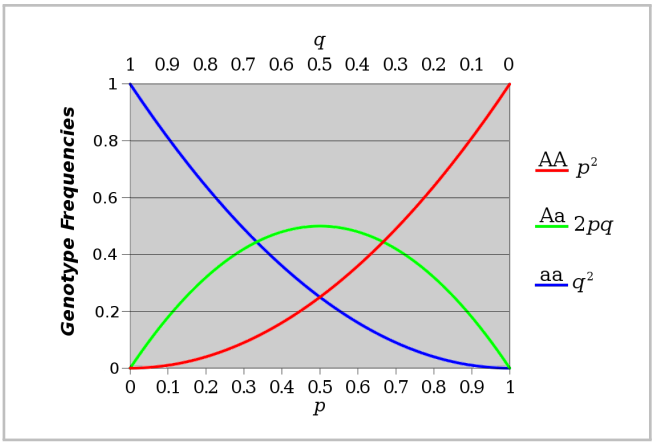


Figure 3.1: *The horizontal axis shows the two allele frequencies p and q and the vertical axis shows the expected genotype frequencies. Each line shows one of the three possible genotypes.* Hardy-Weinberg proportions for two alleles, adapted from [57].

HWE implies that the probability of an allele occurring on one homolog chromosome does not depend on which allele is present on the second homolog. Formally, independence in this setting is equivalent to stating that the joint probability of A and a, given by p_{Aa}, is equal to the product of the individual allele probabilities, p_A and p_a, and also that $p_{AA} = p_A^2$, $p_{Aa} = p_A p_a$ and $p_{aa} = p_a^2$, where pA and $p_a = 1 - p_A$ are the population frequencies of alleles A and a, respectively [55]. The departure from HWE can be tested by **Pearson's χ^2 test** or Fisher's exact test.

The data for checking HWE at a single locus are usually arranged as shown in Table 3.1, where n_{11} and n_{22} are the numbers of individ-

uals with genotypes AA and aa (and these counts are observed). The genotypes Aa and aA will be indistinguishable, and thus, we only can see the sum $n_{12}^* = n_{21} + n_{12}$ and not the individual locus counts (n_{21} and n_{12}) [55].

	Homolog 2		
Homolog 1	**A**	**a**	***Total***
A	n_{11}	n_{12}	$N_{1\cdot}$
a	n_{21}	n_{22}	$N_{2\cdot}$
Total	$N_{\cdot 1}$	$N_{\cdot 2}$	N

Table 3.1: *Genotype counts for two homologous chromosomes.* Genotype counting, adapted from [55].

The expected counts corresponding to these three observed numbers: n_{11}, n_{12}^*, n_{22} are given respectively by $E_{11} = Np_A^2$, $E_{12} = 2Np_A(1\text{-}p_A)$ and $E_{22} = N(1\text{-}p_A)^2$, where p_A is the probability of A and is estimated based on the observed allele count and $N = n_{11} + 2n_{12} + n_{22}$. That is, we let $p_A = (2n_{11} + n_{12}^*)/(2N)$. The χ^2 test statistic is then constructed as follows:

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{(n_{ij} - E_{ij})^2}{E_{ij}} \sim \chi_1^2 \tag{3.1}$$

This statistic is compared with the appropriate quantile of a χ_1^2 distribution to determine whether to reject the null hypothesis of HWE [55]. Knowing the value of any of the n_{ij} observed frequencies determines the counts on the marginal totals.

3.2.2.1.2. Logistic regression in Case-Control Studies

Logistic regression tries to predict the *probability* ***p*** of being a case by the expression:

$$logit(p) = \beta_0 + \sum_{i=1}^{K} \beta_i x_i \tag{3.2}$$

where x_i are explanatory variables and the ***logistic function*** of p is given by:

$$logit(p) = log\frac{p}{1-p} \tag{3.3}$$

From the above, we can easily deduce that:

$$p = \frac{e^{\beta_0+\sum_{i=1}^{K}\beta_i x_i}}{1+e^{\beta_0+\sum_{i=1}^{K}\beta_i x_i}} \tag{3.4}$$

The ***logistic function*** is useful because it can take input with any value from negative to positive infinity, whereas the output always takes values between zero and one ($0 \leq p \leq 1$) and hence it can be easily interpretable as a probability [58].

It is important to note that p is usually interpreted as the probability of the dependent variable be equals to a *success* or *case*, rather than a *failure* or *non-case* [59].

Logistic regression is often the preferred approach in case-control studies having the corresponding coefficients a simple interpretation in terms of ***Odds Ratio*** **(OR)**.

3.2.2.1.3. Frequentist Model in Case-Control Studies

Considering a case-control study (i.e. a binary phenotype), the state of an individual i can be described by the dichotomic variable Φ_i defined as 0 or 1 according to the absence/presence of disease. The probability p_{ij} of an individual i being a case, given its genotype at the j^{th} SNP, can be modelled by a *logistic regression* with parameter $\theta = (\beta_0, \beta_1)$ as follows:

$$p_{ij} = P\left(\Phi_i = 1 \,|G_{ij}, \theta\right) = \frac{e^{\beta_0+\beta_1 G_{i,j}}}{1+e^{\beta_0+\beta_1 G_{i,j}}} \tag{3.5}$$

or equivalently:

$$\log\frac{p_{ij}}{1-p_{ij}} = \beta_0 + \beta_1 G_{ij} \tag{3.6}$$

Obviously, the probability of not having the disease would be:

$$P\left(\Phi_i = 0 \,|G_{ij}, \theta\right) = 1 - p_{ij} = \frac{1}{1+e^{\beta_0+\beta_1 G_{i,j}}} \tag{3.7}$$

The variable G_{ij} can be defined in different manners depending on the inheritance mode. If we denote the *main allele* as "**A**" and the

alternate allele as "**a**", we can consider the following models for the association between genotype and disease:

- **Additive model**: Each copy of the alternate allele a modifies the risk of disease in an additive form, i.e. the homozygous aa have double risk than heterozygous Aa. Thus, the variable G_{ij} may be defined as:

$$G_{ij} = \begin{cases} 0 & \text{if subject } i \text{ has an } AA \text{ genotype} \\ 1 & \text{if subject } i \text{ has an } Aa \text{ genotype} \\ 2 & \text{if subject } i \text{ has a } aa \text{ genotype} \end{cases} \tag{3.8}$$

- **Dominant model**: A single copy of the alternate allele a is enough to change the risk. So heterozygous Aa and homozygous aa genotypes have the same risk. In this case, the variable G_{ij} may be defined as:

$$G_{ij} = \begin{cases} 1 & \text{if subject } i \text{ has an } Aa \text{ or } aa \text{ genotype} \\ 0 & \text{if subject } i \text{ has an } AA \text{ genotype} \end{cases} \tag{3.9}$$

- **Recessive model**: two copies of the alternate allele a are necessary to modify the risk of disease. Hence, Aa and AA genotypes have the same effect. The variable G_{ij} may be defined as:

$$G_{ij} = \begin{cases} 1 & \text{if subject } i \text{ has a } aa \text{ genotype} \\ 0 & \text{if subject } i \text{ has an } AA \text{ or } Aa \text{ genotype} \end{cases} \tag{3.10}$$

- **Heterozygous model** (a.k.a. Overdominant model): the risk of disease changes when the genotype is heterozygous. In this case, the variable G_{ij} may be defined as:

$$G_{ij} = \begin{cases} 1 & \text{if subject } i \text{ has an } Aa \text{ genotype} \\ 0 & \text{if subject } i \text{ has an } AA \text{ or } aa \text{ genotype} \end{cases} \tag{3.11}$$

- **General model** (a.k.a. Codominant model): This model allows every genotype to give a different and non-additive risk. It compares heterozygous genotype A/B and homozygous genotypes for the alternate allele aa and most frequent allele AA. We need to

consider two variables, G_{ij} (the same as in additive model) and T_{ij} (the same as in heterozygous model). Now, the logistic model has the form:

$$p_{ij} = \frac{e^{\beta_0+\beta_1 G_{ij}+\beta_2 T_{ij}}}{1+e^{\beta_0+\beta_1 G_{ij}+\beta_2 T_{ij}}} \tag{3.12}$$

The odds ratios of disease for individuals with genotypes 1 and 2 (relative to individuals with the 0 genotype) are e^{β_0} and $e^{2\beta_0}$, respectively. This model is multiplicative on the odds scale and additive on the log-odds scale.

If we consider that the genotypes for a given SNP can be grouped into other genotype classes or models such as *dominant*, *recessive* or *heterozygote* models plus the *general* two-parameter model, the statistical association can be dealt with a similar way.

3.2.2.2. Multi-loci Analysis

In addition to single-locus analysis, ***Multi-loci analysis*** also presents several advantages, although they are not as straightforward as single-locus tests.

Selecting SNPs to analyze based on main effects will prevent certain multi-loci models from being detected with statistically undetectable marginal effects. With these models, a large component of the heritability is concentrated on the interaction rather than in the main effects. In other words, a specific combination of markers incurs a significant change in disease risk. The benefits of this approach are that it performs an unbiased analysis of interactions within the selected set of SNPs, being more computationally and statistically tractable than analyzing all possible combinations of markers [51].

3.2.2.2.1. Linkage Disequilibrium

Linkage Disequilibrium (LD) is related to the concept of chromosomal linkage, where two markers on a chromosome remain physically joined on a chromosome through generations of a family. When a specific SNPs of a population are in LD, they have to be placed contiguously in the same genome sequence or stretch. The rate of LD decay is dependent on multiple factors, including the population size, the number of founding chromosomes in the population, and the number of generations for which the population has existed [51].

African-descent populations are the most ancestral and have smaller regions of LD due to the accumulation of more recombination events in that group. European-descent and Asian-descent populations were created by founder events, which altered the previously cited factors. These populations on average have larger regions of LD than African-descent groups. LD may, therefore, describe the degree to which one SNP is inherited or correlated with another SNP within a population, i.e. different human sub-populations may have different degrees and patterns of LD [51].

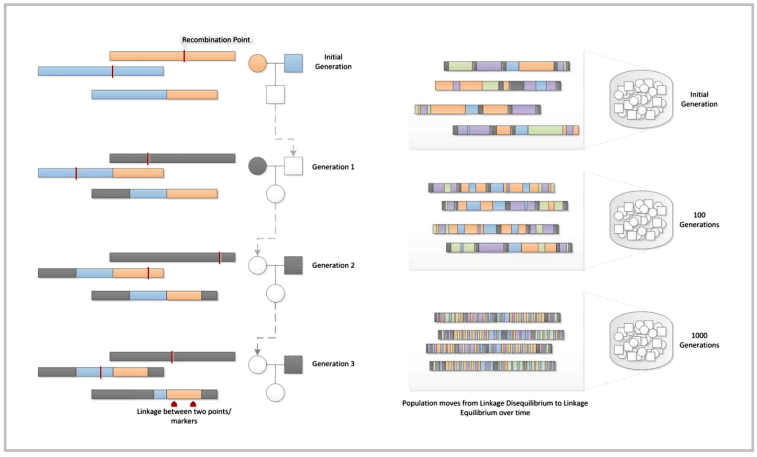


Figure 3.2: *Decay of Linkage over successive generations within a family (right) and a population (left).* Linkage and Linkage Disequilibrium, adapted from [51].

In Figure 3.2, two founder chromosomes are shown (one in blue and one in orange). Recombination events within a family from generation to generation break apart chromosomal segments. This effect is amplified through generations, and in a population of fixed size undergoing random mating, repeated random recombination events will break apart contiguous chromosome segments (containing linked alleles) until eventually all alleles in the population are in linkage equilibrium or are independent. Thus, the linkage between markers on a population scale is referred to as LD [51].

3.2.2.2.1.1. Measures of Linkage Disequilibrium

To explain the principal ***measures of LD***, firstly we have to consider a distribution of alleles for n individuals across two loci. Then, assuming that the two loci are independent of each other, i.e. they are in *linkage equilibrium*, the presence of an allele at one locus should not influence the particular allele observed at the second locus.

Supposing A and a as possible alleles at *Locus 1*, and B and b as possible alleles at $Locus2$, the marginal probabilities of the alleles A, a, B, b will be p_A, p_a, p_B, and p_b, respectively. Since each individual carries two homologous chromosomes, there will be a total of $N = 2n$ homologs across the n subjects in a population.

	Locus 2	
Locus 1	**A**	**a**
A	$N(p_A p_B + D)$	$N(p_A p_b - D)$
a	$N(p_a p_B - D)$	$N(p_a p_b + D)$

Table 3.2: Observed allele distributions under LD, adapted from [55].

On the other hand, if the two loci are associated, the expected values will have be deviated a quantity. Such deviation is commonly symbolized by the scalar D and its amount is represented in Table 3.2.

We can express D regarding the joint probability of A and B, and the product of the individual allele probabilities as follows:

$$D = p_{AB} - p_A p_B \tag{3.13}$$

the value that estimates the *Disequilibrium* being the scalar D':

$$D' = \frac{|D|}{D_{max}} \tag{3.14}$$

where D_{max} represents the upper bound on D and is given by:

$$D_{max} = \begin{cases} min(p_A p_b, p_a p_B) & if D > 0 \\ min(p_A p_B, p_a p_b) & if D < 0 \end{cases} \tag{3.15}$$

Note that D' will be a value with range: $0 \leq D' \leq 1$, so that D' values close to 0 will suggest *linkage equilibrium* (almost no association),

while values close to 1 will indicate high levels of LD and depending on the study, a possible genetic association with a disease.

Another measure that is also a function of the scalar D is the quantity r^2. This measure is based on Pearson's χ^2 test of no association between the rows and columns of table 3.2. Specifically, r^2 can be defined as:

$$r^2 = \frac{\chi_1^2}{N} \tag{3.16}$$

Where, if r^2 is written in terms of the scalar D, results:

$$r^2 = \frac{D^2}{p_A p_B p_a p_b} \tag{3.17}$$

The difference between D and r^2 rests in the type of adjustment made to the scalar D. In both cases, this adjustment involves the marginal allele frequencies since the value of D will depend on these. Investigators commonly use r^2 [55], due to its simpler relationship to the usual Pearson's χ^2 test.

3.2.2.2.2. Indirect Association

The presence of LD creates two possible positive outcomes from a genetic association study. In the first one, the SNP influencing a biological system that ultimately leads to the phenotype is directly genotyped in the study and found to be statistically associated with the trait. The above is referred to as a direct association, and the genotyped SNP is sometimes referred to as the functional SNP. The second possibility is that the influential SNP is not directly typed, but instead, a tag SNP in high LD with the influential SNP is typed and statistically associated with the phenotype (see Figure 3.3). This is referred to as an indirect association [60].

Because of these two possibilities, a significant SNP association from a GWAS should not be assumed as the causal variant and may require additional studies to map the precise location of the influential SNP [51].

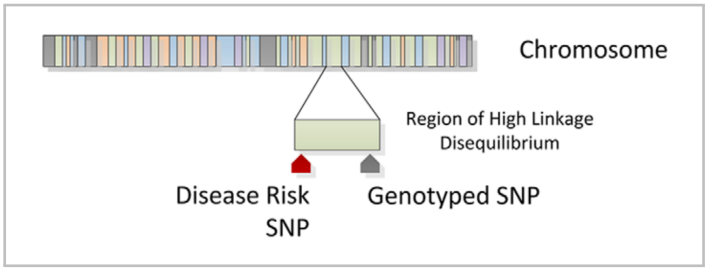


Figure 3.3: *Genotyped SNPs often lie in a region of high linkage disequilibrium with an influential allele. The genotyped SNP will be statistically associated with the disease as a surrogate for the disease SNP through an indirect association.* Indirect Association [51].

3.3. Family-based Studies

As was mentioned above, ***Family-based association studies*** have several advantages in comparison to *population-based association studies*.

Firstly, the need for match cases to controls in population studies may lead to selection bias and confounding effects if gene frequencies differ between case and control populations. By choosing controls from the same families as the cases, the confounding effects will be substantially reduced [61].

Furthermore, as a family-based association can only be detected when the linkage is present, the identification of such association confirms, despite some difficulties in interpreting results, that truly associated markers are physically close to the causal genetic variant, supporting the fact that phenotypes have been inherited [61].

On the other hand, family-based studies have some disadvantages that mainly arise from practical matters of recruitment and cost [62]. This is the main reason why population controls are commonly used for large-scale genome association studies.

3.3.1. Family-based Association Tests

Family-Based Association Test **(FBAT)** include several ways for detecting associations between specific markers and quantitative phenotypes or diseases. In these studies, nuclear families consisting at least of two parents and a number of full siblings are widely used, but extended pedigrees may also be used for testing association [61], often improv-

ing expected results. Such association methods can be broadly classified into two groups: ***nonparametric methods*** (based on allele counting) and ***parametric methods*** (based on the likelihood function).

For the simplest family-based association design (two parents and one affected offspring) previous methods result in similar test statistics, i.e. their extensions on more complex situations vary considerably [50].

Typical statistical tests use general pedigrees and may detect genotype-environment interactions. The subsections that follow briefly describe some of the most used ones.

3.3.1.1. Transmission/Disequilibrium Test

The simplest family-based design for association studies is the case–parent design, in which an affected offspring and both parents are genotyped at bi-allelic markers. A method called **TDT** for evaluating this was proposed by *Spielman and Ewens* [63]. In this, the alleles transmitted from parents to the affected offspring and the alleles of not transmitted can be determined based on the observed genotype data [50]. Thus, a two by two transmission/nontransmission table for a bi-allelic marker with alleles ***A*** and ***B*** from ***t*** case–parent trios can be constructed:

	Non-transmitted		
Transmitted	**A**	**B**	***Total***
A	n_{AA}	n_{AB}	$N_{A\cdot}$
B	n_{BA}	n_{BB}	$N_{B\cdot}$
Total	$N_{\cdot A}$	$N_{\cdot B}$	$4t$

Table 3.3: *Summary of the transmission/non-transmission allele counting for a biallelic marker.* TDT allele counting, adapted from [50].

In Table 3.3, n represents the number of parents who have genotype ***A,B*** and transmit allele ***A*** to the affected offspring. Another approach for the above data would be:

$$TDT = \frac{(n_{AB} - n_{BA})^2}{n_{AB} + n_{BA}} \tag{3.18}$$

which compares the number of ***A*** alleles transmitted to the offspring from theirs parents and the number of ***A*** alleles not transmitted.

Also, it is important to note that the TDT has two key benefits: 1) It only assumes Mendel's first law of inheritance. The specification of the disease model and the distribution of the disease in the general population are not required and will not affect its validity. Thus, the TDT is not only robust to population stratification but also robust to any misspecification of the disease model and the distribution of the disease. 2) The TDT test statistic has an asymptotic chi-square distribution with one degree of freedom if either $\theta = 12$ or $\delta = 0$, where θ and δ are the recombination fraction and the linkage disequilibrium, respectively [50].

3.3.1.2. Sib Transmission/Disequilibrium Test

The TDT requires marker genotypes for affected individuals and their parents. However, for some diseases data from parents may be difficult or impossible to obtain. A method called **sib TDT** or ***Sib Transmission/Disequilibrium Test*** **(S-TDT)** for describing this was implemented by *Spielman and Ewens* [64]. This overcomes the previous problem by use of marker data from unaffected sibs instead of from parents, thus allowing application of the TDT to sibships without parental data. Namely, the S-TDT method uses sibships consisting of at least one affected and one unaffected sib.

In the S-TDT method the number of variant alleles in affected sibs is counted, calculating its mean and variance for each family under the assumption that the proportion of variant alleles is the same in affected sibs as it is in unaffected sibs. These counts are summed over the set of families to form a z score. Several other sibling-based methods have been suggested [65, 66, 67] though the S-TDT remains most closely related to the more recent approaches.

Some families will be suitable to be analysed only by the TDT, and others could be described by the S-TDT. The work by *Spielman et al.* [64] also explains how all the data may be used jointly in one overall TDT-type procedure that tests for linkage in the presence of association. These extensions of the TDT could be interesting for the study of diseases associated with aging [64].

3.3.1.3. Pedigree Disequilibrium Test

The *Pedigree Disequilibrium Test* (PDT) combines the principles of the TDT and S-TDT into a test for general pedigrees [68]. It splits a

pedigree into a list of all case-parent trios and discordant sib pairs with genotype data.

For a trio j, X_{T_j} is defined as the number of transmissions of the variant allele minus the number of its non-transmissions. For a sib pair j, X_{S_j} is defined as the number of copies of the variant allele in the affected sib minus the number in the unaffected sib [61]. The measure of association $\boldsymbol{D}$ for the pedigree would be:

$$D = \frac{1}{N_T + N_S}\left[\sum_{j=1}^{N_T} X_{T_j} + \sum_{j=1}^{N_S} X_{S_j}\right] \tag{3.19}$$

where N_T and N_S are the total number of trios and discordant sib pairs, respectively. D has expectation 0 in any pedigree. After computing this measure for each $i = 1, ..., N$ pedigrees, the PDT statistic $\boldsymbol{T}$ is then:

$$T = \frac{\sum_{i=1}^{N} D_i}{\sum_{i=1}^{N} D_i^2} \tag{3.20}$$

This gives a valid test of linkage or association in any pedigree structure, although some pedigrees are uninformative, notably the affected sib pair. The PDT has been adapted to test quantitative traits [69], haplotypes [70] and genotypes [71]. A very flexible approach for constructing unbiased tests is implemented in the software FBAT [72].

3.3.2. Pedigree Structures with Missing Data

Two general approaches have emerged to deal with the problems of missing family members: 1) Fitting a statistical model to the missing data and conducting an analysis that takes all of the possible completions into account. 2) Developing test statistics that are unbiased under the null hypothesis while using only the available data. This approach retains complete robustness to population stratification and can be readily applied to arbitrary pedigree structures [61].

For instance, a case of particular interest is the late-onset disease in missing parents [61], where unaffected siblings may be used as controls, as long as their relationship to the affected ones has been considered.

Chapter 4

Concepts of Computational Genomics

4.1. Computational Genomics

Computational genomics refers to the use of computational and statistical analysis to decipher biology from genome sequences and related data [73]. Computational genomics focuses on understanding the human genome, and more generally the principles of how DNA controls the biology of any species at the molecular level. With the current abundance of massive biological datasets, computational studies have become one of the most important means to biological discovery [74].

During the past few years, there have been enormous advances in genomics and molecular biology, which carry the promise of understanding the functioning of whole genomes in a systematic manner [75]. The challenge of interpreting the vast amounts of biological data has led to the development of new tools in the fields of ***computational biology*** and ***bioinformatics***, and opened new connections to areas such as chemometrics, exploratory data analysis, statistics, machine learning, and graph theory.

4.2. Computational Biology vs. Bioinformatics

Computational biology can be defined as the study of biology using computational techniques. Its primary goal is to learn new biology, i.e. knowledge about living systems from a more scientific perspective.

Bioinformatics is more focused on the creation of useful tools, algorithms, and software to solve problems relating to biological data. It could also be said that bioinformatics usually manages biological problems and data from an engineering point of view.

Mathematical (statistical) and computational techniques are continuously being developed, analyzed, and improved to offer greater utility in biological arenas. Computational biology and bioinformatics employ powerful tools from computer science, applied mathematics, and statistics to solve those biological problems.

Major avenues of research include genome assembly *–piecing together a large collection of short DNA sequences–*, gene finding *–locating patches of DNA that have biological function–*, genomic sequence alignment *–ordering of multiple sequences to elucidate their parallel structure–*, protein structure prediction *–determining the 3D structure of proteins from their chemical makeup–*, and phylogenetic analysis *–studying and modeling evolutionary relationships between species–* [12].

4.3. Machine Learning in Biology

The term **Machine Learning** [76] (also known as ***Data Mining***) is a research area of computer science which refers to a set of topics dealing with the creation and evaluation of algorithms that facilitate pattern recognition, classification, and prediction, based on models derived from existing data. Its development in recent years is due to the advances in data analysis research, growth in the database industry and the resulting market needs for methods that are capable of extracting valuable knowledge from large databases [77].

In biology-related research areas, machine learning appears as one of the main drivers of progress, where most of the targets of interest deal with complex structured objects: sequences, 2D, and 3D structures or interaction networks. At the same time, bioinformatics and systems biology have already induced new significant developments of general interest in machine learning, for example in the context of learning with structured data, graph inference, semi-supervised learning, system identification, and novel combinations of optimization and learning algorithms [78].

Molecular biology in particular and more generally all biomedical sciences are undergoing a genuine revolution as a result of the emergence and growing impact of a series of new disciplines sharing the

-omics suffix in their name. These include in particular genomics, transcriptomics, proteomics, and metabolomics, devoted respectively to the examination of the entire systems of genes, transcripts, proteins and metabolites present in a given cell or tissue type [79]. Figure 4.1 shows a scheme of some of these biological domains where computational methods are applied for knowledge extraction from biological data.

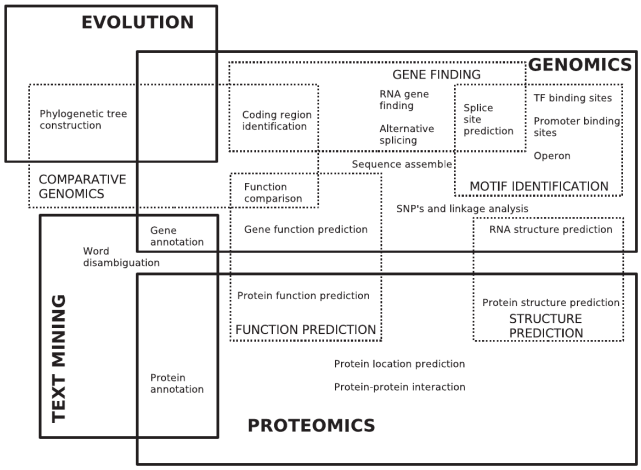


Figure 4.1: *Classification of some topics where machine learning methods are applied.* Machine learning topics in Biology, adapted from [80].

4.3.1. Machine Learning Approaches and Paradigms

As machine learning is mainly concerned with the discovery of models, patterns and other regularities in data, its approaches can be firstly categorized as follows:

- **Symbolic approaches**, including ***inductive learning of symbolic descriptions***, such as rule learning, decision trees or logical representations.
- **Statistical approaches**, including ***Statistical or pattern-recognition methods*** as *k-Nearest Neighbors* (k-NN) or ***instance-based learning*** such as Bayesian classifiers, neural network learning, and *Support Vector Machine* (SVM).

Although the approaches taken in these fields are often considerably different, their effectiveness in learning is often comparable. Also, there are many approaches that cross the boundaries between both approaches. For example, there are decision tree and rule learning algorithms that are firmly based on statistics. Similarly, ensemble techniques such as boosting, bagging or random forests may combine the predictions of multiple logical models on a sound statistical basis [77].

Machine learning can also be classified in two main approaches, both having potential applications in biology:

- In **supervised learning**, objects in a given collection are classified using a set of attributes, or features. The result of the classification process is a set of rules that prescribe assignments of objects to classes based solely on values of features [76].
- In **unsupervised learning**, no predefined class labels are available for the objects under study. The goal is to explore the data and discover similarities between objects. These similarities are used to define groups of objects, referred to as clusters, i.e. unsupervised learning is intended to unveil natural groupings in the data [76].

Under certain applications both paradigms may be mixed, such as protein structure classification, where only a few labeled samples *(protein sequences with known structure class)* are available, while many other samples *(sequences)* with unknown class are available as well [81]. In such cases, **semi-supervised** techniques can be applied to obtain a better classifier than could be obtained if only the labeled samples were used [76].

4.4. Applied Computational Techniques

Diverse computational methods are used to model families of biological sequences as well as for **genotype imputation** and **haplotype estimation**. Many of them are based on the following statistical models and algorithms:

4.4.1. Hidden Markov Model

The ***Hidden Markov Model*** **(HMM)** is a mathematically elegant and computationally tractable class of models in which the observed data are generated by an unobserved Markov process. It is based on ***Markov processes*** in which the distribution of future states (for example, the states

that are further along the chromosome) depends only on the current state and not on previous states [1].

A first-order discrete HMM can be completely defined by a set of states S, an alphabet of m symbols, a transition probability matrix **T**= (t_{ij}), and an emission probability matrix **E**= $(e_{i\alpha})$. When the system is in state i, it has a probability t_{ij} of moving to state j and a probability $e_{i\alpha}$ of emitting symbol α. Only the output string is observed, one of the goals being the reconstruction of the underlying hidden transitions [82].

As in the application of HMMs to speech recognition, a family of biological sequences can be seen as a set of different expressions of the same word generated by a common underlying HMM with a left-right architecture, with m = 4 for DNA or RNA and m = 20 for proteins [82]. Usual HMM operation is based on three classes of states: ***main***, ***start and stop***, and ***side*** states. So, considering N as the length of the model, a graphical representation of the sequence: **Seq** = $m_1, ..., M_N, i_1, ..., i_{N+1}, d_1, ..., d_{N+1}$, is shown in Figure 4.2.

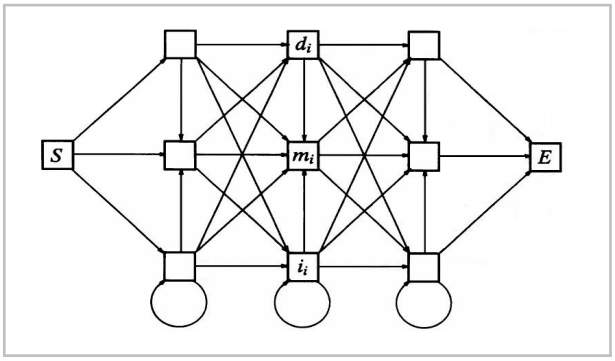


Figure 4.2: *Linear sequence of **main** states transitions where m_i is the backbone of the model, **S** and **E** are the **start** and **end** states, and d_i (deletions) and i_i (insertions) are the **side** states. Self-loops on i states allow for multiple insertions.* HMM Architecture, adapted from [82].

Given a set of ***training sequences***, the parameters of a model can be iteratively modified to optimize the fit of the model to the data according to some measure, usually the product of the likelihoods of the sequences. Different techniques for HMM training are available, such as ***segmental K-means***, ***Baum-Welch*** and ***Baldi-Chauvin*** algorithms [83, 84].

4.4.1.1. HMM in genetics

HMMs can be used to relate an observed process across the genome to an underlying, unobserved process of interest [85]. Such models have been used, for example, to estimate population structure and admixture, to impute genotypes, and for estimate haplotypes.

Regarding haplotype inference and defining a sequence of observed genotypes as: $Y = (y_{t-1}, ..., y_t, ..., y_{t+2})$, the ***Probability*** of observing such sequence would be: $P(Y) = \sum_X P(Y|X)P(X)$, where Y is the observed genotype and X the haplotype pair. Therefore, according to ***Markov Property***, the state of a node X_t only will depend on the state of the previous node X_{t-1} in the sequence.

If the states of the model are defined as ***Emission Probability***: $P(Y|X)$ and ***Transition Probability***: $P(X_t|X_{t-1})$, the first will depend on the possible mutations or genotyping errors of the sequence, while the second will depend on the changes in membership of markers (i.e. inter-marker recombination). The latter condition will control the way the hidden state at time *t* is chosen given the hidden state at time *t-1*. A graphical scheme of the previous explanation is shown in Figure 4.3.

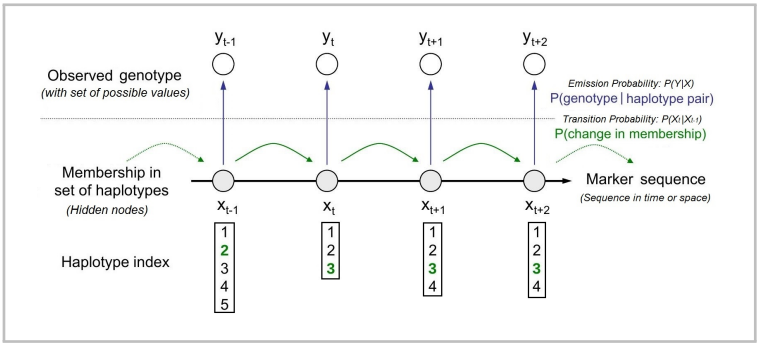


Figure 4.3: *HMM implemented with a marker sequence for inferring Haplotype Data.* HMM for Haplotype Data, adapted from [84].

Another example of genotypic/haplotypic data estimation with a larger number of subjects is depicted in Figure 4.4. In this example, the states are represented as circles, the *j*th column of four states corresponding to the *j*th marker. A gray scale is used to indicate the emission probability of each allele, where black colour corresponds to a probability of 1. The thickness of each transition line indicates the corresponding transition probability [86].

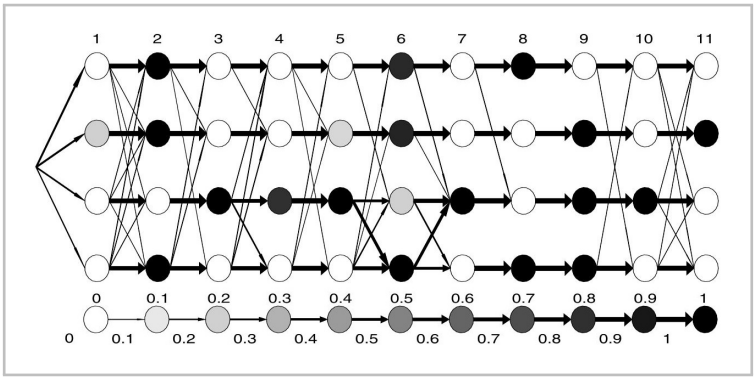


Figure 4.4: *HMM example for inferring genotype data, with m = 11 markers and K = 4 founders.* HMM for Genotype Data, adapted from [86].

4.4.2. Expectation-Maximization Algorithm

The *Expectation-Maximization* (EM) algorithm is a general method of finding the *Maximum a Posteriori* (MAP) or *maximum-likelihood* estimates of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values [87]. It is essentially an iterative optimisation algorithm that, at least, under certain conditions, will converge to parameter values at a local maximum of the likelihood function [88].

EM is iterative, which means that it alternates between performing an expectation (E) step and a maximization (M) step. The E step creates a function for the expectation of the log-likelihood evaluated using the current estimate of the parameters, and an M step computes parameters maximizing the expected log-likelihood found on the E step (see Figure 4.5). These parameter estimates are then used to determine the distribution of the latent variables in the next E step [89].

The M step of the algorithm may be only partially implemented, with the new estimate for the parameters improving the likelihood given the distribution found in the E step, but not necessarily maximizing it. Such a partial M step always results in the true likelihood improving as well, referring to such variants as *Generalized Expectation Maximization* (GEM) algorithms [91]. In many cases, when the distribution for one of the variables is re-calculated, it makes sense to re-estimate the parameters immediately before performing the E step for the next unobserved variable, as this utilizes the new information instantly [92].

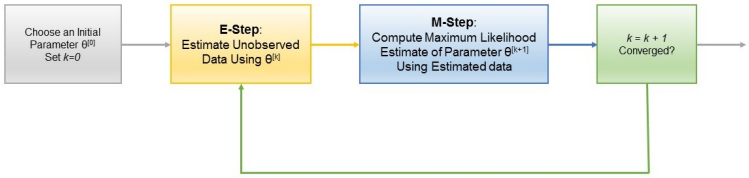


Figure 4.5: *After initialization, the E-step and the M-step are alternated until the parameter estimate has converged (no more change in the estimate).* Overview of the EM algorithm, adapted from [90].

There are two main applications of the EM algorithm. The first occurs when the data indeed has missing values, due to problems with or limitations of the observation process. The second occurs when optimizing the likelihood function is analytically intractable but when the likelihood function can be simplified by assuming the existence of and values for additional but *missing* (or *hidden*) parameters. The latter application is more common in the computational pattern recognition community [87].

4.4.2.1. EM Algorithm in genetics

One of the first implementation of an EM algorithm applied to genotype/haplotype estimations was developed by [93]. Its use may be extended to other different goals, such as finding the list of the most frequent haplotypes in a sample, estimating haplotype population frequencies, inferring which gametes are most likely associated to form genotypes in all sampled individuals, or finding the best estimates of coefficients of linkage disequilibrium between loci [94].

A brief summary of the EM algorithm implementation for calculating haplotype frequencies follows:

1. **Initial parameter values**: Haplotype frequencies: $f_1, ..., f_h$.
2. **Expectation step**: compute expected values of missing data based on initial data.
3. **Maximization step**: compute *Maximum-Likelihood Estimation* (MLE) for parameters from the complete data.
4. **Repeat** with the new set of parameters until changes in the parameter estimates are negligible.

4.5. International Genetic Databases

Large-scale genetic databases of human populations, containing data on genome-wide SNPs, genotypes, inferred haplotypes, mutation and recombination structures, have become publicly available [50]. These databases have provided valuable information for understanding genetic variation patterns and for inferring evolutionary histories of human populations. In the next, two of the most freely used resources are briefly described. Furthermore, the international T1DGC program is also disclosed.

4.5.1. HapMap Project

The **International HapMap Project** [95] was a worldwide effort to identify and build a catalog of genetic variants in human populations. This catalogue describes what these variants are, where they occur in the genome, and how they are distributed among individuals within and among populations distributed around the world [50].

To develop a comparison of the DNA sequences among individuals to identify chromosomal regions where genetic variants are shared, firstly it is necessary to know which are common to the entire world population, but there are some more prevalent in certain populations. The HapMap Project tried to detect those common haplotypes selecting so-called tag SNPs which might uniquely identify these haplotypes for all the populations. The number of tag SNPs that capture most of the information of genetic variation patterns was estimated to be between 300000 and 600000 SNPs [95]. A graphical description of previous explanation is depicted in Figure 4.6.

To achieve its main purpose, the project comprised three phases, and each one led to the identification of more than a million of SNPs:

- **Phase I** (2005): Genotyping of 1 SNP for each 5 kb (5000 bases). More than a million SNPs were identified.
- **Phase II** (2007): Identification of 2 million SNPs (in addition to those already described in Phase I), reaching more than 3.1 million.
- **Phase III** (2009): Identification of 1.6 million additional SNPs. Increment of the number of analyzed populations from 5 to 11.

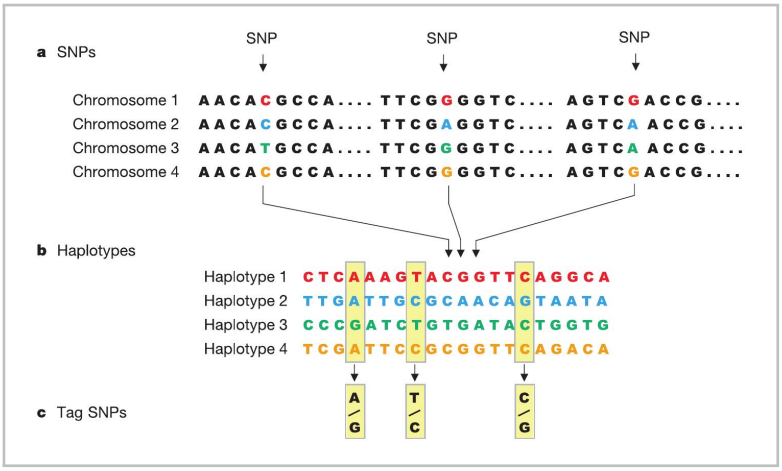


Figure 4.6: ***a) SNPs.*** *A short stretch of DNA from four versions of the same chromosome region in different people.* ***b) Haplotypes.*** *Observed genotypes for 20 SNPs that extend across 6000 bases of DNA.* ***c) Tag SNPs.*** *Genotyping just the three tag SNPs out of the 20 SNPs is sufficient to identify these four haplotypes uniquely.* SNPs, haplotypes, and tag SNPs [95].

The DNA samples of HapMap Project in ***Phase I***, ***II*** and ***III*** came from a total of 1184 individuals from the following 11 populations: **ASW**: African ancestry in Southwest (USA), **CEU**: Utah residents with Northern and Western European ancestry (USA), **CHB**: Han Chinese in Beijing (China), **CHD**: Chinese in Metropolitan Denver (USA), **GIH**: Gujarati Indians in Houston (USA), **JPT**: Japanese in Tokyo (Japan), **LWK**: Luhya in Webuye (Kenya), **MXL**: Mexican ancestry in Los Angeles (USA), **MKK**: Maasai in Kinyawa (Kenya), **TSI**: Toscani in Italy and **YRI**: Yoruba in Ibadan (Nigeria) [96].

The development of the HapMap has enabled geneticists and other specialists to take the advantage of how SNPs and other genetic variants are organized on the same chromosome [95].

The HapMap project has also helped researchers to find functional regions that influence human health outcomes as well as responses to therapeutic drugs and environmental factors. The project itself has not identified such regions directly. Instead, HapMap has provided a tool that can be used in both population-based and family-based disease association studies [97].

4.5.2. 1000 Genomes Project

The ***1000 Genomes Project*** [98] has sequenced the genomes of more than 1000 people, to provide a comprehensive resource on human genetic variation. Its primary goal was to find most genetic variants that have frequencies of at least 1 percent in the populations studied [98].

The samples used in this project were mostly anonymous and had no associated medical or phenotypic data. Despite this fact, the genetic variation data produced by the 1000G Project has been used by researchers to study many diseases, in sets of case and control samples that were carefully phenotyped [98]. Extended information about using the project data is available in 1000 Genomes Project website.

This resource, which captured up to 98 percent of accessible SNPs at a frequency of 1 percent in related populations, has enabled numerous analysis of common and low-frequency variants in subjects from diverse, including admixed, populations. The individual samples collected were grouped in five geographic areas according to the regions of the ancestries [99]:

I. ***East Asian Ancestry*** **(EAS)**: Chinese Dai in Xishuangbanna (CDX), Han Chinese in Bejing (CHB), Vietnamese Kinh in Ho Chi Minh City (KHV), Southern Han Chinese (CHS).

II. ***South Asian Ancestry*** **(SAS)**: Bengali in Bangladesh (BEB), Gujarati Indian in Houston (GIH), Indian Telugu in the UK (ITU), Pakistani Punjabi in Lahore (PJL), Sri Lankan Tamil in the UK (STU).

III. ***African Ancestry*** **(AFR)**: African Ancestry in Southwest US (ASW), African Caribbean in Barbados (ACB), Esan in Nigeria (ESN), Gambian in Western Division (GWD), Kenyan Luhya in Webuye (LWK), Mende in Sierra Leone (MSL), Nigerian Yoruba in Ibadan (YRI).

IV. ***European Ancestry*** **(EUR)**: British in England and Scotland (GBR), Finnish in Finland (FIN), Iberian populations in Spain (IBS), Toscani in Italia (TSI), Utah residents with Northern and Western European ancestry (CEU).

V. ***Americas Ancestry*** **(AMR)**: Colombian in Medellin (CLM), Mexican Ancestry in Los Angeles (MXL), Peruvian in Lima (PEL), Puerto Rican in Puerto Rico (PUR).

In this project 2500 samples at 4X coverage have been sequenced. The first set of samples for sequencing included 1167 samples and were collected from 13 populations during 2010 and early 2011. The second set included 633 samples that was collected from 7 populations in early 2011. The third set, consisting of 700 samples, was collected for sequencing in late 2011 [100]. Full details of the samples are shown in Table 4.1.

Population Group	Pilot Samples	Set 1 Samples	Set 2 Samples	Set 3 Samples	Total
EAS	185	286	515	504	523
SAS	0	0	494	489	494
AFR	208	246	669	661	691
EUR	160	379	505	503	514
AMR	0	181	352	347	355
Total	553	1092	2535	2504	2577

Table 4.1: *Summary of the sequenced samples the 1000 Genomes project.* 1000 Genomes Samples, adapted from [99].

Finally, the international 1000 Genomes Project has provided a validated haplotype map of more than 84 million SNPs, 3.1 million short insertions and deletions, 42.279 biallelic deletions, 6.025 biallelic duplications, 2.929 mCNVs (multiallelic copy-number variants), 786 inversions, 168 *Nuclear Mitochondrial Insertion* (NUMT)s, and 16.631 mobile element insertions of 2.504 individuals from 26 populations. Despite the great genetic diversity, most variants (86% of 84.7 million) are restricted to a single continental group, particularly among sub-Saharan populations in Africa [100].

4.5.3. T1DGC Project

The T1DGC [101] is an international, multicenter program organized to identify genes and their alleles that determine an individual's risk for *Type 1 Diabetes* (T1D). The program had two primary goals:

I. Identifying genomic regions and candidate genes whose variants modify an individual's risk of T1D and help explain the clustering of the disease in families.

II. Making research data available to and establish resources that can be used by the research community.

The T1DGC assembled a resource of affected sib-pair families, parent-child trios, and case-control collections with banks of DNA, serum, plasma, and cell lines [101]. In addition to T1DGC-recruited *Affected Sib-Pair* (ASP) families, the T1DGC recruited trio families from ethnic groups with a lower prevalence of T1D. The T1DGC also welcomed the inclusion of earlier ascertained case-control collections [102].

Research with T1DGC data has included genome-wide linkage scans, evaluation of the human MHC, examination of published candidate genes for T1D, and analysis of autoimmune disease genes and those affecting β-cell function in type 2 diabetes [101]. All the information on T1DGC can be accessed at the T1DGC website.

Figure 4.7 shows the criteria of pedigree structure for inclusion of families. The maximal included pedigree structure includes five affected and two unaffected siblings in affected sibling pair families; no additional siblings were collected in trio families. All recruited family members were typed for Class I and II HLA loci [101].

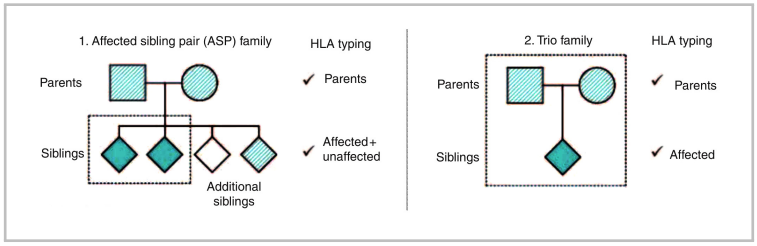


Figure 4.7: *Dark fill represents a family member with type 1 diabetes; no fill, unaffected; and crosshatch may be either. The dotted line indicates the minimum inclusion criteria for family recruitment into the T1DGC collection.* Pedigree structures into the T1DGC, adapted from [103].

Eligibility criteria [102] included the following:

1. Siblings with a diagnosis of T1D.
2. Diagnosis before 35 years of age.
3. Use of insulin within 6 months of diagnosis.
4. Continuous use of insulin (without stopping for 6 months or more).
5. Informed consent for blood collection, genetic analysis, and exam.

In addition, trio families were collected in selected populations with a low prevalence of the disease throughout the Asia-Pacific, European and North American Networks. The required family structure was an affected child and both biological parents. Eligibility criteria are the same as listed above. Cases and controls were collected throughout the Asia-Pacific, European and North American Networks in selected low-prevalence populations. Outcome measures included the establishment of resources for research into the genetic origins of T1D and identification of genomic regions and genes whose variants contribute to an individual's risk of T1D. Phenotype and genotype data from study participants has been widely used in research studies concerning the genetic origins of T1D risk in families and the general population [102].

Chapter 5

State of the Art

5.1. Introduction

The knowledge about human genetic variation has been growing exponentially over the last decade. Collaborative efforts such as the International HapMap [97] and 1000 Genomes [98] projects have contributed to increase the rate of discovery about human genetic diversity.

Genome population-based studies usually employ DNA microarrays to produce genotype information for a set of individuals. If those studies where collected with different array types, some markers may not be assayed at the same genomic positions. However, recent computational advances enable researchers to use algorithms to fill in or impute genotypes (see Figure 5.1) at the markers that are not common among the genotyping arrays [104].

Given the genotypes of a sample of individuals from a population, using haplotype pre-phasing to infer firstly the haplotypes of the sample (using haplotype sharing information within the sample) allows to build a phased reference panel which is used to estimate later missing markers of the sample. Therefore, genotype imputation (together with haplotype pre-phasing) techniques usually increases the sample size at each marker, boosting thus the study's power to detect fine-map associations and facilitating the combination of results across different studies using meta-analysis [85].

This chapter comprises the state of the art of several different statistical methods for *genotype imputation*, *haplotype reconstruction*, and *analysis of genome variation*.

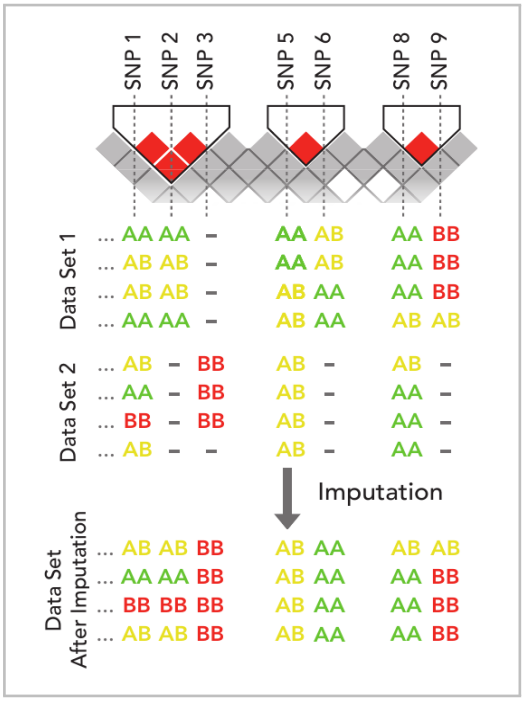


Figure 5.1: *SNPs 1–9 form three blocks of high LD, indicated by the red diamonds between the SNPs.* Imputation overview, adapted from [104].

5.2. Genotype Imputation

Genotype imputation can be fundamental in the analysis of GWA. The most common approach works by finding haplotype segments that are shared between study individuals, which are typically genotyped on a commercial array with 300000–2500000 SNPs [105]. This process can be carried out across the whole genome as part of a GWAS or in a more focused region as part of a fine-mapping study. The goal is to predict the genotypes at the SNPs that are not directly genotyped in the study sample [85]. These *in-silico* genotypes can then be used to boost the number of SNPs which can be tested for association, thus increasing the power of the study, improving fine-mapping of causal variants and facilitating meta-analysis.

On the other hand, many existing genotype imputation methods require substantial computing power to run using large reference datasets. This problem may be aggravated by the fact that reference panels are regularly improved and expanded, in order to investigators can re-impute their samples multiple times over the course of a study [105].

5.2.1. Uses of Genotype Imputation

The uses of genotype imputation are several and can range from the *imputation of untyped variation* to *meta-analysis*. Next subsections enumerate and describe them briefly.

5.2.1.1. Imputation of Untyped Variation

Imputation of SNPs that have not been typed in either the haplotype reference panel or the study sample is also possible. Some methods do this via inference of the genealogy between study sample haplotypes [106, 107] while others aim to identify haplotype effects more directly [108]. These methods can lead to a boost in power, especially when the causal variant is rare, or where there is local heterogeneity in the signal of association [109].

5.2.1.2. Imputation of Non-SNP Variation

The general idea of imputation is readily extended to other types of genetic variation such as *Copy Number Variant* (CNV)s and classical HLA alleles [110]. The imputation of large numbers of small insertions and deletions (indels) which will be discovered from recent international sequencing based projects (as the 1000 Genomes project [100]) are likely to be widely adopted in GWAS studies [116].

5.2.1.3. Boosting Power

Imputation can lead to a notable boost in the power of a GWAS. *Spencer et al.* [111] illustrate how imputation could produce a power improvement of 5 to 10% if the density of the chips is close to that of a hypothetical complete chip consisting of all SNPs. Other simulations have shown that the biggest benefit occurs for rare SNPs that are harder to tag [112].

5.2.1.4. Fine-mapping

Imputation provides a much higher resolution view of an associated region than would be seen by just considering genotyped SNPs (see Figure 5.2) and increases the chance that a causal SNP can be directly identified. When imputed SNPs produce larger signals than any of the genotyped SNPs they can become better candidates for replication in new samples. Imputation methods can also help elucidate when multiple variants or allelic heterogeneity occurs in a region of interest [113, 109].

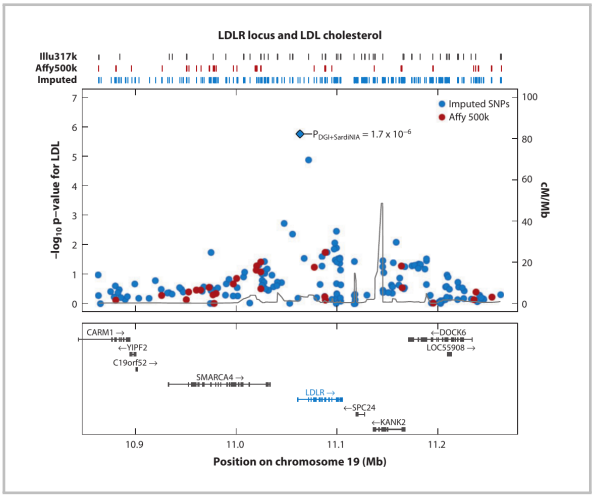


Figure 5.2: *Association of genetic variants near LDLR with LDL-cholesterol levels using imputed data.* An example of genetic variants association using imputed data [114].

5.2.1.5. Meta-analysis

Imputation has been widely used to facilitate meta-analysis of GWAS from different cohorts that may have been genotyped using different genotyping chips, i.e., different sets of SNPs. Imputation effectively enlarges and equates the set of SNPs in each study for which genotypes are available for testing. A useful practical guide is provided by [115]. Results from cohort-specific GWAS are combined using fixed effects models rather than combining the raw data from all studies and then carrying out one association test.

5.2.2. Genotype Imputation Methods

Several methods have been proposed for genotype imputation. These methods can provide a boost in imputation accuracy, mostly at rarer SNPs and those SNPs that are not well tagged by a small number of flanking SNPs. Most imputation methods are based on HMM, a very useful class of statistical model readily applicable in genetics. HMM model is used for relating an observed process across the genome to an underlying, unobserved process of interest [116]. Thus, the most commonly used programs for genotype imputation are:

5.2.2.1. IMPUTE v1

IMPUTE1 [112] is based on an extension of the HMM models originally developed as part of importance sampling schemes for simulating coalescent trees [117, 118] and for modeling linkage disequilibrium and estimating recombination rates [119]. The method is based on the HMM of each individual's vector of genotypes conditional upon a reference set of haplotypes and an estimate of the fine-scale recombination map across the region. Exact marginal probability distributions for the missing genotypes are obtained using the forward-backward algorithm for HMMs [120].

5.2.2.2. IMPUTE v2

IMPUTE2 [121] takes a different and more flexible approach. SNPs are first divided into two sets: a T set that is typed in both study sample and reference panel, and a U set that is untyped in the study sample but typed in the reference panel. As is depicted in Figure 5.3 the algorithm estimates haplotypes at SNPs in T (blue) using IMPUTE1 and then imputing alleles at SNPs in U (green) conditional upon the current estimated haplotypes. Since imputation performance is driven by accurate matching of haplotypes, the method focuses on accurate haplotype estimation at the SNPs in T using as many individuals as possible [85].

5.2.2.3. MACH

MACH [122] uses an HMM model very similar to that used by HOT-SPOTTER [119] and IMPUTE. Using this method, genotype phasing can be performed and, therefore, used for imputation. The method works by successively updating the phase of each individual's genotype

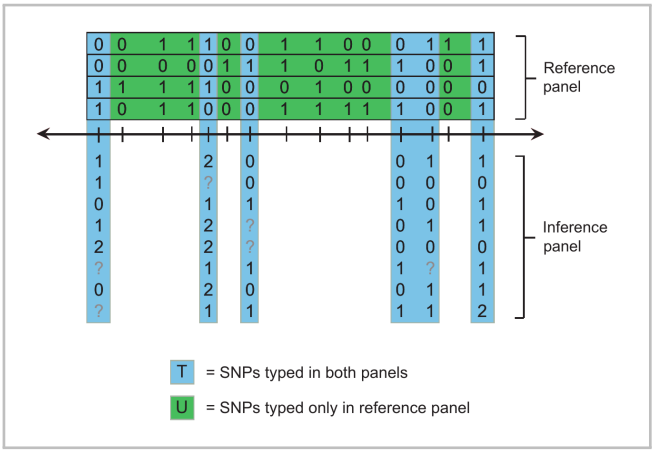


Figure 5.3: *The goal of imputation in this scenario is to estimate the genotypes of SNPs in set U in the study sample, or "inference panel".* Standard imputation scenario used by IMPUTE2 [121].

data conditional upon the current haplotype estimates of all the other samples. The model involves "crossover" and "error" parameters that are updated as the algorithm progresses. This process naturally imputes any missing genotype data, and marginal genotype probabilities can be reported as a summary.

5.2.2.4. MINIMAC

MINIMAC [105] is a computationally efficient implementation of the MACH algorithm. This low memory software was designed to work on phased genotypes, handling very large reference panels with hundreds or thousands of haplotypes. Later, MINIMAC2 [123] and MINIMAC3 [124] were released as improved versions of MINIMAC [105], the last version designed to handle larger reference panels in a more computationally efficient way. It accomplishes this by identifying repeat haplotype patterns and using these to simplify the underlying calculations, with no loss of accuracy [124].

5.2.2.5. BEAGLE

BEAGLE [125] is a program based upon a graphical model of a set of haplotypes. The method works iteratively by fitting the model to the current set of estimated haplotypes and then resampling new estimated haplotypes for each individual [2]. This model is empirical in the sense that it has no parameters that need to be calculated and is applied to a given set of haplotypes in two steps. Another advantage of the program is that, as it involves no parameter estimation steps, it is very fast to fit to a dataset [116].

5.2.2.6. fastPHASE/BIMBAM

fastPHASE [126] is a method that can be used to haplotype estimation and genotype imputation. Later, it has been incorporated into an association testing program called BIMBAM [113]. The method uses the observation that haplotypes tend to cluster into groups of closely related or similar haplotypes. The model specifies a set of K unobserved states (clusters) that are meant to represent common haplotypes, and each individual's genotype data is then modeled as a HMM on this state space [127].

5.2.2.7. Other Approaches

Other interesting approaches are based on SNP tagging. This method has been established for some time in association studies [128, 129] and the methods for genotypes imputation implemented in the programs PLINK [130], SNPMStat [131], UNPHASED [132] and TUNA [133] all are based on this approach. The advantages of these methods are their simplicity and speed. In contrast, the main disadvantage is that these methods do not provide as accurate results as the others because they do not use all the data and the phasing is carried out using a simple multinomial model of haplotype frequencies [93].

5.2.2.8. Method Benchmarking

Benchmarking conducted by Laughbaum [134] tested the three most used imputation programs: BEAGLE, IMPUTE2, and MINIMAC. In this study BEAGLE and IMPUTE2 tests were performed both with and without pre-phasing (later explained in Section 5.3.2.1). Since the

MINIMAC method utilizes pre-phasing by default computational constraints, the pre-phasing step was not necessary to separately implement it. All programs outperformed others in certain areas. Based on all of the metrics measured, IMPUTE2 seemed to perform with the greatest accuracy and quality although other programs performed better on other aspects [134].

Software	Comp. Time[1]	SNP Conc.[2]	Total No. SNPs	No. HQ[3] SNPs	% HQ[3] Imputed
IMPUTE2	23 h	99.98%	668,180	620,792	92.9%
BEAGLE	213 h	98.43%	484,023	320,991	66.3%
IMPUTE2[3]	8 h	99.92%	668,180	297,196	44.5%
BEAGLE[3]	34 h	98.05%	484,023	293,890	60.7%
MINIMAC[3]	18 h	96.25%	667,870	450,790	67.5%

[1] Computation time includes all steps required.
[2] Mean SNP Concordance.
[3] *High Quality* (HQ).
[4] With pre-phasing.

Table 5.1: Comparison of traditional and pre-phasing imputation methods, adapted from [134].

As will be later explained in Section 5.3.2.1, pre-phasing the original dataset drastically improved computational times. Then, when the data was pre-phased, IMPUTE2 ran the quickest, followed by MINIMAC, and then BEAGLE. Without pre-phasing, IMPUTE2 was much faster than BEAGLE. IMPUTE2 also had superior concordance rates, although all software programs performed well in this area. MINIMAC had the lowest concordance rate at 96.25% [134].

5.3. Haplotype Reconstruction

Estimation of haplotypes is becoming increasingly important as we enter the era of large-scale sequencing because many of its applications, such as imputing low-frequency variants and characterizing the relationship between genetic variation and disease susceptibility, are particularly relevant to sequence data [1].

Haplotype phase can be generated through laboratory-based experimental methods (***Haplotype Assembly***), or it can be estimated using computational approaches (***Haplotype Phasing***). The following subsections

will describe them, emphasizing pre-phasing methods and comparing practical computational aspects. A summary of the haplotype reconstruction methods for family-based data and HLA region will also be presented.

5.3.1. Haplotype Assembly

Haplotype assembly, also called *single individual haplotyping* [135], is the process in which the haplotypes for a single individual are built from a set of sequence reads [136]. The haplotype assembly problem aims to compute the haplotype sequences for each chromosome given a set of aligned sequence reads to the genome and variant information. After mapping the reads on a reference genome, reads are translated into haplotype fragments containing only the polymorphic SNP sites. A fragment covers a SNP if the corresponding sequence read contains an allele for that SNP. Because DNA sequence reads originate from a haploid chromosome, the alleles spanned by a read are assumed to exist on the same haplotype [137]. Haplotype assembly algorithms operate on either a SNP-fragment matrix containing a row for each fragment and columns for SNPs or an associated graph that models the relationship between fragments or their SNP alleles [138].

5.3.2. Haplotype Phasing

Methods for haplotype phasing and genotype imputation are based on computational [139] and statistical inference techniques [1], but both use the fact that closely spaced markers tend to be in linkage disequilibrium, and smaller haplotypes blocks are often shared in a population of seemingly unrelated individuals [137].

For population-based studies, the typical scenario is to consider a sample of unrelated individuals. If these subjects are genotyped at three SNPs (with alleles denoted A and a at the first locus, B and b at the second, and C and c at the third) for an individual with those three loci and genotype **AaBbCc**, there are four possible diplotypes:

1:	**ABC**	**abc**
2:	**ABc**	**abC**
3:	**AbC**	**aBc**
4:	**Abc**	**abC**

Therefore, if a diploid genotype contains k heterozygous variants, then there are 2^k possible diplotypes consistent with a multi-locus genotype. Fortunately, although an exponential number of haplotype pairs are possible, very few exist in the population. Thus, this problem has prompted the development of statistical methods to resolve diplotypes directly using unphased SNP genotype data [116].

Next subsections will describe and compare most used phasing programs, and discuss how haplotype phasing has been used as a previous step of genotype imputation.

5.3.2.1. Pre-Phasing Process

Genotype imputation using large reference panels is usually a very demanding computational task. Thus, a strategy called *pre-phasing* has been widely adopted to maintain the accuracy of leading methods while reducing computational costs. In order to carry out this task, the first step is to estimate the haplotypes for each individual within the GWAS sample and then impute missing genotypes into these estimated haplotypes [105].

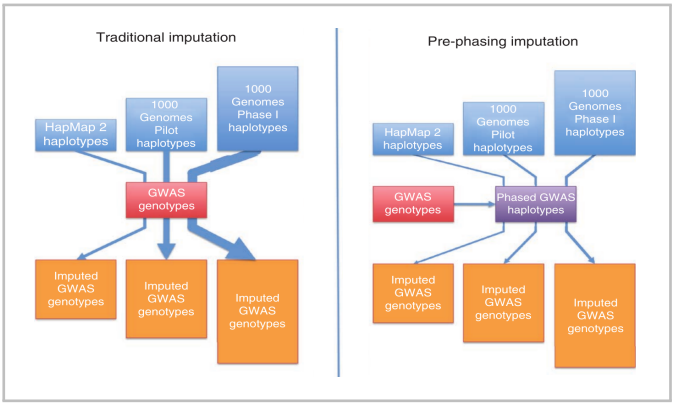


Figure 5.4: *Each box represents a genetic data set, and each arrow represents an analysis step.* Traditional and pre-phasing imputation scenarios, adapted from [105].

Schemes of a traditional workflow and the more efficient pre-phasing approach are shown in Figure 5.4. The sizes of the boxes reflect the relative numbers of genotypes they contain, and the widths of the arrows reflect the relative computational costs of the analyses. Given a

single GWAS data set (red box), successively larger reference panels (blue boxes) lead to larger and more accurate imputed data sets (orange boxes) [105].

5.3.2.2. Phasing Methods

Among all haplotype phasing (or pre-phasing) existing methods, one deserves a particular mention: the *Segmented Haplotype Estimation and Imputation Tool* (SHAPEIT), proposed by *Delaneau et al.* [140]. This method uses genotype data from unrelated samples or small nuclear families, which leads to improved accuracy and speed compared to other widely used methods. A more recent version called SHAPEIT2 [141] combines features of SHAPEIT and IMPUTE2 to enhance performance substantially.

In [140] the performance of SHAPEIT method against other widely used methods (as BEAGLE, FASTPHASE, IMPUTE2 and MACH) was investigated. In order to perform this, 925 unrelated phase-known diploid European samples (with 7821 SNPs from male X chromosome) were used.

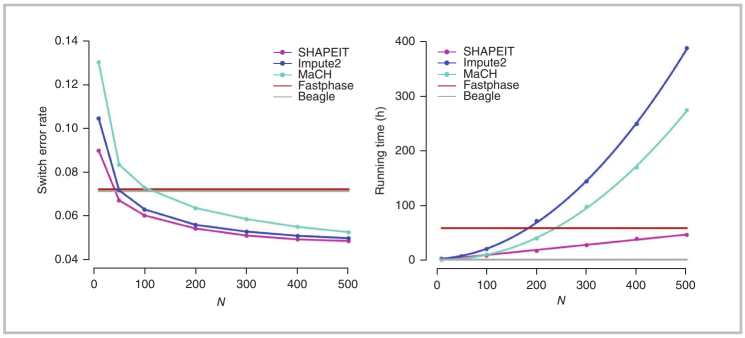


Figure 5.5: *Switch error rate and running times plotted against N (Number of conditioning states).* Phasing methods comparison, adapted from [140].

As is depicted in Figure 5.5 (left), SHAPEIT produced the most accurate results, followed by IMPUTE2, MACH, BEAGLE, and FASTPHASE. Running time (right) illustrates the linear complexity of SHAPEIT compared to the quadratic complexity of Impute2 and MaCH. Memory usage followed a similar pattern [140].

5.3.3. Haplotype Reconstruction within Pedigrees

Existing approaches for haplotype reconstruction can be categorised according to the type of cohort each method is designed to phase, and the level of relatedness between the individuals in such cohort [142]. Much of the recent literature is devoted to phasing nominally unrelated (or distantly related) individuals. Currently, the most accurate methods use HMMs to model local haplotype sharing between individuals [143, 140] and take advantage of LD. Some of these methods can also handle mother-father-child trios and parent-child duos [144, 2]. For more complex pedigrees there are several comprehensive pedigree analysis software packages [145, 146].

As a general strategy for phasing cohorts with any level of implicit or explicit relatedness between individuals, a novel HMM method has been recently proposed by *O'Connell et al.* [142]. To utilize this program (called DuoHMM), firstly SHAPEIT2 has to be run ignoring all explicit family information, to combine later those haplotypes with any family information to infer the inheritance pattern. This method has the advantage that allows the detection of recombination events, genotyping errors and the correction of switch errors (see Figure 5.6).

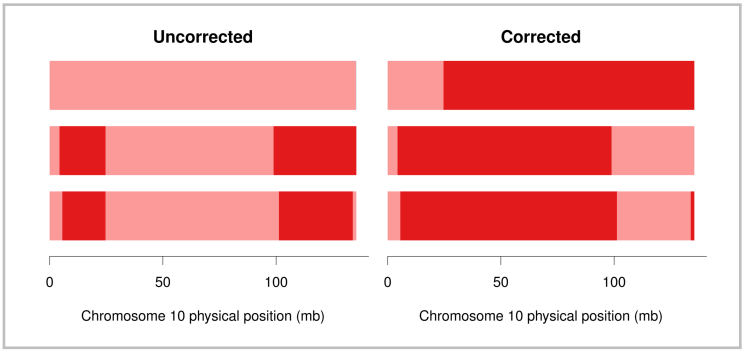


Figure 5.6: *Paths for a three father–child duos from a nuclear family on chromosome 10. Two possible IBD states are shown using the colors light and dark red.* Haplotype correction example using the DuoHMM method [142].

In Figure 5.6, the left panel shows the path prior to any corrections, and the right panel after a minimum recombinant correction is applied. The second and third sibling initially had a transition at around 25 mb, which is more likely a recombination event in the first child hence the parental haplotypes are switched at this point. The panel on the right

has the corrected haplotypes, the number of recombination events required to explain the observed data has been reduced [142].

5.3.4. Haplotype Reconstruction programs

Numerous software packages are tackling the haplotype reconstruction problem. The following tables group them in three categories: most used, R packages and other related programs.

Package Name	Last Ver.	Main aims	Samples Type	Ref.
BEAGLE	4.1	Imputation, phasing and analysis.	Unrelated and nuclear families.	[108]
fastPHASE	1.2	Imputation and phasing.	Unrelated	[126]
IMPUTE	2.3.2	Imputation and phasing.	Unrelated	[121]
MACH	1.0.18	Imputation and phasing.	Unrelated	[122]
Minimac3	1.0.13	Imputation and phasing.	Unrelated	[124]
SHAPEIT	2.20	Alignment and phasing.	Unrelated and nuclear families.	[141]

Table 5.2: Most used programs for genotype imputation and haplotype phasing.

There are almost no R packages related to haplotype reconstruction, inference, phasing or assembly. The ones listed in the CRAN repository are shown in the following table:

Package Name	Last Ver.	Main aims	Samples Type	Ref.
haplo.ccs	1.3.1	Relative risk estimation	Unrelated case-control	[147]
haplo.stats	1.7.1	Inference and analysis	Unrelated	[148]
hsphase	2.0.1	Imputation and phasing	Half-sib families	[149]

Table 5.3: R packages related to haplotype reconstruction.

Other related but less-used programs encompass from genotype calling, haplotype assembly to LD mapping or haplotype association analysis.

Program Name	Last Ver.	Main aims	Samples Type
Arlequin	3.5.2.2	Phasing and analysis	Unrelated
DuoHMM	0.1.7	Phasing	Complex pedigrees
HapCUT	0.6	Assembly	Unrelated
HAPI-UR	1.01	Phasing.	Unrelated
HaploBlock	1.2	Phasing and LD mapping	Unrelated
HaploRec	2.3	Reconstrucion	Unrelated
HapFerret	-	Phasing	Unrelated
HapSeq	2	Calling and phasing	Unrelated
HARSH	0.21	Phasing.	Unrelated
MERLIN	1.1.2	Phasing, LD mapping and analysis	General pedigrees
PLINK	1.07	Imputation, phasing and analysis	Unrelated and pedigrees
PedPhase	3.0	Phasing	Pedigrees
S-MIG++	1.0.0	Phasing and LD mapping	Unrelated
SpeedHap	-	Phasing	Unrelated
ReFHap	1.0.0	Phasing.	Unrelated
WinHAP	2.0	Phasing	Unrelated

Table 5.4: Other programs related to haplotype reconstruction.

5.4. Analysis of Genome Variation

Over the last decade, there has been an increasing number of high-profile GWAS for a variety of different human diseases (see Figure 5.7). These studies have revealed hundreds of disease-associated loci and have provided insights into the study of complex traits. All these kind of studies produced a ***test statistic*** and a ***p-value*** indicating how significant the statistical association between a given SNP and a phenotype is, i.e. how likely a specific phenotype (disease) may have occurred by chance.

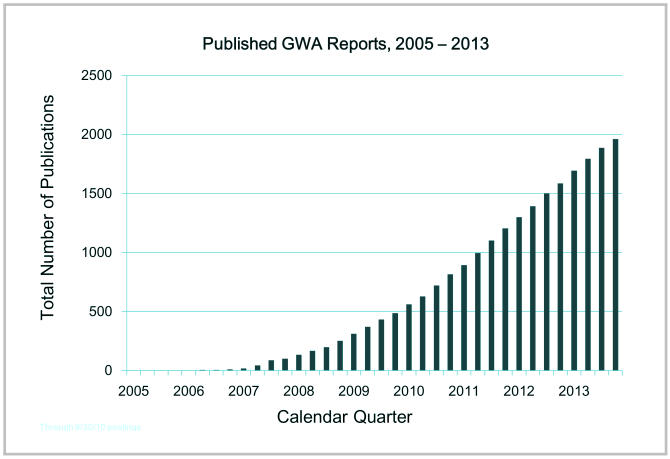


Figure 5.7: *Published GWA reports between the beginning of 2005 and end of 2013.* GWA published reports, adapted from [150].

Usually, GWAS require stringent significance levels ($p \leq 5 \times 10^{-8}$) to overcome the multiple testing problem incurred when testing SNPs throughout the genome. However, in studies with a short-medium sample sizes (less than 1000 subjects), a more lax threshold has to be utilized in order to detect associations (in spite of losing statistical power). In these cases, GWAS may be insufficient to explain the functional or causal variants since the identified associations discovered commonly have small ORs (< 1.5), which suggests that these effects are not such significant.

Recently, the *National Human Genome Research Institute* (NHGRI) in collaboration with *European Bioinformatics Institute* (EBI) have published a catalogue of GWAS that provides a publicly available manually curated collection of published GWAS assaying at least 100000 SNPs

and all SNP-trait associations with $p \leq 1 \times 10^{-5}$. Up to December 2013, this catalogue included almost 2000 curated publications for more than 12000 SNPs [150]. A screen-shot of this interactive catalogue grouping 17 GWA trait categories is shown in Figure 5.8.

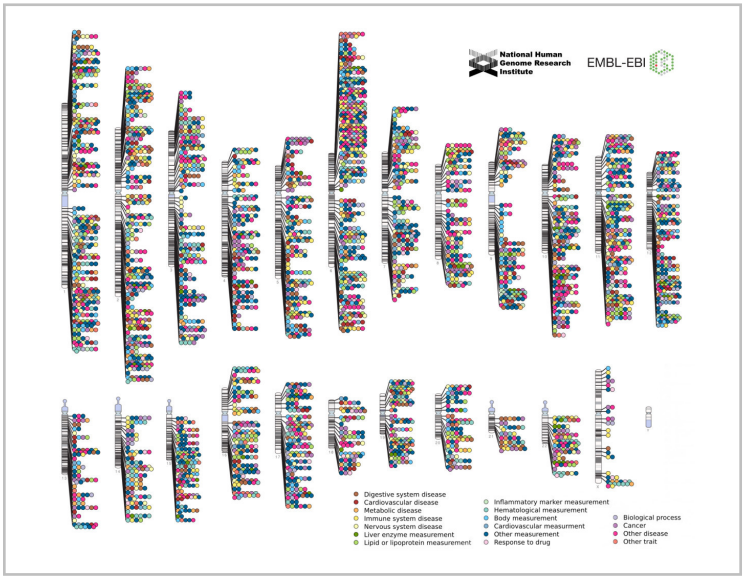


Figure 5.8: *Published GWAs at $p \leq 5 \times 10^{-8}$ for 17 trait categories.* Published Genome-Wide Associations, adapted from [151].

5.4.1. GWA Testing using Imputed Data

The probabilistic nature of imputed SNPs means that association analysis for these SNPs requires some care, regardless of what software or reference sets are used to generate the imputed data [85, 152].

While genotype platforms usually produce exact genotype calls (i.e. each individual is assigned genotype AA, AB, or BB –*coded as 0, 1 or 2*–), imputation programs generate probabilities for each of the three possible genotypes [152]. Using just those imputed genotypes with posterior probability above some threshold (or using the best guess genotype) is a reasonable method of comparing the accuracy across methods, but it is not recommended when carrying out association tests at imputed SNPs. Removing genotypes in this way can lead to both false positives and loss

of power [85]. Luckily, many GWA packages can analyze the genotype dosages (the expected number of copies of a specified allele, from 0 to 2) that are produced by imputation programs [152].

5.4.1.1. GWA Testing programs

Several GWA testing tools can be applied to conduct analysis of imputed SNPs using the corresponding posterior probabilities. These can provide additional insights beyond what is provided by testing on typed tagging SNPs only. For this reason, numerous stand-alone packages as BEAGLE [125], BIMBAM [113], MACH2dat [122], ProbABEL [153], PLINK [130], SNPMStat [131] and SNPTEST [154] have been proposed.

Despite the amount of GWAS performed using imputed data, only a few reviews [85, 155] carry out a comprehensive comparison of specific GWA testing programs for imputed data. Since the performance of these programs are affected by a variety of genetic factors, *Pei et al.* [155] investigated through a comprehensive comparison of these methods the effects, for example, of LD, *Minor Allele Frequency* (MAF) of untyped causal SNPs, and imputation accuracy rate. Hence, as part of these results, Table 5.5 and Table 5.6 were generated.

Table 5.5 shows ***Type-I error*** rates of various imputation-based association methods for the causal SNP at the significant level of 5%.

	Quantitative Trait			Qualitative Trait		
	Low LD	Mid LD	High LD	Low LD	Mid LD	High LD
SNPTEST	5.0	5.0	4.8	5.0	5.0	4.8
SNPTEST-BG	5.0	5.0	5.1	5.0	5.1	5.1
MACH2qtl/dat	5.0	5.0	5.0	4.9	4.9	5.0
BIMBAM	5.0	4.8	5.0	5.0	4.8	5.0
BEAGLE	-	-	-	4.9	5.1	4.9
PLINK	-	-	-	4.4	4.3	4.4
ProbABEL	5.1	5.1	5.0	5.1	5.2	5.1
SNPMStat	-	-	-	7.0	6.0	4.7

Table 5.5: GWA imputation-based tools comparison: type-I error rates [155].

From Table 5.5, it can be seen that, when testing association at the imputed potential causal SNP, all tools had type-I error rates close

to 5%. When testing for an entire genomic region, all programs but SNPMStat continue to have reasonable error rates, whereas SNPMStat had an inflated type-I error rate under low LD level. However, when testing for the whole region under high LD level, all methods were conservative.

On the other hand, Table 5.6 depicts the ***accuracy*** of the GWA imputation-based methods for testing a whole genomic region under the significant level of 5%.

	Quantitative Trait			Qualitative Trait		
	Low LD	Mid LD	High LD	Low LD	Mid LD	High LD
SNPTEST-BG	49.6	52.8	65.5	50.1	52.3	62.2
SNPTEST	49.9	54.2	67.2	50.1	53.2	62.4
MACH2qtl/2dat	50.4	54.0	66.3	50.0	53.7	62.7
BEAGLE	-	-	-	50.3	51.3	61.9
PLINK	-	-	-	50.3	51.2	56.9
ProbABEL	50.7	53.9	66.6	50.1	52.7	62.9
SNPMStat	-	-	-	50.5	51.2	59.0

Table 5.6: GWA imputation-based tools comparison: accuracy [155].

For both quantitative and qualitative traits, MACH2qtl/dat, ProbABEL and SNPTEST had the best performance under most situations, followed by SNPTEST-BG. BEAGLE program had similar performance to SNPTEST-BG under high LD level, but was inferior under medium LD level. SNPMStat and PLINK had the lowest power. As BIMBAM estimated p-value through permutation with 1000 replicates, its output had a resolution 1×10^{-3}, which did not reach the significant level (2×10^{-4}) with Bonferroni correction. For this reason, BIMBAM was not included in the analysis. Note that for all developed analysis, SNPTEST-BG utilized the *best-guess* genotype method, while SNPTEST considered the uncertainty and took the posterior probability into analyses [155].

PART

III

Approaches for Population Data

This part of the thesis is motivated by the problem of identification of genetic variants associated with advanced diabetic nephropathy in a T2D population from the Gran Canaria island. For this purpose, we analysed a dataset from a case-control study where more than 2.5 million of SNPs were genotyped. The objective was to identify which SNPs could be associated with the time since diagnosis of diabetic nephropathy. Both cases and controls were individuals with several years of T2D progress. Cases were selected after nephropathy development and controls were individuals of similar characteristics (age, sex, history of diabetes, etc.) but not affected by nephropathy. Once these SNPs were identified, a new sample of subjects will be genotyped in those regions where such SNPs were located to confirm the association and identify related genes to the disease. This part will describe diverse processes that have to be carried out to this aim: quality control, alignment and phasing, imputation, GWAS testing, as well as it will be explained obtained results and selection of candidate SNPs. These tasks have required an extensive use of complex bioinformatics resources. In Appendix A, we have included a tutorial that synthesizes the different steps of the analysis which we expect that can be useful to others facing the same problems.

Chapter 6

Quality Control

6.1. Introduction

Quality control (QC) is a critical stage previous to applying association tests to SNP data. Therefore, the need for performing QC lies on several reasons. Firstly, since hundreds of thousands or million of genotypes may be properly generated, sometimes genotyping errors appear (in a small proportion), which if unidentified, may lead to spurious GWAS results [156] or in false positive association errors. Secondly, the quality of genotype calling usually is lower in practice than in the manufacturer's panels. This is due to real samples normally are not such carefully prepared as the ones used to benchmark the panels. Finally, some developments such as custom SNPs panels, separately-genotyped reference control cohorts, and combined analyses of separate studies, may all increase the need for a QC phase [157].

In summary, QC can be differentiated by two aspects: **issues related to genotype calls** (from genotyping chips) and **downstream issues** [156]. This part of the dissertation will exclusively focus the last (i.e. those procedures to be applied once genotype calling is already performed), covering both **subject-based** and **variant-based** quality measures. These measures will include QC on individuals (samples) and variants (markers).

A flowchart overview of the entire QC process is depicted in Figure 6.1. Each QC aspect is detailed along the corresponding section of this chapter.

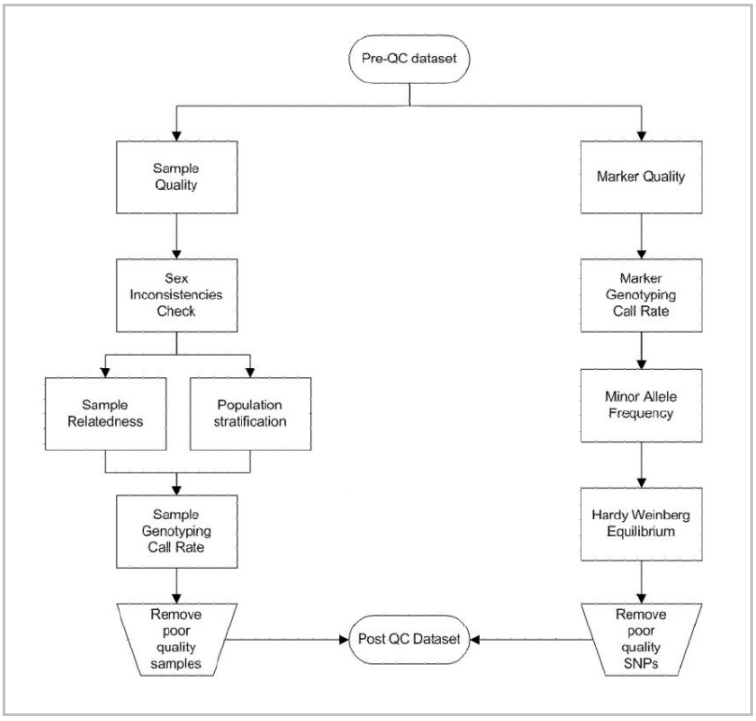


Figure 6.1: *Workflow scheme of the QC process. Squares represent steps, ovals symbolizes input or output data, and trapezoids represent the filtering of data.* Flowchart of the Quality Control process, adapted from [56].

6.1.1. Data Format

The population data used in this part of the dissertation are binary PED files (*.bed). The PED file format is specified in subsection B.1.1 of Appendix B. Saving space and time, the BED files store the pedigree/phenotype information in three separate files: ***.bim**, ***.fam** and ***.ped**. BIM files are extended MAP files, containing information about the allele names, FAM files comprised all the pedigree/phenotype information of subjects and BED files include the genetic information but compressed in binary format.

The FAM and BIM files are plain text files which can be viewed with a standard text editor. However, the PED files are compressed files that can not be previewed with standard tools.

So, the FAM file for first seven individuals would be:

famID	indID	patID	matID	sex	phenotype
4	4	0	0	2	2
12	12	0	0	2	2
13	13	0	0	1	2
14	14	0	0	1	2
17	17	0	0	1	2
22	22	0	0	1	2
24	24	0	0	2	2

Where the column names are: ***famID*** (family identifier), ***indID*** (individual identifier), ***patID*** (paternal identifier), ***matID*** (maternal identifier), ***sex*** (1=male, 2=female, other=unknown) and ***phenotype*** (1=unaffected, 2=affected, 0=missing).

An extract of the BIM file for seven markers would be:

chr	markerID	distance	position	refAl	altAl
1	kgp15236529	0	850528	G	A
1	kgp12109133	0	850780	C	T
1	kgp3324955	0	851190	A	G
1	kgp575483	0	851390	T	G
1	kgp11435978	0	852875	C	T
1	kgp5226443	0	852964	T	G
1	rs7537756	0	854250	G	A

Where the column names are: ***chr*** (chromosome), ***markerID*** (marker identifier), ***distance*** (genetic distance in Morgans), ***position*** (base-pair position in bp units), ***refAl*** (reference allele) and ***altAl*** (alternate allele).

6.1.2. Data Summary

Summary of the raw data contained in our study, where QC has been applied:

- 2391739 markers to be included
- 112 individuals read
- 112 founders and 0 non-founders found
- 110 individuals with non-missing phenotypes
- 55 cases, 55 controls and 2 missing
- 51 males, 61 females, and 0 of unspecified sex

6.2. Marker Quality Measures

The **Marker-based Measures** that we have considered are: ***genotyping efficiency*** (proportion of failed genotyped calls for a marker), ***MAF*** (frequency of the alternate alleles) and ***HWE*** (allele and genotype frequencies in a population remain constant from one generation to the next).

6.2.1. Genotyping Efficiency

The ***genotyping efficiency*** or call rate (also known as 1-missingness rate) is defined as the proportion of samples with a genotype call for each marker. It may be an indicator of good genotyping quality. Classically, a typical threshold of 95% efficiency is usually determined [158]. Although each threshold may vary depending on the study and genotyping platform used. In this manner, markers should be removed based on the specified threshold by using the `-geno` option of PLINK [130]. In our case, according to figure 6.2, we used 0.1 as a limit of missingness, where the SNPs above that threshold were removed, i.e. those SNPs with less than 10% missing rate (or more than a 90% call rate) were kept.

Genotyping efficiency and *sample missingness rate* (further detailed in section 6.3.1) can be checked using the `-missing` option of PLINK [130], which will produce two files: ***.imiss*** and ***.lmiss***. The *.lmiss file stores the missingness rate for each locus as follows:

CHR	SNP	N_MISS	N_GENO	F_MISS
1	kgp15236529	29	112	0.2589
1	kgp12109133	0	112	0
1	kgp3324955	0	112	0
1	kgp575483	0	112	0
1	kgp11435978	0	112	0
1	kgp5226443	2	112	0.01786
1	rs7537756	22	112	0.1964
				..

Where the column names are: ***CHR*** (chromosome), ***SNP*** (SNP or locus identifier), ***N_MISS*** (number of individuals missing in this locus), ***N_GENO*** (number of non-obligatory missing genotypes) and ***F_MISS*** (proportion of individuals missing for each locus or SNP).

The next plot was generated to investigate or analyze the genotyping efficiency (also known as SNP coverage):

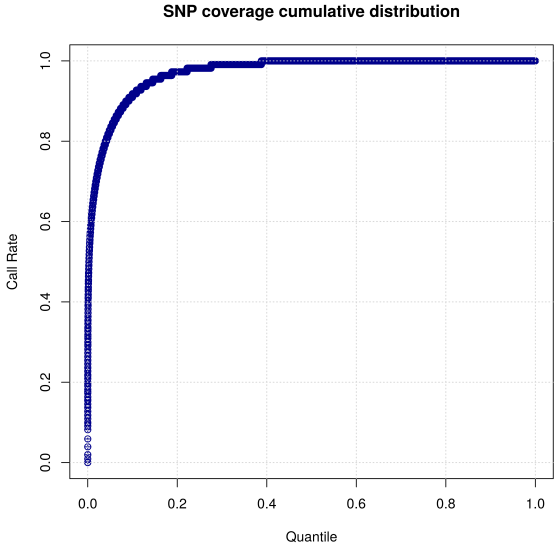


Figure 6.2: SNP coverage cumulative distribution

Finally, after the marker missingness analysis, a total **genotyping rate of 0.971871** for all individuals was obtained.

6.2.2. Minor Allele Frequency

Genetic associations with SNPs containing a low ***Minor Allele Frequency*** can give spurious results, causing a decrease in data quality, since there is a reduction of the power to detect an association signal, i.e. statistical power is extremely low for rare SNPs. This is mainly due to the fact that low MAFs imply rare genotypes which will be seen only a few times in the data. In addition, missingness can affect low-MAF SNPs more strongly and thereby increase the chances of a false positive signal [157]. Therefore, it makes no sense to include SNPs below a certain MAF in the analysis, because it will never be able to detect an association signal with them.

In our case, we have chosen a MAF threshold depending on the *number of subjects* (n), but not the typical relation $MAF = 10/n$, since our population is not large. Then, our MAF definition will be given by the following expression:

$$MAF = \frac{1}{n \times 2} = \frac{1}{110 \times 2} = 0.0045 \tag{6.1}$$

When this MAF threshold is used, all SNPs with *homozygous* alleles are excluded from our dataset since in later analysis (specifically in the association tests) these values may confound. So, at the end of this QC step, an amount of **484556 SNPs failed frequency test** from the total number of SNPs.

6.2.3. Hardy-Weinberg Equilibrium

The ***Hardy-Weinberg*** principle states that the genetic variation in a population will remain constant from one generation to the next in the absence of disturbing factors. When mating is random in a large population with no disruptive circumstances, this principle predicts that both genotype and allele frequencies will remain constant because they are in *equilibrium* [159].

In cases where the genotype distribution is different from what one would expect based on the allele frequencies, there may be several potential explanations: 1) it could be a genotyping error, 2) there was natural selection throughout different generations [156], 3) there was population stratification, and 4) even actual association to the trait under study [160].

In case-control data, it is customary to examine HWE p-values in control data only. This is because a strongly-associated causal SNP will lead to a departure from HWE in cases. In fact, this can happen in controls as well, but only if the disease is very common or the controls have been selected to exclude that disease (rather than just representing a random sample of the population at large) [161].

So, if we want to check the deviation from HWE in the controls for a case-control study, we can perform the analysis with the `--hardy` option of PLINK [130], thus generating a *Quantile-Quantile* (QQ) representation of the p-values of previous p-values we can appreciate if there is deviation from HWE for each SNP.

From Figure 6.3 we can note that a SNP whose corresponding p-value is considerably out of HWE. This variant (named as **rs114833138**) is placed at the *position 186204334 of the chromosome 4* and may indicate a genotyping problem.

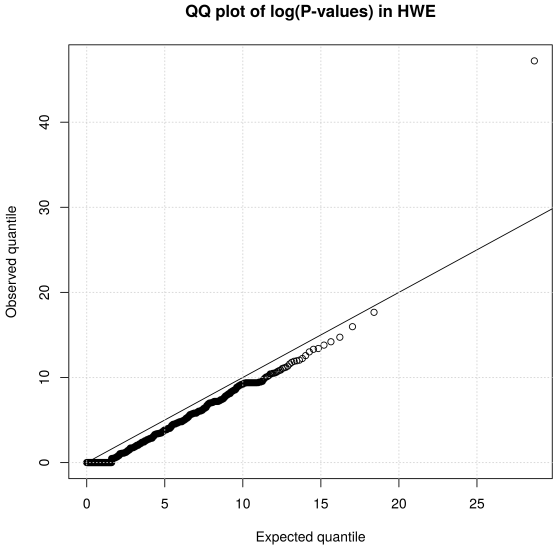


Figure 6.3: QQ plot of those log-p-values in HWE (for control samples).

6.3. Sample Quality Measures

The **Sample-based Measures** that we have taken into account are: ***Missingness Rate*** (proportion of missing genotypes per individual sample), ***Gender Mismatches*** (checks if self-reported gender matches genotyped gender), ***Population Stratification*** (individuals with a significantly different genetic background from rest of study sample), ***Heterozygosity Rate*** (proportion of heterozygous genotypes for a given individual) and ***Individual Relatedness*** (checks if subjects are close family members).

6.3.1. Missingness Rate

The ***missingness rate*** can be defined as the proportion of SNPs that failed per individual sample. Those samples with low genotyping efficiency should be dropped for later analysis. A high proportion of markers failing in a sample may be indicative of poor genotyping quality, which could lead to aberrant genotype calling [56].

To deepen in the sample missingness analysis, a threshold has to be specified through a balance between the maximization of genotyping efficiency and the minimization of the number of samples to remove. Thus, a sensible limit has to be placed where there is a qualitative change in data loss (as is depicted in Figure 6.4).

The **.imiss* file, generated by the `--missing` option of PLINK [130] (as mentioned in section 6.2.1), shows the missingness rate for each individual sample as follows:

FID	IID	MISS_PHENO	N_MISS	N_GENO	F_MISS
4	4	N	64370	2389506	0.02694
12	12	N	25846	2389506	0.01082
13	13	N	23048	2391739	0.009637
14	14	N	33146	2391739	0.01386
17	17	N	116259	2391739	0.04861
22	22	N	29320	2391739	0.01226
24	24	N	97750	2389506	0.04091
..					

Where the column names are: ***FID*** (family identifier), ***IID*** (individual identifier), ***MISS_PHENO*** (missing phenotype?), ***N_MISS*** (number of missing SNPs), ***N_GENO*** (number of non-obligatory missing genotypes) and ***F_MISS*** (proportion of missing SNPs).

Figure 6.4 shows the cumulative non-missingness distribution, in order to investigate the proportion of failed SNPs per sample, according to the *F_MISS* values of the **.imiss* file.

Based on Figure 6.4, from a call rate >91%, all individuals should have good genotyping quality. In the case of being more stringent (regarding call rate per sample), we could choose a threshold >97% using the PLINK [130] option `--mind` and a value of 0.03. In that case, we should eliminate 11 of 110 samples, i.e. the 10% of the total number of individuals in our study, which we thought highly not inadvisable.

6.3.2. Gender Mismatches

This QC step is performed to check that the ***gender of individuals*** (samples) matches with the corresponding number of X chromosomes.

Those subjects where the X-chromosome data disagrees with the reported gender are defined as *problematic* subjects. So, a *PROBLEM* arises if the two sexes do not match, or if the SNP data or pedigree data are ambiguous with regard to sex. A male call is made if F is more than 0.8 and a female call is made if F is less than 0.2 [130].

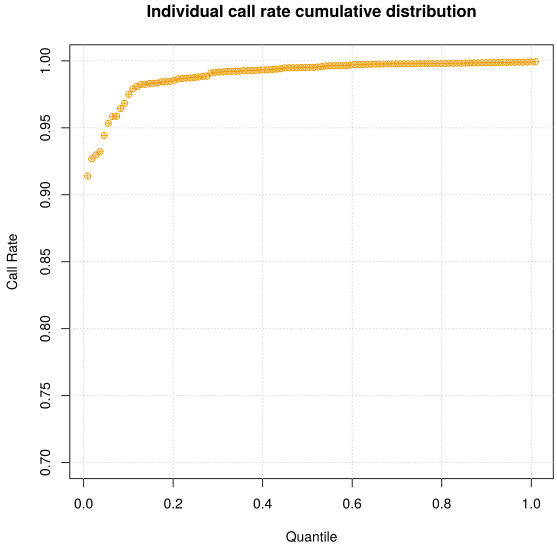


Figure 6.4: Individual Call Rate cumulative distribution.

The `--check-sex` option of PLINK [130] will generate a ***.sexcheck*** file containing the gender mismatches as follows (except the *Explanation* column):

FID	IID	PEDSEX	SNPSEX	STATUS	F	Explanation
35	35	1	1	OK	0.99	Male
41	41	1	1	OK	0.993	Male
45	45	2	2	OK	0.03151	Female
53	53	1	1	OK	0.9905	Male
55	55	1	1	OK	0.9938	Male
57	57	1	1	OK	0.9892	Male
60	60	2	0	PROBLEM	0.4347	Likely a female
61	61	2	2	OK	0.004779	Female
..						

Where field names are: ***FID*** (family identifier), ***IID*** (individual identifier), ***PEDSEX*** (sex as determined in pedigree file: 1=male, 2=female and 0=unknown), ***SNPSEX*** (sex as predicted based on genetic data -stored in X chromosome-), ***STATUS*** (displays "PROBLEM" or "OK" for each individual) and ***F*** (the actual X chromosome inbreeding -homozygosity- estimate).

If there are loads of mismatches, it can be assumed that all *Sample Identifiers* could have become scrambled in some way, but in our case, there is only one mismatch, so we have assumed that the majority of sample labels were allocated correctly between our clinical and genetic data.

6.3.3. Population Stratification

Population outliers (also known as *ethnic outliers*) occur when the study samples comprise multiple groups of individuals who differ systematically in both genetic ancestry and phenotype. Spurious apparent associations in admixed populations may be due to differences in ancestry rather than a true association of alleles to disease, leading to both false positives or false negatives [62]. So, although this is an important step of QC for population data, in our study it was not necessary to take this into account since the subjects were collected from the same population.

6.3.4. Individual Relatedness

Individual relatedness occurs when pairs or groups of subjects are more closely related to each other than the population average, thus indicating they are close family members [157]. Those individuals induce a correlation structure which may cause mistaken associations, i.e. they can introduce false positive or false negative results. Furthermore, this is particularly problematic if the number or degree of *cryptic relatedness* differs between cases and controls.

The `--genome` command of PLINK [130] generates a ****.genome*** file which can be used to estimate genetic relatedness for all pairs of samples in the dataset. The PI_HAT column of **.genome* file stores the *Identity-by-Descent* (IBD) estimates of the individuals. A PI_HAT value close to 1 indicates a sample duplicate or monozygotic twins, a value close to 0.5 indicates 1^{st} degree relatives (full sibs, parent-offspring), a value close to 0.25 indicates 2^{nd} degree relatives (half-sibs, uncle/aunt-nephew/niece, grandparent-grandchild), and a value close to 0.125 indicates 3^{rd} degree relatives (cousins, etc.) [161].

The applied criterion in our analysis to decide which sample would be dropped was to remove the one with the greater proportion of missing SNP data (extracted from the ****.imiss*** file, described in Section 6.3.1). So, according to Figure 6.6, two individual samples had to be removed.

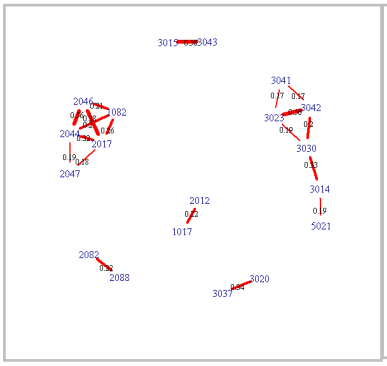


Figure 6.5: Example of more complex relatedness networks [161].

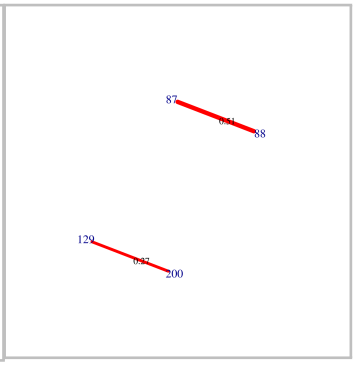


Figure 6.6: Relatedness networks in our study, where two cryptic relatednesses emerge.

Next figure comprises the IBD estimates histograms before and after the *Variant Quality Control* (VQC) and before *Sample Quality Control* (SQC).

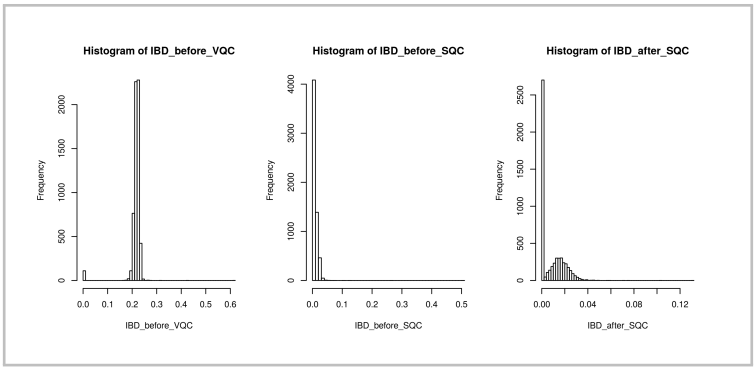


Figure 6.7: Comparison between IBD estimates histograms.

6.3.5. Heterozygosity Rate

The ***heterozygosity rate (H)*** is the proportion of heterozygous genotypes for a given individual [158]. This proportion is predictable from *Hardy-Weinberg* expectations and the MAF at each SNP according to the *Wright's Inbreeding Coefficient (F)*, which becomes one minus the ob-

served number of heterozygotes in a population divided by its expected number of heterozygotes at HWE, i.e.:

$$F = 1 - \frac{O.HET}{E.HET} \tag{6.2}$$

Positive F indicates an excess of homozygotes (low heterozygosity), negative F indicates an excess of heterozygotes (high heterozygosity) [157]. High heterozygosity can also indicate sample contamination (i.e., a mixture of two or more DNAs, leading to more apparent heterozygotes. Low heterozygosity can indicate membership in a different population (the Wahlund effect [162]) or indeed could indicate inbreeding. Thereby, deviation from expected heterozygosity may indicate either low genotyping quality (efficiency) or relatedness of individuals [157], which can justify their later removal.

The F parameter can be estimated with the `--het` option of PLINK [130] (using the standard MLE for one locus, and then using a method-of-moments). This method is an unbiased procedure for combining information across loci that involve separate summations of the number of observed and expected homozygous genotypes at each locus [161].

Thereupon, if we represent the H and F values in the same figure we can compare if there are two individuals with unusually high F (indicating either a genotyping problem or that they come from a different population), and thus should be dropped.

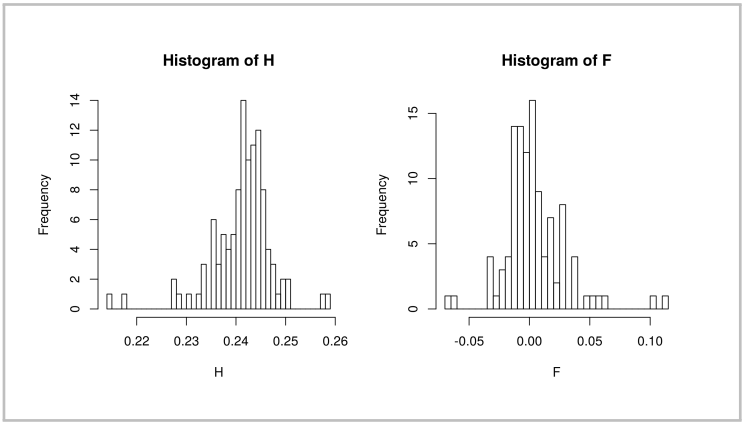


Figure 6.8: Histogram of Heterozygosity H, and an inversely related value F (before Sample-QC).

The heterozygosity histograms for before and after the Sample-QC in Figure 6.8 and Figure 6.9, respectively. In both figures some outliers can be noted, exceeding in three times the *Standard Deviation* (SD), which is the usual limit used to analyse heterozygosity [158]. Although such H,F values were out of the limits, we decided not remove their corresponding subjects, since this would unbalance the number of case-control samples.

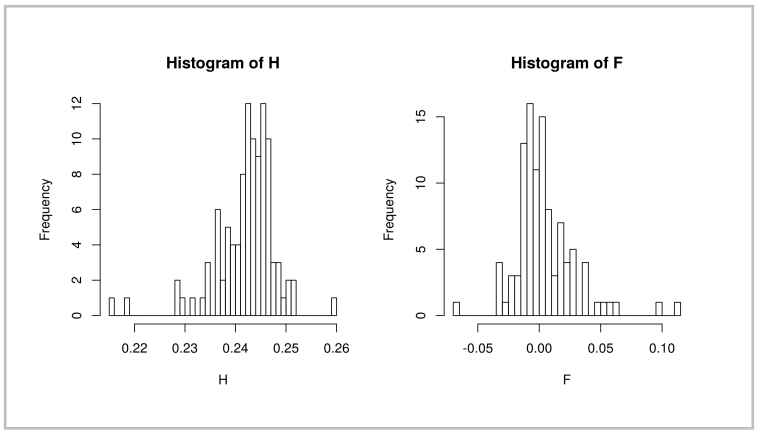


Figure 6.9: Histogram of Heterozygosity H, and an inversely related value F (after Sample-QC).

Chapter 7

Alignment and Phasing

7.1. Introduction

This chapter will detail the alignment and phasing of data once it has been passed the QC stage (previously explained in Chapter 6).

The main purpose of aligning and phasing the study genotypes is to get later a faster imputation from a large reference panel of haplotypes such as **HapMap** [95] or **1000 Genomes** [98] projects.

The procedure to implement at this stage is composed of two steps: 1) alignment of study samples with the reference panel and 2) estimation of haplotypes from genotype data (also known as phasing).

7.1.1. Data Format

Resulting data of QC stage have to be divided as many parts as chromosomes, to process later each part in a manageable file by the program SHAPEIT2 [141] (which will be used to perform both alignment and phasing).

Data formats used by SHAPEIT2 [141] are quite different, and they may be classified into three types: ***Input Data*** file (described in subsection 6.1.1 of Chapter 6), ***Genetic Map*** file (genetic map per chromosome) and ***Reference Panel*** files (haplotype text file format used by IMPUTE2 [121]).

7.1.1.1. Genetic Map File Description

Each ***genetic map file*** (per chromosome) should have the following structure:

Position	Combined_rate	Genetic_Map
10906723	0.6231151531	0.025416867094949
10906915	0.4976528486	0.0255124164418802
10906989	0.4965798052	0.025549163347465
10907208	0.4953800875	0.0256576515866275
10913973	0.442217683	0.0286492542121225
10916916	0.4231906694	0.0298947043521667
..		

Where the column names are: ***Position*** (physical position in base pairs 'bp'), ***Combined_rate*** (recombination rate in centiMorgans per Megabase 'cM/Mb'), and ***Genetic_Map*** (genetic position in centiMorgans 'cM').

7.1.1.2. Reference Panel Files Description

The ***reference panel files*** used by SHAPEIT2 [141] have the default text file format that can handle IMPUTE2 [121], namely the ***.sample**, ***.legend** and ***.hap** formats.

To describe them shortly, we can consider four unrelated individuals from EUR and AMR groups, specifically from three different populations, for which two haplotypes per individual are available at three markers (SNP_1, SNP_2, and SNP_3):

			Haplotype 1			Haplotype 2		
Sample	Group	Popul	SNP_1	SNP_2	SNP_3	SNP_1	SNP_2	SNP_3
Indiv1	EUR	IBS	A	T	A	A	C	T
Indiv2	EUR	IBS	G	C	T	A	T	A
Indiv3	EUR	TSI	A	T	T	A	C	T
Indiv4	AMR	MXN	G	T	T	G	C	T
..								

From the previous description, we can construct a very large file per chromosome, but this would be inefficient and computationally costly. For this reason, the adaptation into SAMPLE/LEGEND/HAP files is necessary, since SHAPEIT2 [141] program need to reduce the dimensions of each file to decrease the computing time. So, a brief explanation of each format is the following:

- The **SAMPLE file** describes the non-genetic information per individual sample:

```
Sample   Population  Group  Sex
Indiv1   IBS         EUR    male
Indiv2   IBS         EUR    female
Indiv3   TSI         EUR    female
Indiv4   MXN         AMR    female
..
```

  Where the column names represent: ***sample*** (individual identifier), ***Population*** (population identifier), ***Group*** (group inside the population), ***Sex*** (1=male, 2=female, 0=unknown).

- The **LEGEND file** describes the genetic information per single marker:

```
id       position   a0  a1
SNP_1    10906723   A   G
SNP_2    10906915   T   C
SNP_3    10906989   A   T
..
```

  Where the column names are: ***id*** (marker identifier), ***position*** (marker position), ***a0*** (main allele) and ***a1*** (alternate allele).

- The **HAP file** contains the haplotypes of the reference panel in binary format, where 0 stands for the main allele and 1 represents the alternate allele:

```
0 0 1 0 0 0 1 1
0 1 1 0 0 1 0 1
0 1 1 0 1 1 1 1
..
```

  Where each line corresponds to a SNP, storing the allele pairs for all individuals of the dataset. For example, the allele pair "0 1" means that the first haplotype carries the main allele while the second carries the alternate allele. Haplotypes are given in the same order than in the **.sample* file.

7.1.1.3. Phased Files Description

The outputs of SHAPEIT2 [141] may be directly read by IMPUTE2 [121] as long as they are in SAMPLE/HAPS format. Also, it is necessary to describe the VCF format, since the MINIMAC [105] program only accepts this kind of files as input. A concise description of such formats is the following:

- This **SAMPLE file** differs from the one described in previous subsections. It includes the non-genetic information per individual sample together with missing data proportion. Usually, it may also represent another kind of information as paternal/maternal identifiers, phenotypes, covariates, etc.

```
ID_1  ID_2  missing  cov1  cov2  cov3    pheno  bin
0     0     0        D     D     C       P      B
Unr1  Unr1  0.0005   2     1     0.0023  1.324  1
Unr2  Unr2  0.002    1     3     0.1253  0.856  0
Unr3  Unr3  0        2     2     −0.032  2.027  1
Unr4  Unr4  0.0031   1     4     0.0081  1.009  0
..
```

  Where the column names represent: ***ID_1*** (first individual identifier), ***ID_2*** (second individual identifier), ***missing*** (missing data proportion), ***cov1*** (first covariate, discrete value), ***cov2*** (second covariate, discrete), ***cov3*** (third covariate, continuous value), ***pheno*** (phenotype, continuous value) and ***bin*** (phenotype, binary value). *Note that the second line is also part of the header and describes the type of variables included in each column, existing four types of variables: D (discrete covariate), C (continuous covariate), P (continuous phenotype) and B (Binary Phenotype; 0=Controls, 1=Cases).*

- The **HAPS file** contains the SNPs information together with the haplotypes of the phased dataset:

```
21  SNP_1  10906723  A  G  0 0 1 0 0 0 1 1
21  SNP_2  10906915  T  C  0 1 1 0 0 1 0 1
21  SNP_3  10906989  A  T  0 1 1 0 1 1 1 1
..
```

  Where each line corresponds to a single SNP with the following columns: ***chromosome number***, ***marker identifier***, ***marker position***,

main allele, ***alternate allele*** and the rest of columns store the allele pairs which, shape the haplotypes for all individuals of the dataset.

- **VCF file** is a text format that contains meta-information lines (not described below), a header line, and then data lines each containing information about a position in the genome. A more detailed description can be found in [163].

```
CHR POS   ID     REF ALT QUAL FILTER INFO FORMAT Ind1 Ind2 Ind3
22  12345 SNP_1  A   G   .    PASS   .    GT     0|0  1|0  0|0
22  12346 SNP_2  T   C   .    PASS   .    GT     0|1  1|0  0|1
22  12347 SNP_3  A   T   .    q10    .    GT     0|1  1|0  1|1
22  12348 SNP_3  C   .   .    PASS   .    GT     0|0  0|0  0|0
22  12349 SNP_3  A   G,T .    s50    .    GT     2|2  2|1  2|2
..
```

  Where each column stands for: ***CHR*** (chromosome identifier), ***POS*** (marker position), ***ID*** (marker identifier), ***REF*** (main allele), ***REF*** (alternate allele), ***QUAL*** (phred-scaled quality score), ***FILTER*** (filter status, q10 = genotype quality is below 10%, s50 = samples with data is below 50%), ***INFO*** (additional information), ***FORMAT*** (format of genotype fields) and the rest of columns are the genotype fields. *Note that genotype values are encoded as alleles that can be separated by either of | or \, depending if they are already phased or not.*

7.2. Alignment

The ***alignment*** is a prephasing step that checks if a given dataset is well aligned with the reference panel of haplotypes. The *strand alignment* can be performed using PLINK [130], GTOOL [164] or SHAPEIT2 [141].

From these programs, we have decided to utilize SHAPEIT2 [141] to check the strand alignment (using the `-check` option). Then, once the study dataset has been divided per chromosomes (using the `-chr` option of PLINK [130]), the alignment files are generated. This is a major advantage since they can be easily integrated into the SHAPEIT2 [141] scripts to later perform the phasing process (using the `-exclude-snp` command).

Once there has been specified a reference panel (either **HapMap** [95] or **1000 Genomes** [98]), the program proceeds with several checks in order to make sure that alignment between the study samples and the

chosen panel is well done. When it encounters any problem, the following two files are generated:

- The ***.strand** file, which describes in detail the *errors* or problems found.
- The ***.strand.exclude** file, which simply lists the physical *positions* of those SNPs with problems found (for easy exclusion using the `-exclude-snp` option).

A content example of the **.strand* file is the following:

type	pos	main_id	main_A	main_B	main_flip	ref_id	ref_A	ref_B	ref_flip
Missing	10761587	kgp4096519	C	C	1	NA	NA	NA	1
Missing	10766457	kgp242945	T	T	1	NA	NA	NA	1
Missing	10773140	kgp6657356	T	T	1	NA	NA	NA	1
Strand	10776405	kgp4463422	G	G	1	rs201941533	G	C	0
Strand	10778454	kgp8477819	A	A	1	rs201320074	A	C	0
Missing	10778712	kgp22784830	T	T	1	NA	NA	NA	1
Missing	10780816	kgp10770004	T	T	1	NA	NA	NA	1
Missing	10783741	kgp13094653	G	G	1	NA	NA	NA	1
..									

Where the columns are: ***type*** (type of the alignment problem), ***pos*** (physical position of the SNP that has an alignment problem), ***main_id*** (SNP identifier in the study sample), ***main_A*** (first allele in the study sample), ***main_B*** (second allele in the study sample), ***ref_id*** (marker identifier in the reference panel), ***ref_A*** (first allele in the reference panel) and ***ref_B*** (second allele in the reference panel).

Therefore, in the SNP alignment process, alignment errors has been classified in the following manner:

1. SNPs from the study sample that are not in the reference panel, since they will not be able to use as a reference in the imputation process (**missing error**).
2. SNPs that do not match their alleles between study samples and the reference panel (**strand error**).

So, once generated the corresponding **.strand.exclude* file, we can easily distinguish the type of error associated with each SNP, just observing the listed identifiers.

7.3. Phasing

The ***phasing*** process (also known as haplotype estimation) can be defined as the identification of those haplotypes that are present in the study genotypes. This requires assigning each one of the SNP alleles to its corresponding haplotype. Once the SNPs of the study samples have been correctly aligned, the study haplotypes are identified by comparing with their corresponding ones in the reference panel. This is a complex and computationally costly task that is implemented through HMM algorithms.

As phasing program, we have used SHAPEIT2 [141] which generates as output a pair of SAMPLE/HAPS files per chromosome, being the haplotype pairs with greater probability stored in the **.haps* file (described in subsection 7.1.1.3). According to *Delaneau et al.* [165], it is strongly recommended to phase each chromosome in a single run instead of making chunks.

On the other hand, using the IMPUTE2 [121] program, several pairs of *possible* haplotypes per individual may also be displayed as output. Each one of them will store the estimated probability, albeit at a higher computational cost.

7.3.1. Phasing Method Description

The phasing method proposed in *Delaneau et al.* [166] improves the state-of-the-art HMM algorithms by implementing this two enhancements:

1. Collapsing of all K haplotypes in H into a graph structure H_g, to perform later the HMM calculations on such graph.
2. Sampling a pair of compatible haplotypes for G with a new method that has linear complexity $O(MJ)$, in the number of conditioning states J.

By the first improvement, the haplotypes are partitioned into chunks each with J different haplotypes. The principal state space has J states for each marker, weighted by the number of haplotypes [141]. Regarding the second enhancement, it has much lower complexity than the quadratic complexity $O(MN2)$ of algorithms used by IMPUTE2 [121] and MaCH [122], leading to faster executions.

Each iteration of the phasing algorithm is based on updating all the genotypes of the sample in random order by using the update step

described above. This iteration scheme forms a Gibbs sampler. Between each iteration, a new segmentation of the current haplotype estimates is found to build a new *Compact Hidden Markov Model* (CHMM) that contains, on average, J haplotypes by segment.

According to *Delaneau et al.* [140], the algorithm starts from random haplotype estimates for the sample and performs three stages of iterations.

i. A first stage of iterations is run to find a better starting point. These are called burn-in iterations.
ii. The second stage of iterations is then run when the transition probabilities of the graphs S_g are averaged. Then, the lowest transitions probabilities are pruned and the segments that become poorly connected are concatenated. The above results in more parsimonious graphs S_g that contain lower numbers of haplotype segments, thus further increasing computational efficiency. This can be referred as a pruning step.
iii. Another stage with a larger number of iterations (called main iterations) are used to obtain a final estimate of the transition probabilities in S_g and thus compact representations of the uncertainty of phasing in the sample.

These compact representations can be used in several ways. First, the most likely haplotypes for a genotype can be estimated by applying the Viterbi algorithm in the associated graph S_g. Second, sets of haplotypes for a genotype can be sampled by the forward algorithm described before. And third, the plausible haplotypes defined just over a few closely located SNPs can be extracted by appropriate marginalization [140].

Then, the basic phasing operation implemented by *Delaneau et al.* [140, 141] is: 1) the vector G is divided into segments, 2) all haplotypes compatible with G are enumerated in each segment and 3) the haplotypes in each segment are represented as nodes of a graph S_g.

Figure 7.1 shows a set $K = 8$ haplotypes in H and an individual's genotype G, both defined over $M = 8$ markers. H_g is built by splitting the haplotypes of H between marker 4 and 5, resulting in two segments that contains each $J = 3$ distinct haplotypes. The nodes of the graph are represented by filled squares for allele 1 and empty squares for allele 0. Each edge is weighted by the number of haplotypes in H that traverse it. A haplotype of H and its corresponding path in H_g is illustrated in magenta. S_g is built by making two segments of 5 and 3 mark-

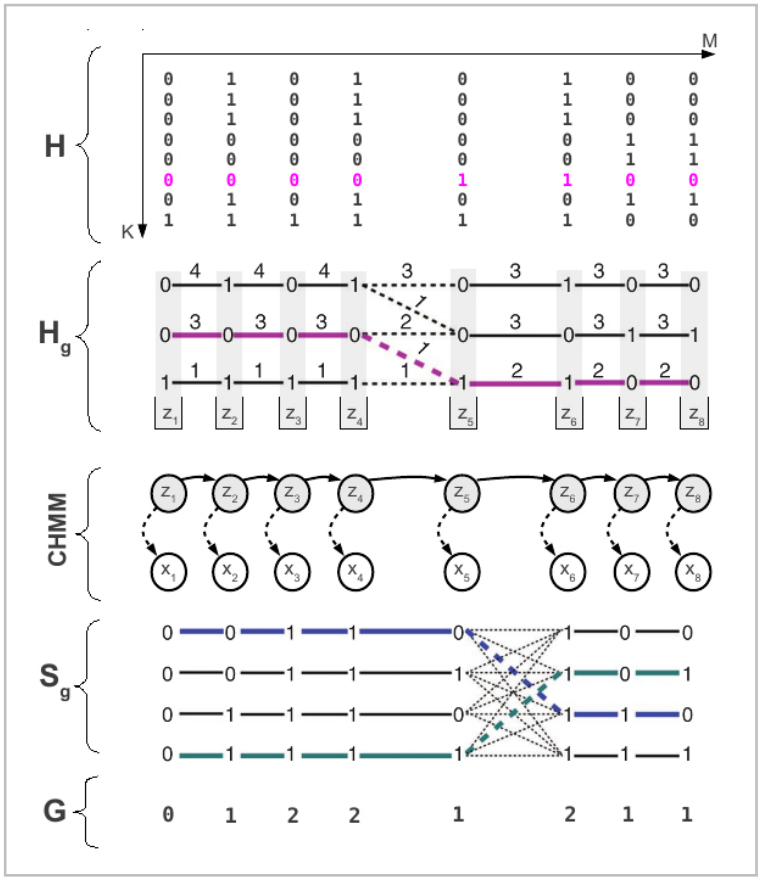


Figure 7.1: *Illustration of phasing model and its associated graphs on a simple example.* SHAPEIT2 method example, adapted from [140].

ers, each one containing two heterozygous markers in G (represented as state 1; state 0 and 2 are wild type and homozygous, respectively). Each segment has four possible haplotypes compatible with G. A pair of paths in S_g compatible with G is colored blue and green. In the middle of the image, CHMM hidden and observed states are denoted respectively $\{z_1, ..., z_8\}$ and $\{x_1, ..., x_8\}$, where plain arrows represent transition probabilities between hidden states and dashed arrows represent emission probabilities [140].

Chapter 8

Imputation

8.1. Introduction

The main purpose of the **imputation** process is estimating unobserved genotypes in study samples by the extrapolation of the allelic correlations from a reference panel. This task must necessarily include a dataset composed of the study samples (genotyped at a subset of SNPs) and a reference panel (genotyped at a denser set of SNPs). It can be carried out by different programs, such as MaCH [122], BIMBAM [113], BEAGLE [108], MINIMAC3 [124] or IMPUTE2 [121].

After analyzing the state of the art (see Chapter 5), we have decided to use the last two, because of their good relationship performance/precision. In both cases, a list of the imputed alleles (with the corresponding probability distribution on them (in ***.gen** and **.vcf* formats) can be obtained as output.

In this chapter, data format and methods implemented by the corresponding imputation programs will be described.

8.1.1. Data format

Data formats used throughout the *imputation* stage may be classified into three types: ***Input Data*** file (described in subsection 6.1.1 of Chapter 6), ***Genetic Map*** file (genetic map per chromosome) and ***Reference Panel*** files (haplotype text file format used by IMPUTE2 [121]).

8.1.1.1. Imputed Files Description

The default imputed files of IMPUTE2 [121] and MINIMAC3 [124] programs are ***.gen** format and the ***.dose.vcf** format, respectively. The latter is a modified version of the *.vcf format previously described in Section 7.1.1.3. A brief description of both is the following:

- The **GEN** file contains the SNPs information together with the haplotypes of the phased dataset. In this format, the SNPs are clearly identifiable from the study samples (respect to those from the reference panel). So, being the study SNPs labelled as SNP_01, SNP_02, SNP_03, etc. and the panel SNPs as rs0031, rs0032, rs0033, etc., we would get the following genotypes for 2 individuals:

Panel SNPs	**Study SNPs**
rs0031: AA AA	SNP_01: AA AA
rs0032: GG GC	SNP_02: GG GG
rs0033: CC CC	SNP_03: GT GC
rs0034: CC TT	SNP_04: CC GG
⋮	⋮

  The corresponding **.gen* file would be:

```
21 rs0031 10979170 A G 1 0 0 1 0 0
21 rs0032 10979323 G C 1 0 0 0 1 0
21 SNP_01 10979896 T C 1 0 0 1 0 0
21 rs0033 10979913 C T 0.960 0.040 0 0.986 0.014 0
21 rs0034 10980199 C T 1 0 0 0 0 1
..
21 SNP_02 14622336 G A 1 0 0 1 0 0
21 rs0191 14622684 C T 0 0.984 0.016 0 0.986 0.014
21 rs0192 14622862 G A 1 0 0 0 1 0
..
```

  Where each line corresponds to a single SNP with the following columns: ***chromosome number***, ***marker identifier***, ***marker position***, ***main allele***, ***alternate allele*** and the rest of columns store the three probabilities for each individual corresponding to the genotypes AA, AB and BB respectively.

- The default format of MINIMAC3 [124] is the **VCF file** although it can output files in both **.vcf* and **.dose* formats, being the last one the usual MINIMAC [105] output format. In order to facilitate the posterior statistical analysis, the probabilities of the im-

puted alleles by setting the `-format` command with the *GT,GP* labels, where each one means:

- **GT**: Estimated most likely genotype.
- **GP**: Estimated posterior genotype probabilities.

Then, this format is the same that the one described in Section 7.1.1.3, but including posterior genotype probabilities (GP) in the individual columns, namely those columns would look like as follows:

```
.. FORMAT Ind1                    Ind2
   GT:GP  0|0:1.000,0.000,0.000   1|0:1.000,0.000,0.000
   GT:GP  0|1:0.002,0.000,0.998   1|0:0.992,0.000,0.008
   GT:GP  0|1:0.000,0.000,1.000   1|0:1.000,0.000,0.000
   GT:GP  0|0:1.000,0.000,0.000   0|0:1.000,0.000,0.000
   GT:GP  2|2:0.410,0.550,0.040   2|1:0.882,0.110,0.008
..
```

Where ***Ind1***, ***Ind2*** columns now include the three probabilities for each individual (one per genotype 'AA, AB or BB').

8.2. Imputation

Once the SNP alleles of the study samples have been phased (previously described in Chapter 7), the imputation stage starts. If a haplotype composed by k markers is identified in a study sample, normally it will be possible to find several haplotypes of length $K > k$ in the reference panel that match with the study haplotype in the k considered makers. Genotypes present in the remaining $K - k$ markers will be imputed to the study haplotype. Obviously, there is no single allocation, but it must be conducted in a probabilistic way, according to the observed frequencies in the reference panel in $K - k$ imputed markers.

Most HMM-based imputation methods infer the missing genotypes for each sample independently, depending on the reference panel. In this way, the reference data is used to integrate the phase uncertainty in each individual's genotype. All imputation methods normally run *Markov Chain* algorithms, such as *Markov Chain Monte Carlo* (MCMC) or HMM, which are essentially based on this two steps:

1. Imputation of any sporadic missing genotype at the study SNPs, once all the observed genotypes have been phased.
2. Use of the reference panel to impute missing alleles at untyped SNPs.

Imputation programs repeat the previous steps over multiple iterations, averaging the imputation probabilities inferred for each missing genotype and also integrating the uncertainty phase.

8.2.1. Imputation Method Description

The imputation method proposed by *Marchini et al.* [112] employs a HMM network to compare the set of genotypes for each study sample to the reference panel in order to solve the untyped SNPs. This HMM method contains ***hidden states*** as haplotype backgrounds or SNPs loci (points in time). The movement between these states is a Markov Chain, being the probability of belonging to state j (where $j = \{1..k\}$ at time i depends on time $i - 1$). The k states are the k possible haplotype backgrounds of the reference panel, being the probability of moving between states based on recombination rates. In a state j at point i, the ***emission probabilities*** of the HMM will determine what an outcome could be (an allele, in this case).

A detailed explanation of how such imputation is performed starts with the description of the following variables:

- $H = \{H_1, \ldots, H_N\}$ is the reference haplotypes set, which is formed by N distinct haplotypes.
- L represent the number of loci placed in the SNPs of interest.
- $H_i = (H_{i1}, \ldots, H_{iL})$ is i^{th} haplotype, being $H_{ij} \in \{0, 1\}$.
- K are the number of individuals of the study.
- G_i is the genotype of the i^{th} subject, where $G_i = (G_{i1}, G_{i2}, \ldots, G_{iL})$ with $G_{ij} \in \{0, 1, 2, missing\}$.
- $G = \{G_1, \ldots, G_K\}$ represents the set of genotypes of K subjects in the study.

For each subject, the corresponding genotype G_i is distributed as $\{G_{iO}, G_{iM}\}$ being G_{iO} the observed genotyped and G_{iM} the missing genotype (not observed in the study samples, but available in reference haplotypes). We can then consider $G_O = \{G_{1O}, G_{2O}, \ldots, G_{KO}\}$ as the observed genotype set for K subjects and $G_M = \{G_{1M}, G_{2M}, \ldots, G_{KM}\}$ as the missing genotype set.

To carry out the G_M imputation, we must first calculate the probability distribution of G_M. Obviously, this will depend on the reference haplotypes and the observed genotypes per individual. Therefore, we have:

$$P\left(G_M = g_M \,|G_O = g_O, H\right) =$$
$$P\left(G_{1M} = g_{1M}, \ldots, G_{KM} = g_{KM} \,|G_{1O} = g_{1O}, \ldots G_{KO} = g_{KO}, H\right) =$$
$$\prod_{i=1}^{K} P\left(G_{iM} = g_{iM} \,|G_{iO} = g_{iO}, H\right) \tag{8.1}$$

Where we have supposed that the study subjects are independent each other. Thus, from the basic rules of conditional probability, we know that the *joint probability* is the product of the probabilities:

$$P\left(A \,|B \cap C\right) = \frac{P\left(A \cap B \cap C\right)}{P\left(B \cap C\right)} =$$
$$= \frac{P\left(A \cap B \cap C\right)}{P\left(C\right)} \cdot \frac{P\left(C\right)}{P\left(B \cap C\right)} = \frac{P\left(A \cap B \,|C\right)}{P\left(B \,|C\right)} \tag{8.2}$$

Being $P\left(A \,|B \cap C\right)$ proportional to $P\left(A \cap B \,|C\right)$, which can be written as $P\left(A \,|B \cap C\right) \propto P\left(A \cap B \,|C\right)$. According to our case, this rule results:

$$P\left(G_M = g_M \,|G_O = g_O, H\right) \propto \prod_{i=1}^{K} P\left(G_{iM} = g_{iM}, G_{iO} = g_{iO} \,|H\right) =$$
$$\prod_{i=1}^{K} P\left(G_i = g_i \,|H\right) \tag{8.3}$$

Now, for calculating the probabilities $P\left(G_i = g_i \,|H\right)$, a HMM model is used, where the hidden states are $\left(Z_i^{(1)}, Z_i^{(2)}\right)$ haplotype pairs formed from the H set, namely $Z_i^{(j)} = \left(Z_{i1}^{(j)}, Z_{i2}^{(j)}, \ldots, Z_{iL}^{(j)}\right)$ and $Z_{il}^{(j)} \in \{1, ..., N\}$. Furthermore, the $Z_{il}^{(j)}$ value indicates from which haplotype comes the j (where $j = \{1, 2\}$) of l locus.

Note that previous definition means that $Z_i^{(j)}$ may not exactly match with an H haplotype, but consist of linked fragments from such haplotypes (namely, $Z_i^{(j)}$ would become a new haplotype that is obtained by recombining those found in H). The HMM model, therefore, implies that:

$$P\left(G_i\,|H\right) = \sum_{Z_i^{(1)},Z_i^{(2)}} P\left(G_i\,\middle|Z_i^{(1)},Z_i^{(2)},H\right)\cdot P\left(Z_i^{(1)},Z_i^{(2)}\,|H\right) \quad (8.4)$$

For each locus, the hidden states can then be considered as the haplotype pairs of H set which are copied at that locus to form later the G_i vector of genotypes.

The $P\left(Z_i^{(1)},Z_i^{(2)}\,|H\right)$ term defines the *prior probability* of changing $Z_i^{(1)}$ and $Z_i^{(2)}$ along the i^{th} subject. Thus, starting from the first locus, the $\left(Z_{i1}^{(1)},Z_{i1}^{(2)}\right)$ initial state would have the probability:

$$P\left(Z_{i1}^{(1)},Z_{i1}^{(2)}\,|H\right) = \frac{1}{N^2} \quad (8.5)$$

Where the ***transition probabilities*** would be:

$$P\left(\left\{Z_{il}^{(1)},Z_{il}^{(2)}\right\}\rightarrow\left\{Z_{i(l+1)}^{(1)},Z_{i(l+1)}^{(2)}\right\}|H\right) =$$
$$\begin{cases}\left(e^{-\frac{\rho_l}{N}}+\frac{1-e^{-\frac{\rho_l}{N}}}{N}\right)^2 & \textit{No recombination}\\ \left(e^{-\frac{\rho_l}{N}}+\frac{1-e^{-\frac{\rho_l}{N}}}{N}\right)\left(\frac{1-e^{-\frac{\rho_l}{N}}}{N}\right) & \textit{Recombination in a haplotype}\\ \left(\frac{1-e^{-\frac{\rho_l}{N}}}{N}\right) & \textit{Recombination in both haplotypes}\end{cases} \quad (8.6)$$

The ρ term in the previous expression depends on a estimation of the ***recombination rate*** throughout the genome. Such rates have been collected from HapMap project data in order to build a *recombination map* of the human genome [112].

Based on the initial state probability and such transition probabilities, the $P\left(Z_i^{(1)},Z_i^{(2)}\,|H\right)$ probability could calculated as:

$$P\left(Z_i^{(1)},Z_i^{(2)}\,|H\right) =$$
$$P\left(Z_{i1}^{(1)},Z_{i1}^{(2)}\,|H\right)\prod_{i=1}^{L-1}P\left(\left\{Z_{il}^{(1)},Z_{il}^{(2)}\right\}\rightarrow\left\{Z_{i(l+1)}^{(1)},Z_{i(l+1)}^{(2)}\right\}|H\right) \quad (8.7)$$

The $P\left(G_i\,\middle|Z_i^{(1)},Z_i^{(2)},H\right)$ term models how the observed genotypes are close, but not exactly the same as those that would result from

H haplotypes that are being copied. Somehow this expression mimics the effects of mutation. Assuming that mutations occur independently along loci, we have:

$$P\left(G_i \left| Z_i^{(1)}, Z_i^{(2)}, H\right.\right) = \prod_{l=1}^{L} P\left(G_{il} \left| Z_{il}^{(1)}, Z_{il}^{(2)}, H\right.\right) \tag{8.8}$$

The $P\left(G_{il} \left| Z_{il}^{(1)}, Z_{il}^{(2)}, H\right.\right)$ probabilities depend on the *mutation rate* are further described in [112].

A graphical description of the typical set up for an imputation study is depicted in Figure 8.1. The **blue data** are study genotypes, and the aim is to estimate genotypes up to the density of the reference haplotypes (**in black**). In some studies, however, the aim may be to impute only up to the density of some reference genotypes (**green data**). It is possible to have only haplotype reference data, genotype reference data or both. Sporadic missing data (***coded as '?'***, can also be imputed both in reference data and inference data genotypes) [167].

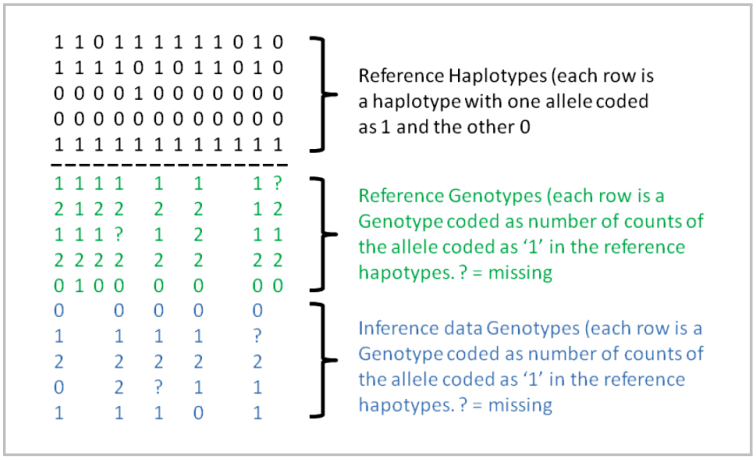


Figure 8.1: *Schematic drawing of a standard imputation scenario.* Example of a standard imputation, adapted from [167].

8.2.2. IMPUTE and MACH/MINIMAC programs

The program IMPUTE [112] was created to determine the probability distribution of missing genotypes conditional upon a set of known haplotypes and an estimated fine-scale recombination map (calculating the historical mutation through a theoretical result from population genetics theory). Later, a new version called IMPUTE2 [121] was released. This handles larger population genetic datasets and may use additional reference panels (or study samples) typed on different chips.

On the other hand, we have MACH/MINIMAC programs. The first of these set of programs is MACH [122], which imputes unobserved genotypes for case-control studies, employing a reference dataset and a HMM approach. The second set of algorithms, called MINIMAC, are more computationally efficient implementations of the MACH program, which were initially developed by [105], and later by [123] (MINIMAC2) and [124] (MINIMAC3).

Both IMPUTE and MACH/MINIMAC programs have *"best guess"* option for imputed genotypes (i.e. those with probability of 0.50 or greater), which is the simplest way to carry out the imputation.

IMPUTE2 [121] performs a single imputation step rather than running MCMC in this situation, and the main change on the command line is to replace the `-g` file (unphased genotypes) with a `-known_haps_g` file. The program imputes the untyped alleles in each phased haplotype, to later convert the corresponding allelic probabilities into diploid genotype probabilities [168].

Otherwise, MACH/MINIMAC programs output a file (**.dose* file) that provides an estimated dosage of each imputed genotype along with the dosage of observed genotypes. The dosage is a range between 0 and 2, where 0 represents no copies of the SNP reference allele, and 2 represents two copies of the reference allele. Therefore, while observed genotypes have discrete values included in [0 1 2], imputed genotypes are an estimate of the number of copies of the reference allele, represented by a decimal falling anywhere between 0 and 2. MACH [122] also outputs a file of posterior probabilities and a file with quality scoring for each SNP. The "best-guess" genotypes can be analyzed just as observed genotypes, with any standard statistical genetics software package, caution should be used in interpretation of results due to the uncertainty of the imputed data. The dosage file is particularly well-suited for analysis methods that take this into account, such as logistic and linear regression [167].

Chapter 9

GWA Testing of Imputed Data

9.1. Introduction

This chapter will describe the final stage of Part III of the dissertation: the **statistical analysis** (i.e. association tests) of previously imputed data.

During last years, imputed SNPs have been frequently used in GWAS analyzes, although, for a properly implementation of association tests, the uncertainty of such imputed data has to be taken into account.

We have chosen SNPTEST2 [154] as the program for analyzing single marker associations in GWAS, since it has implemented different methods for dealing with imputed SNPs, including the analysis of binary (case-control) phenotypes, single, and multiple quantitative phenotypes and Bayesian or Frequentist tests.

9.1.1. Data format

Input data formats used by the SNPTEST2 are the same as described in Section 8.1.1.1 (namely SAMPLE, GEN and VCF files).

The `-summary_stats_only` option of SNPTEST2 calculates the data summaries associated with each SNP (i.e. genotype counts, allele frequencies, missing data proportions and odds ratios), leading to an *output* file as follows:

```
id rsid chr pos allele\_A allele\_B index
average\_maximum\_posterior\_call info . .
21:10979170 rs0031 21 10979170 A G 1 1 1
```

```
21:10979323 rs0032 21 10979323 G C 2 1 1
21:10979896 SNP\_01 21 10979896 T C 3 1 1
21:10979913 rs0033 21 10979913 C T 4 1 0.960
21:14622336 SNP\_02 21 14622336 G A 5 1 1
..
```

Where the column names represent: ***id*** (marker identifier), ***rsid*** (rs id of the SNP), ***chr*** (chromosome number), ***pos*** (marker position), ***allele_A*** (main allele, coded as 0), ***allele_B*** (alternate allele, coded as 1), ***index*** (markers' index), ***average_maximum_posterior_call*** (average maximum posterior probability across all individuals) and ***info*** (measure of the observed statistical information at each SNP).

Tables 9.1 and 9.2 give a detailed of *additional columns* when a test for a binary phenotype is carried out:

Column Name	Description
cohort_1_AA **cohort_1_AB** **cohort_1_BB** **cohort_1_NULL**	Counts of AA, AB, BB and NULL genotypes in the 1^{st} cohort.
cohort_2_AA **cohort_2_AB** **cohort_2_BB** **cohort_2_NULL**	Counts of AA, AB, BB and NULL genotypes in the 2^{nd} cohort. Subsequent cohorts should be included similarly.
all_AA all_AB **all_BB** **all_NULL** **all_total**	Counts of AA, AB, BB and NULL thresholded genotypes, as well as the total number of samples considered, across all cohorts.
all_maf	MAFs in the combined controls, combined cases and combined across all cohorts.
missing_ data_proportion	The proportion of missing data across all cohorts.

Table 9.1: *Detailed description of the output file columns of SNPTEST2.* SNPTEST2 output file description (1 of 2), adapted from [154].

Furthermore, depending on the chosen test to perform the association analysis, more additional columns will appear at the output file.

Column Name	Description
controls_AA **controls_AB** **controls_BB** **controls_NULL**	Counts of AA, AB, BB and NULL genotypes across all control cohorts.
cases_AA **cases_AB** **cases_BB** **cases_NULL**	Counts of AA, AB, BB and NULL genotypes across all case cohorts.
cases_maf **controls_maf**	MAFs in the controls and cases across all cohorts.
het_OR **het_OR_lower** **het_OR_upper**	Estimated odds ratios and lower and upper 95% confidence limits for the heterozygote genotype AB versus the (baseline) AA genotype.
hom_OR **hom_OR_lower** **hom_OR_upper**	Estimated odds ratios and lower and upper 95% confidence limits for the homozygote genotype BB versus the (baseline) AA genotype.
all_OR **all_OR_lower** **all_OR_upper**	Estimated allelic odds ratios and lower and upper 95% confidence limits for the B allele versus the (baseline) A allele.

Table 9.2: *Detailed description of the additional fields included by SNPTEST2.* SNPTEST2 output file description (2 of 2), adapted from [154].

These will store the p-values (**pvalue**), the statistical information or info score (**info**), the standard error (se) and the coefficient β (**beta**) for each test. Note that for the *general* test are included two beta coefficients and two standard errors.

9.2. Association Method Description

The ***description of the association method*** used in this dissertation begins by contrasting the association between the j^{th} SNP and a disease. For this purpose, we must consider that the alleles have been codified as 0 and 1. Then, we can define the observed genotypes for a set of N

individuals, i.e. N_1 and N_2 for cases and controls, respectively. Next, we can also specify Y_i as the binary variable that will indicate the i^{th} state of a study subject (where $Y_i = 1$ for a case and $Y_i = 0$ for a control). Besides, also considering $G_{ij} \in \{0, 1, 2\}$ as the genotype in the j^{th} SNP of the i^{th} individual (where $i = \{1, 2, \ldots, K\}$), we could summarize all in a table as follows:

G_j	**0**	**1**	**2**
Cases	s_0	s_1	s_2
Controls	r_0	r_1	r_2

Table 9.3: Contrast between the genotype of the j^{th} SNP among all individuals of a case-control study.

If we suppose that the effect of the **1** allele is additive, and defining $p_i = P(Y_i = 1)$ as the association between the j^{th} SNP and a disease, we could model such expression by the following *logistic regression*:

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 G_{ij} \tag{9.1}$$

$$p_i = \frac{e^{\beta_0+\beta_1 G_{ij}}}{1+e^{\beta_0+\beta_1 G_{ij}}} \tag{9.2}$$

In previous equation, β_1 indicates the risk variation for each copy of the **1** allele. Furthermore, for estimating the $\theta = (\beta_0, \beta_1)$ parameters, we can use the MLE method, whose function is the following:

$$L(\theta) = P(Y|G, \theta) = \prod_{i=1}^{N} P_i^{Y_i} (1 - P_i)^{1-Y_i}$$
$$\prod_{i=1}^{N} \left(\frac{e^{\beta_0+\beta_1 G_{ij}}}{1+e^{\beta_0+\beta_1 G_{ij}}}\right)^{Y_i} \left(1 - \frac{e^{\beta_0+\beta_1 G_{ij}}}{1+e^{\beta_0+\beta_1 G_{ij}}}\right)^{1-Y_i} \tag{9.3}$$

Wherein each j^{th} SNP, $Y = (Y_1, \ldots, Y_N)$ and $G = (G_{1j}, \ldots, G_{Nj})$ represent the individual's phenotype and genotype, respectively.

Besides, if we want to get the maximum of the previous function, we should then use iterative optimization techniques such as the *Newton-Raphson* method, which has good convergence properties. In such method, for each iteration, the estimation can be got as follows:

$$\theta^{t+1} = \theta^t - H^{-1}\left(\theta^t\right) U\left(\theta^t\right) \tag{9.4}$$

being:

$$U(\theta) = \frac{d \log L(\theta)}{d\theta}, \qquad H(\theta) = \frac{d^2 \log L(\theta)}{d^2\theta} \tag{9.5}$$

Where $U(\theta)$ and $H(\theta)$ expressions are commonly known as Score and *Hessian*, respectively.

On the other hand, to contrast the association, we should utilize the following hypothesis test:

$$\begin{cases} H_0: & \beta_1 = 0 \\ H_1: & \beta_1 \neq 0 \end{cases} \tag{9.6}$$

Which can be solved by these two ways:

- By the Likelihood-ratio test:

$$\lambda = \frac{max_{H_0} L(\theta)}{max_{H_0 \cup H_1} L(\theta)} \tag{9.7}$$

  Where -2 $\log \lambda \approx \chi_1^2$ when $N_1, N_2 \to$ inf.

- By the **Score test**, which is based on the asymptotic distribution of $U(\theta)$ under H_0 (*null hypothesis*). When H_0 is true the $L(\theta)$ function reaches its maximum close to $\theta_0 = (\beta_0, 0)$, so that $U(\theta_0)$ should be close to zero. Obviously, the θ_0 value will depend on the study samples, but when $N \to$ inf and H_0 are true, the following applies:

$$U(\theta_0) \approx N(0, I(\theta_0)), \qquad \text{being } I(\theta_0) = -H(\theta_0) \tag{9.8}$$

  From here, it results:

$$S = U(\theta_0)^T I^{-1}(\theta_0) U(\theta_0) \approx \chi_d^2, \text{ where } d = \dim(\theta) \tag{9.9}$$

  Note that *Score test* has the advantage that it is not necessary to maximize the likelihood (thereby saving computing time) since its computation only needs to calculate the $U(\theta)$ and $H(\theta)$ values for $\theta = \theta_0 = (\beta_0, 0)$. When H_0 is true, $\beta_1 = 0$ and β_0 simply would be:

$$\beta_0 = \log \frac{p_i}{1-p_i} = \log \frac{N_1/N}{1-N_1/N} = \log \frac{N_1}{N-N_1} = \log \frac{N_1}{N_0} \quad (9.10)$$

A detailed calculation of S for such $_0$ value is further described in [112], resulting the following:

$$S = \frac{N\left(N_2(s_1+2s_2) - N_1(r_1+2r_2)\right)}{N_1 N_2\left(s_1+r_1+4(s_2+r_2) - (s_1+r_1+2(s_2+r_2))^2\right)} \quad (9.11)$$

For *recessive*, *heterozygous* and *general* models we can proceed similarly.

9.3. Association testing of imputed data

The probabilistic nature of imputed SNPs means that testing for association at these SNPs requires some care. Using only imputed genotypes that have a posterior probability above some threshold (or using the best-guess genotype) is a reasonable method of comparing the accuracy across methods, but it is not recommended when carrying out association tests at imputed SNPs. Removing genotypes in this way can lead to both false positives and loss of power [68].

Most programs for performing GWAS association tests assume that all SNPs have been directly identified in the study samples, observed in each case one of these three "possible" genotypes: $G_{i,j} = \{0,1,2\}$. However, when imputed SNPs are included, the genotype uncertainty for each individual has to be modeled according to the probability distribution:

$$\pi_{i,j} = \{\pi_{i,j}(0), \pi_{i,j}(1), \pi_{i,j}(2)\}, \quad \text{where: } \pi_{i,j}(k) = P(G_{i,j} = k) \quad (9.12)$$

SNPTEST2 utilizes a MLE approximation, considering the uncertainty probability distribution in model settings. In this case, three procedures may be used:

I. Given a particular SNP whose association with the disease, verify if $P(G_{SNP} = g|H) > \alpha$ for a pre-specified α threshold uses the imputed value (as if it had been observed), and then proceed with the previously described contrasts. *Therefore, the uncertainty about the true genotype at such SNP is not taken into account.*

II. After the imputation process, calculate the expected values once estimated the $P(G_{SNP} = g|H)$ probabilities and then proceed as in previous contrasts, substituting the S and R values for the expected values. Obviously, this procedure will only work properly if the SNP genotype is known with quite certainly, so the expected values are very close to the real ones. Hence, the lower the certainty of the SNP knowledge, the worse the estimate of the expected values. *Since this process does not take into account the variability in these expectations (due to the genotype uncertainty), it is somewhat a useless procedure in such cases.*

Z	**0**	**1**	**2**
Cases	$E[s_0]$	$E[s_1]$	$E[s_2]$
Controls	$E[r_0]$	$E[r_1]$	$E[r_2]$

Table 9.4: Contrast of a case-control study at the j^{th} SNP among all individuals (after the imputation process).

III. Use statistical theory available to missing values. In this case, if the j^{th} SNP was not observed in the study subjects (nor cases or controls) but it has been imputed from the reference haplotypes, it will not be available G_{ij} values but $P(G_{ij} = k|H), k = 0, 1, 2$ probabilities. Let us remember that, when the G_{ij} values are known, the likelihood to estimate the additive model is:

$$L(\theta) = P(Y|G, \theta) = \prod_{i=1}^{N} P_i^{Y_i} (1 - P_i)^{1-Y_i}$$
$$\prod_{i=1}^{N} \left(\frac{e^{\beta_0+\beta_1 G_{ij}}}{1 + e^{\beta_0+\beta_1 G_{ij}}} \right)^{Y_i} \left(1 - \frac{e^{\beta_0+\beta_1 G_{ij}}}{1 + e^{\beta_0+\beta_1 G_{ij}}} \right)^{1-Y_i} \quad (9.13)$$

When the G value is not known, this likelihood is amended as follows:

$$L(\theta) = P(Y|G, H, \theta) =$$
$$\prod_{i=1}^{N} \sum_{k=0}^{2} \left(\frac{e^{\beta_0+\beta_1 k}}{1 + e^{\beta_0+\beta_1 k}} \right)^{Y_i} \left(1 - \frac{e^{\beta_0+\beta_1 k}}{1 + e^{\beta_0+\beta_1 k}} \right)^{1-Y_i} P(G_{ij} = k|H, G, \theta) \quad (9.14)$$

This expression shows that when there is uncertainty in some SNP, the likelihood function depends on $P(G_{ij} = k|H, G, \theta)$ terms form that correspond to the probability distribution of the unobserved genotypes (conditioned by the reference haplotypes and by the model parameter values). As in the case of known genotypes, there are several ways to test the association between SNP and disease, and in this context we can consider:

a) Using the ***likelihood ratio test***. In this case, it is necessary to find the maximum likelihood estimator of θ for which the *Newton-Raphson* method can be used identically as the above, although now considering the new $U^*(\theta)$ score and $I^*(\theta)$ information functions (either use an EM algorithm). More details are shown in the supplementary information of [85].

b) Using a ***Score test***. As we have seen for the estimate with known genotypes, it is not necessary to estimate β_1, since in this test it is assumed that the null hypothesis ($\beta_1 = 0$) is true. The parameter theta is thus reduced to $(\beta_0, 0)$. As with known genotypes $\beta_0 = \log \frac{p_i}{1-p_i} = \log \frac{N_1/N}{1-N_1/N} = \log \frac{N}{N-N_1} = log \frac{N1}{N_0}$, so β_0 is a fixed value and therefore $P(G_{ij} = k|H, G, \theta) = P(G_{ij} = k|H, G) = p_{ijk}$. The supplementary material of *Marchini and Bryan* [85] the Score test is explicitly described in this case.

c) Using a ***Wald test***. The statistic is $S_{WALD} = \frac{\theta^2}{I^*(\hat{\theta})}$, which is distributed according to χ^2_1 when H_0 is true.

Further explanations of statistical theory for missing data problems can be found in [112] and [85].

9.4. Results using HapMap as Reference Panel

This section will depict the resulting log-p-values of *additive* test using HapMap as the reference panel. We have chosen Manhattan and QQ plots in order to perform the most representative illustrations of data.

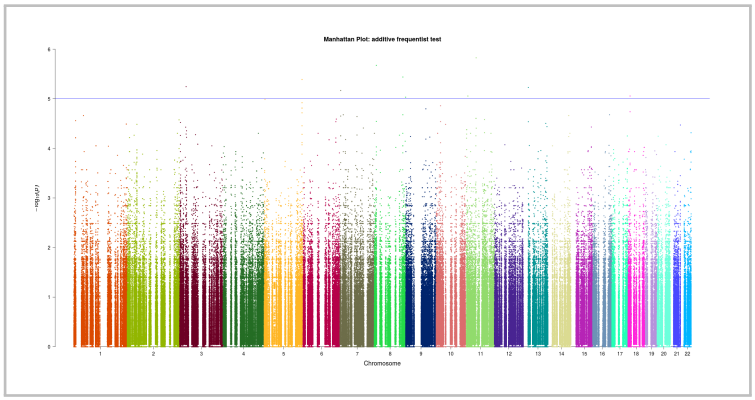


Figure 9.1: Manhattan plot of resulting log-p-values of *additive* test using HapMap as reference panel and IMPUTE2 as imputation software.

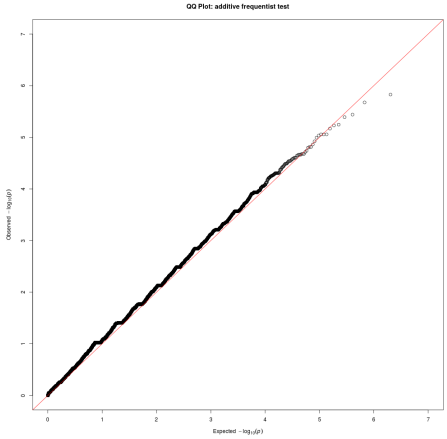


Figure 9.2: QQ plot of the resulting p-values of the *additive* test using HapMap as reference panel and IMPUTE2 as imputation software.

The Manhattan plot of Figure 9.1 represents all log-p-values throughout the Y-axis, being possible a clearly distinction among the corresponding values for each chromosome. It can be noted numerous "gaps" per chromosome, which are due to the low density of HapMap panel.

On the other hand, the QQ plot of Figure 9.2 shows the log-p-values, but differentiating between observed and expected, although with no distinction among chromosomes. It also can be noted that the observed log-p-values are quite similar to the expected ones. Therefore, we can assert that the imputation of study data with the HapMap reference panel **do not lead to significant results**.

9.5. Results using 1000 Genomes as Reference Panel

This section will illustrate the resulting log-p-values of additive tests using 1000 Genomes as the reference panel. We have chosen Manhattan and QQ plots in order to perform the most representative illustrations of data.

We have analysed the imputation results of both MINIMAC3 [124] and IMPUTE2 [121] programs in order to detect if imputation method could have influence the results.

9.5.1. Manhattan and QQ plots from MINIMAC3 imputation

In this subsection, once performed the imputation with MINIMAC3, the resulting log-p-values of *additive* test (using 1000 Genomes as reference panel) are depicted in Figures 9.3.

Such figure shows more density than results from HapMap, leading to an improvement of the significant log-p-values in all tests contrasted. It can be noted that in certain genome regions these values surpass a threshold of 10^{-5}.

As well, it has been illustrated the observed and expected log-p-values of the *additive* test (using 1000 Genomes as reference panel) in Figure 9.4.

Such QQ plot illustrates that a greater quantity of expected log-p-values differ from the observed ones (in comparison with depicted values in QQ plot of Section 9.4). From a threshold of 10^{-5}, it can be noted how numerous observed log-p-values walk away from the expected ones.

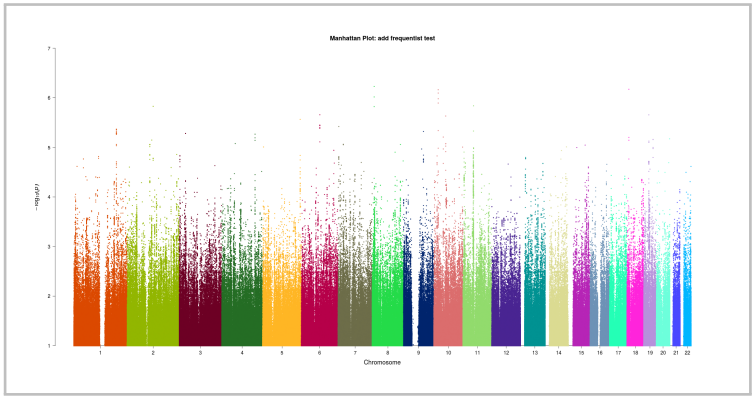


Figure 9.3: Manhattan plot of resulting log-p-values of the *additive* test using 1000 Genomes as reference panel and MINIMAC3 as imputation software.

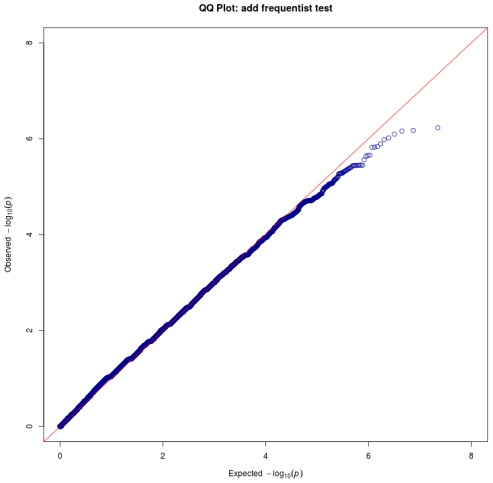


Figure 9.4: QQ plot of the resulting p-values of the *additive* test using 1000G as reference panel and MINIMAC3 as imputation software.

9.5.2. Manhattan and QQ plots from IMPUTE2 imputation

In this subsection, once performed the imputation with IMPUTE2, the resulting log-p-values of the *additive* test (using 1000 Genomes as reference panel) is depicted in Figure 9.5.

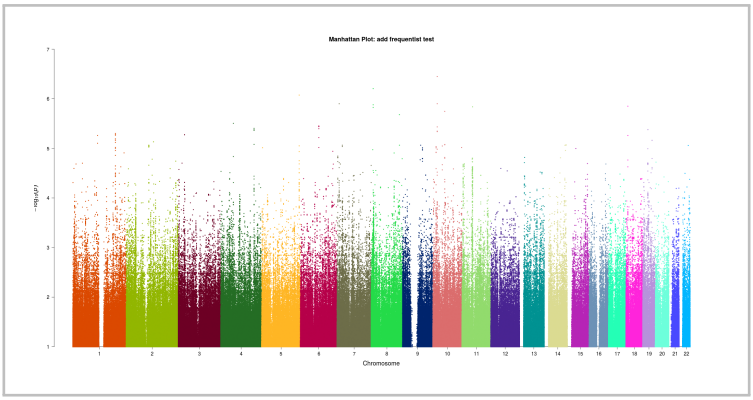


Figure 9.5: Manhattan plot of resulting log-p-values of the *additive* test using 1000G as reference panel and IMPUTE2 as imputation software.

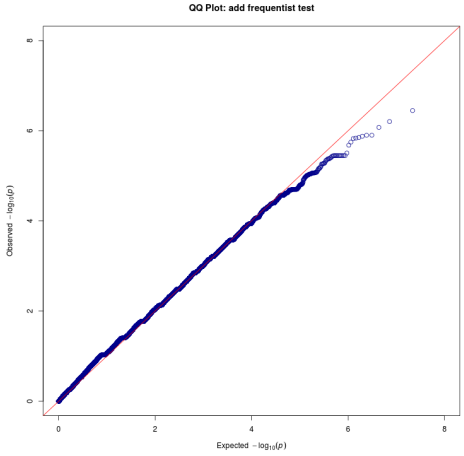


Figure 9.6: QQ plot of the resulting p-values of the *additive* test using 1000G as reference panel and IMPUTE2 as imputation software.

Figure 9.5 shows more density of values than results using HapMap as reference panel (see Section 9.4), leading to even more significant log-p-values in comparison with MINIMAC3 results.

The observed and expected log-p-values of the *additive* test (using 1000 Genomes as reference panel) has also been represented in Figure 9.6. In such QQ plot depicts how observed log-p-values move away of expected values from lower threshold than in Section 9.4. So, the total amount of significant values (comparing them from a given threshold) is greater using IMPUTE2 as imputation software than with MINIMAC3 program.

9.6. Selection of Significant Results

The corresponding association analysis was performed using all frequentist tests (i.e. additive, dominant, general, recessive and heterozygous). We ultimately chose the *additive* results with and without adjusting by the "retinopathy" co-variable (i.e., if subjects were affected by the retinopathy disease or not). To filter and select the most significant results, we have decided to group them in regions containing, at least, three significant SNPs (with p-values smaller than 10^{-5}). Furthermore, we have selected those INFO values greater than *0.99*. In Tables 9.5 and 9.6 above results' selection are shown.

In both tables, each line stores the information of a genomic region where CHR column is the chromosome, REG is the selected region, RSID is the original SNP name, RSIDNEW is the updated SNP name (by the Build 142 of the SNP locations for Homo sapiens), POSITION is the genomic position where each SNP is located (in base-pairs), ALA is the main allele, ALB is the alternate allele, PVALUE is the resulting p-value, TESTINFO is the imputation quality, BETA is the coefficient β, SE is the standard error, GENESYMBOL is the symbol of the gene to which the SNPs belong.

From Table 9.5 it can be seen that the three lowest p-values are in regions 8.1, 10.2 and 11.1, as well in Table 9.6 but with the difference SNP that, in such table the p-values adjusted by the *retinopathy* co-variable are better. From each region we selected one SNP, being the chosen: rs4841106, rs2358658, and rs35649357.

Figures 9.7, 9.9 and 9.11 show the typed SNPs. Representations of the same SNPs but with imputed information are arranged in Figures 9.8, 9.10 and 9.12.

Chr	Reg	Typed	rsID	rsIDnew	Position	alA	alB	Pvalue	testINFO	Beta	SE	GENEsymbol
1	1.2	No	rs3850619	rs3850619	199109114	G	C	5.4161×10^{-6}	0.994034	-1.37199	0.336038	
1	1.2	Yes	rs10800611	rs10800611	199115592	T	C	5.2404×10^{-6}	1	-1.35607	0.331267	
3	3.1	No	rs7306069	rs7306069	28948053	G	A	5.2923×10^{-6}	0.998066	-2.05171	0.53348	
3	3.1	Yes	rs11710772	rs11710772	28960287	C	T	5.2970×10^{-6}	1	-2.0502	0.53266	
4	4.2	No	rs1490674	rs1490674	155330282	T	C	4.3292×10^{-6}	0.99295	1.48872	0.361206	DCHS2
4	4.2	No	rs990185	rs990185	155330882	G	C	4.1907×10^{-6}	0.994467	1.4906	0.361129	DCHS2
4	4.2	No	rs6818438	rs6818438	155332511	G	A	3.9360×10^{-6}	0.997257	1.49442	0.361039	DCHS2
6	6.1	No	rs78261651	rs78261651	85799282	A	C	3.5482×10^{-6}	0.999626	-8.4999	16.1244	
6	6.1	Yes	kgp4641172	rs78335147	85802481	C	A	3.5506×10^{-6}	1	-8.50257	16.1523	
6	6.1	Yes	kgp4434658	rs55897238	85804399	C	T	3.5506×10^{-6}	1	-8.50257	16.1523	
8	8.1	Yes	rs4841106	rs4841106	9044856	A	G	1.4934×10^{-6}	1	-1.53146	0.360835	LOC101929128
8	8.1	No	rs4841107	rs4841107	9044897	C	G	1.3173×10^{-6}	0.997214	-1.54226	0.361905	LOC101929128
10	10.2	No	rs2884507	rs2884507	20305489	A	T	1.2544×10^{-6}	0.997256	1.54494	0.361962	PLXDC2
10	10.2	Yes	kgp517295	rs2358658	20307083	G	A	4.5752×10^{-6}	1	1.46649	0.358001	PLXDC2
11	11.1	Yes	kgp8554447	rs35649357	49552399	T	G	1.4489×10^{-6}	1	-8.28856	14.3315	
18	18.1	No	rs1784763	rs1784763	9067924	G	A	5.4694×10^{-6}	0.999983	8.33617	15.4622	
19	19.1	No	rs4254447	rs4254447	24477702	T	C	4.1705×10^{-6}	1	1.89291	0.473502	
19	19.1	No	rs10414165	rs10414165	24478348	T	C	4.1705×10^{-6}	1	1.89291	0.473502	

Table 9.5: Most significant SNPs from *additive* test (without adjusting by any co-variable).

Chr	Reg	Typed	rsID	rsIDnew	Position	alA	alB	Pvalue	testINFO	Beta	SE	GENEsymbol
3	3.1	No	rs73060069	rs73060069	28948053	G	A	3.67826×10^{-6}	0.997356	-2.12155	0.545348	
3	3.1	Yes	rs11710772	rs11710772	28960287	C	T	3.68872×10^{-6}	1	-2.11937	0.544212	
8	8.1	Yes	rs4841106	rs4841106	9044856	A	G	1.36741×10^{-6}	1	-1.54889	0.364094	LOC101929128
8	8.1	No	rs4841107	rs4841107	9044897	C	G	1.42376×10^{-6}	0.996237	-1.54557	0.363528	LOC101929128
8	8.3	Yes	rs17289507	rs17289507	131493326	C	T	3.93318×10^{-6}	1	-8.65919	16.7397	
9	9.1	No	rs9696078	rs9696078	92503499	G	A	5.99158×10^{-6}	0.99762	1.3629	0.335209	
10	10.1	No	rs2884507	rs2884507	20305489	A	T	1.00068×10^{-6}	0.998724	1.56999	0.36486	PLXDC2
10	10.1	Yes	kgp517295	rs2358658	20307083	G	A	3.2376×10^{-6}	1	1.50004	0.360917	PLXDC2
11	11.1	Yes	kgp8554447	rs35649357	49552399	T	G	1.30259×10^{-6}	1	-8.33769	14.2969	
18	18.1	No	rs1784763	rs1784763	9067924	G	A	2.74509×10^{-6}	0.999994	8.49673	15.3799	
19	19.1	No	rs62116441	rs62116441	24189406	C	T	4.1583×10^{-6}	0.993668	1.92528	0.470914	
19	19.2	No	rs4254447	rs4254447	24477702	T	C	4.96705×10^{-6}	1	1.87717	0.473332	
19	19.2	No	rs10414165	rs10414165	24478348	T	C	4.96705×10^{-6}	1	1.87717	0.473332	
19	19.3	Yes	rs7259375	rs7259375	43885085	T	C	4.78323×10^{-6}	1	1.61444	0.392688	
19	19.3	No	rs8109116	rs8109116	43901799	G	C	2.18859×10^{-6}	1	-1.68328	0.397624	TEX101
19	19.3	Yes	rs8109124	rs8109124	43901831	G	A	2.18859×10^{-6}	1	-1.68328	0.397624	TEX101

Table 9.6: Most significant SNPs from *additive* test (adjusted by the *retinopathy* co-variable).

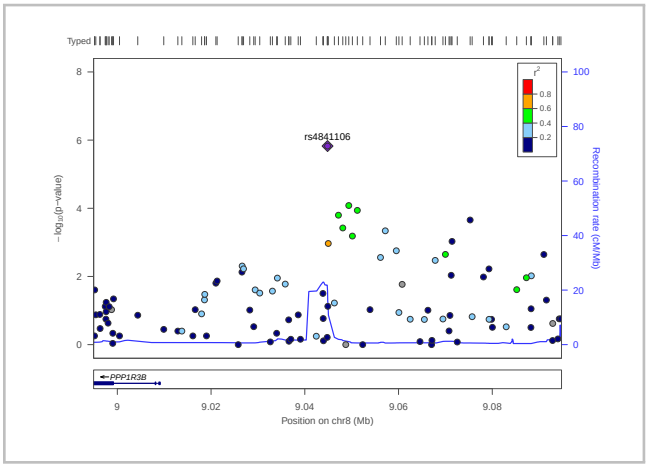


Figure 9.7: First region of Typed SNPs, where the rs4841106 SNP is located at the center of the image.

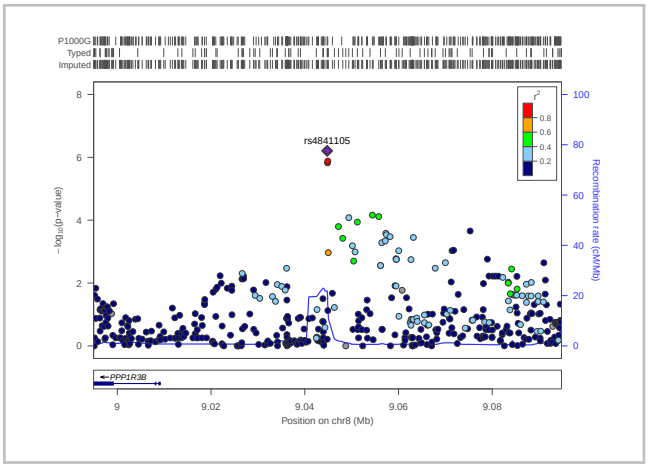


Figure 9.8: First region of typed and imputed SNPs, where the rs4841106 SNP is located at the center of the image.

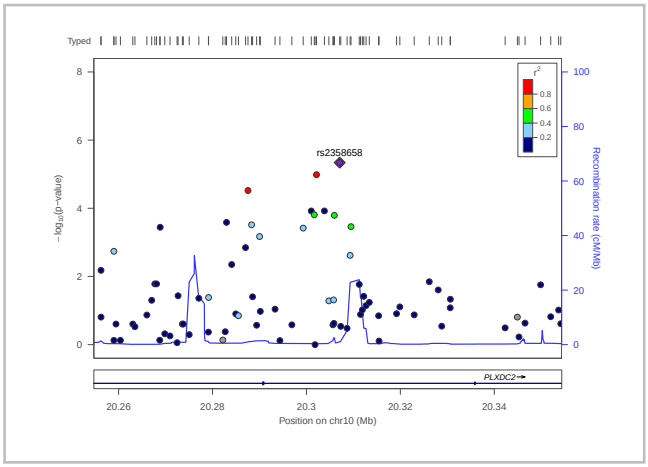


Figure 9.9: Second region of Typed SNPs, where the rs2358658 SNP is located at the center of the image.

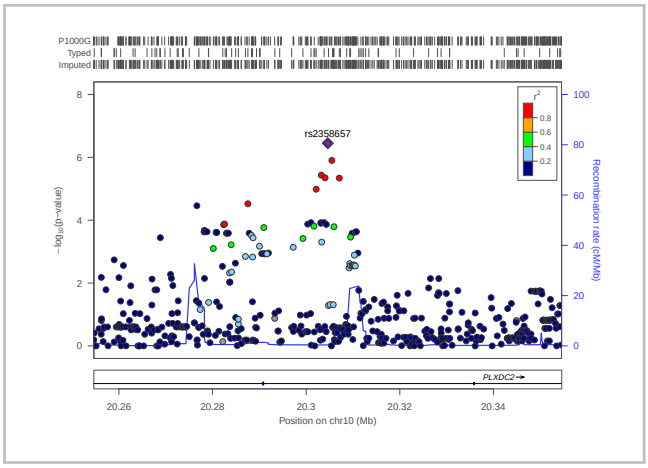


Figure 9.10: Second region of typed and imputed SNPs, where the rs2358658 SNP is located at the center of the image.

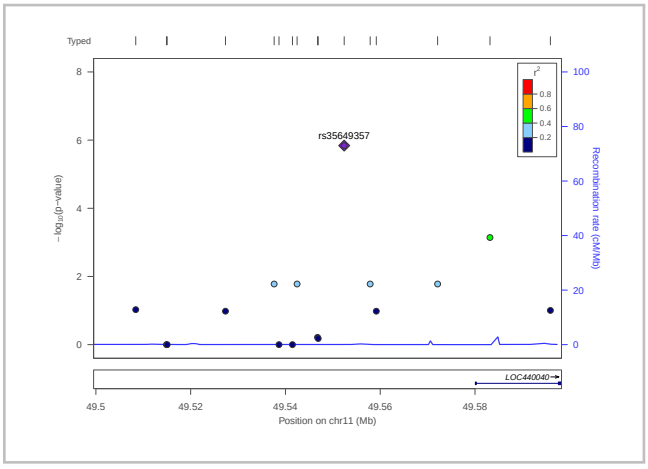


Figure 9.11: Third region of Typed SNPs, where the rs35649357 SNP is located at the center of the image.

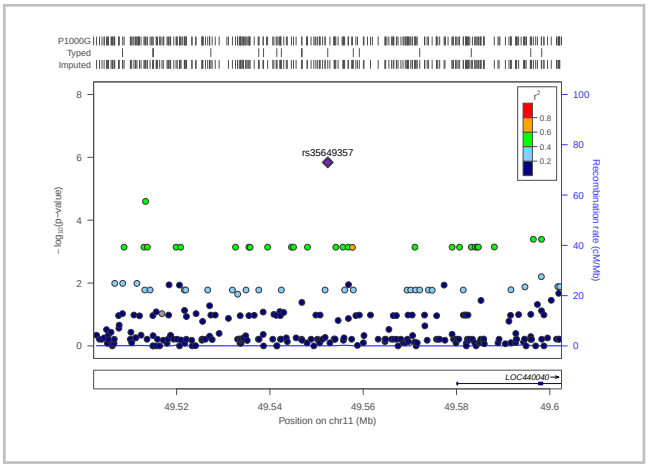


Figure 9.12: Third region of typed and imputed SNPs, where the rs35649357 SNP is located at the center of the image.

It can be clearly seen that p-values how the imputation results are better than the corresponding only to the typed SNPs.

PART

IV

Approaches for Family-based Data

This part of the thesis is motivated originally by two different problems. The first one was to identify possible maternal effects on the age of onset of T1D. The second was to detect (if possible) differences between the genetic structure in HLA haplotypes that confer a high risk of developing T1D in a population from the Canary Islands compared to the rest of Spain, and from Spain to the rest of Europe. For this purpose, we used data from the T1DGC. These data contain the genetic information of Class I and II markers of the HLA region for several thousand families all around the world with at least two affected children per family. As part of the analysis, it was necessary to identify the haplotypes carried by individuals on these markers. To this aim, we have developed a new R package capable of imputing missing alleles and identifying haplotypes from non-recombinant regions considering the genetic information present in parents and offspring and the mechanism of heredity. The algorithm is deterministic in the sense that haplotypes are identified from the existing genotypes guaranteeing compatibility between parents and children. When a haplotype cannot be identified (due to genotyping errors, or recombination events in the genetic region), the procedure does not infer more haplotypes in the corresponding family members. The following chapters will describe the implementation of this package as well as its application to the original problem.

Chapter 10

alleHap Package: Description

10.1. Introduction

We have developed an R package called alleHap, which allows the imputation of alphanumeric alleles from parent-offspring pedigrees, even containing large amounts of missing data. It also reconstructs their corresponding unambiguous haplotypes, when possible. alleHap is a **deterministic** proposal that is very robust against inconsistencies within the genotypic data and consumes very little time, even when handling large datasets. It has been tested by simulations and also with real data. The program is implemented in R language and available for Linux, Mac and Windows platforms as an R package from CRAN website.

10.2. Theoretical Description

The theoretical description of the alleHap package is based on a preliminary analysis of all possible combinations that may exist in the genotype of a family, considering that each member, due to meiosis, should unequivocally have inherited two alleles, one from each parent (see Figure 10.1).

Genetic information may be arranged so that an allele may correspond to a single nucleotide (A, C, G, T) or a DNA nucleotide sequence, taking into account that the allele nomenclature in both cases could comprise alphanumeric values chosen for a given purpose, such as described in section 2.3.3.1.

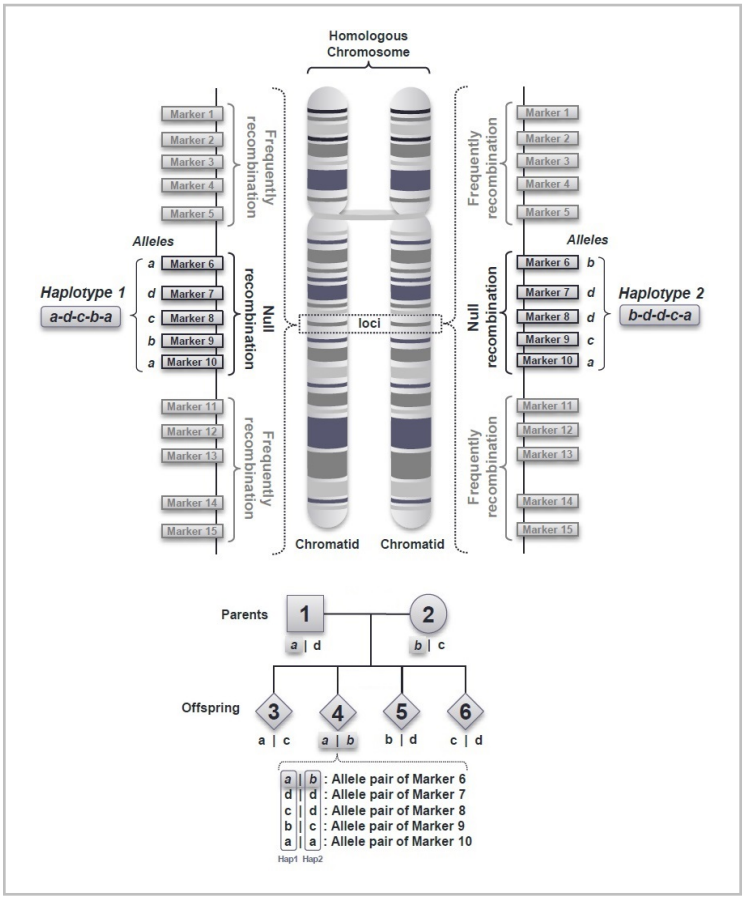


Figure 10.1: Above: *Description of null and frequently DNA recombination regions and their corresponding alleles/haplotypes located in a homologous chromosome.* Below: *Illustration of a parent-offspring pedigree containing 6 members: 2 parents and 4 children.*

The analysis is based on the differentiation of seven configurations, as was described in [169]. Each case has been grouped considering the number of unique (or different) alleles per family. So, using the notation N_{par}: *Number of unique alleles in parents* and N_p: *Number of unique alleles in parent p*, the expression: (N_{par}, N_1, N_2) allows to identify all the non-recombinant configurations in families with one line of descent

(i.e. parent-offspring pedigrees). The table 10.1 shows the different configurations in biallelic mode:

Configuration	**1**	**2**	**3**	**4**	**5**	**6**	**7**
N_{par}	1	2	2	3	2	3	4
(N_1, N_2)	(1,1)	(1,1)	(1,2)	(1,2)	(2,2)	(2,2)	(2,2)
Parental Genotypes	a/a a/a	a/a b/b	a/a a/b	a/a b/c	a/b a/b	a/b a/c	a/b c/d
Possible Offspring Genotypes	a/a	a/b	a/a a/b	a/b a/c	a/a b/b a/b	a/a a/b a/c b/c	a/c a/d b/c b/d

Table 10.1: Biallelic configurations in a parent-offspring pedigree

In configurations 1,2,5,6 and 7, this designation may be used without the need to specify who are the father and the mother. For configurations 3 and 4, it will be necessary to add an index (1 or 2) to indicate whether is homozygous the father (1) or the mother (2).

The **configuration number 5** (or ***Case 5***), deserves a special mention since the heterozygous genotypes of both parents share the same alleles. This fact complicates a deterministic identification of the corresponding alleles to in those children with heterozygous genotypes. Therefore, it is not possible to determine with a 100% of certainty, which allele has been inherited from the father and which one from the mother, i.e. it is impossible to phase ***deterministically*** such children genotypes.

10.2.1. Basic Notation

To identify (in each child) which of the two alleles has been inherited from the father and which one from the mother, the following notation is used:

- $\boldsymbol{F = \{F_1, F_2\}}$: paternal alleles, *sorted lexicographically*;
- $\boldsymbol{\bar{F} = \{\bar{F}_1, \bar{F}_2\}}$: paternal alleles, *sorted by the offspring alleles*;
- $\boldsymbol{M = \{M_1, M_2\}}$: maternal alleles, *sorted lexicographically*;
- $\boldsymbol{\bar{M} = \{\bar{M}_1, \bar{M}_2\}}$: maternal alleles, *sorted by the offspring alleles*;
- $\boldsymbol{C_h = \{C_{h1}, C_{h2}\}}$: child h alleles, *sorted lexicographically*;
- $\boldsymbol{\bar{C}_h = \{\bar{C}_{h1}, \bar{C}_{h2}\}}$: child h alleles, *sorted in such way that* $\bar{C}_{h1} \in F$ *and* $\bar{C}_{h2} \in M$, where the first allele comes from the father and the second from the mother, and $h \in \{1, ..., H\}$, being H the number of children per family.

10.2.2. Advanced Notation

When K markers are observed, in order to clarify the genotype identification and to simplify posterior computations, the following notation will be used for describing the alleles in the i^{th} subject of a family, where $i = 1$ is the father, $i = 2$ the mother, $i > 2$ are the offspring:

$$\boldsymbol{A_i} = \begin{bmatrix} A_{11i} & A_{12i} & \dots & A_{1Ki} \\ A_{21i} & A_{22i} & \dots & A_{2Ki} \end{bmatrix} \tag{10.1}$$

Each column k of this matrix represents a marker, being (A_{1ki}, A_{2ki}) the pair of alleles identified in that marker. Either allele (or both) may be missing, and would then be denoted as *Not Available* (NA).

10.2.2.1. Inheritance identifiers

Associated with previous matrix, ***inheritance identifiers*** for all subjects can be defined, being IDS_{hki} the ***inheritance identifier*** of the marker k in the individual i. Then, the corresponding matrix of the IDS_{hki} values can be defined as:

$$\boldsymbol{IDS_i} = \begin{bmatrix} IDS_{11i} & IDS_{12i} & \dots & IDS_{1Ki} \\ IDS_{21i} & IDS_{22i} & \dots & IDS_{2Ki} \end{bmatrix} \tag{10.2}$$

Where:

a) ***For parents ($\boldsymbol{i = 1, 2}$)***, $IDS_{hki} = 1$ if the allele transmitted to the reference child can be determined, and $IDS_{hki} = 0$ in other case.

b) ***For offspring ($\boldsymbol{i \geq 3}$)***, $IDS_{hki} = 1$ if the pair of alleles is completely identified (i.e. it is possible to determine which allele is inherited from the father and which one from the mother), and $IDS_{hki} = 0$ in other case.

Namely,

$$\boldsymbol{IDS_{hki}} = \begin{cases} 0 & \text{if allele } A_{hki} \text{ does not belong to} \\ & \text{haplotype } h, \text{ or is missing} \\ 1 & \text{if allele } A_{hki} \text{ belongs to haplotype } h \end{cases}, \quad h = 1, 2$$

In this way, if all terms in the matrix IDS_i are 0, the phase of each allele is unknown. In turn, when all terms are equal to 1, the alleles are phased, and the rows of the matrix A_i can be read as the haplotypes of the i^{th} subject of the family.

When familial genotypes are read, the matrices IDS_i are initially equal to 0 for all members, as the genotype phase is unknown. At the *Data Haplotyping* stage (see Section 10.3.4), the main objective is to order the A_{hki} alleles in each marker of every subject in such a way that the IDS_i matrices may contain as values equal to 1 as possible. When the h row in IDS_i is completely (or partially) filled with ones, the corresponding h^{th} row in the A_i matrix can be deterministically phased.

10.2.2.2. Homozygosity identifiers

In the same way, as described in the previous paragraph, a vector of ***homozygosity identifiers (HMZ)*** per individual can be defined. Therefore, $\mathbf{HMZ_{ik} = 1}$ if individual i is **homozygous** in marker k (i.e. identical alleles), and $\mathbf{HMZ_{ik} = 0}$ if the subject is **heterozygous**. Consequently, considering a given individual of family, the HMZ vector is defined as:

$$\mathbf{HMZ_i} = \left[\begin{array}{cccc} HMZ_{i1} & HMZ_{i2} & \ldots & HMZ_{ik} \end{array} \right] \tag{10.3}$$

10.3. Practical Description

This section describes the practical operation of alleHap. Regarding the workflow of the package, alleHap is comprised of the following stages: ***Data Loading***, ***Data Imputation*** and ***Data Haplotyping***, such as depicted in Figure 10.2.

Thus, in this section, the alleHap procedures, as well as their implementation by R functions will be described. The functions corresponding to the three central stages are:

- alleLoader, which reads genotypic data from an external PED file or an R data frame.
- alleImputer, which imputes missing alleles (marker by marker) from the available genotypes.
- alleHaplotyper, which re-imputes missing alleles (using information from adjacent markers), to later reconstruct the corresponding haplotypes.

In order to test the accuracy and performance of the package, it was necessary the development of a data simulator that is included in a *"pre-stage"* called ***Data Simulation***, which is also explained in Subsection 10.3.1.

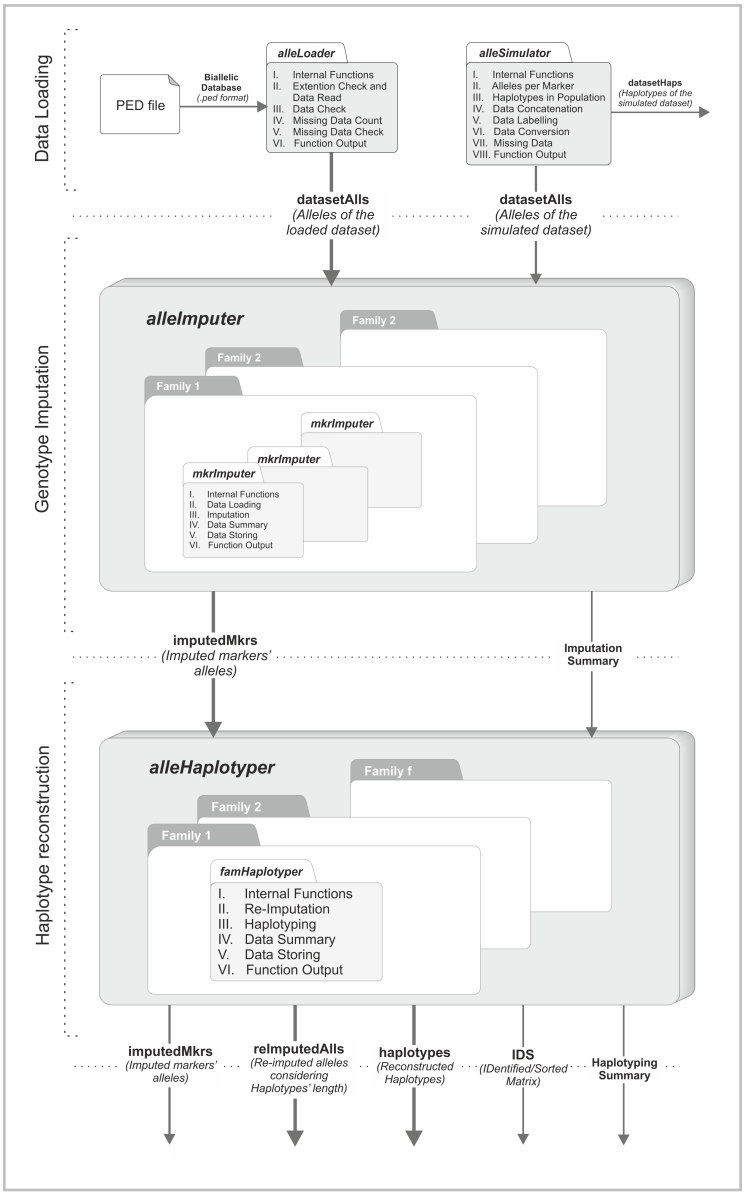


Figure 10.2: Diagram describing the three main stages of the package.

10.3.1. Data Simulation Pre-stage

Data simulation was implemented by an R function called alleSimulator, which simulates genotypic data for parent-offspring pedigrees.

This function creates biallelic pedigree databases, which can be generated taking into account many different factors such as: number of families to generate, number of markers (allele pairs), maximum number of different alleles per marker in the population, type of alleles (numeric or character), number of different haplotypes in the population, probability of parent/offspring missing genotypes, proportion of missing genotypes per individual, probability of being affected by disease, and recombination rate.

10.3.1.1. alleSimulator Function

alleSimulator function generates the clinical and genetic information of a group of families according to the previously defined parameters. In order to simulate the data, this function has been developed in several steps:

I. **Internal Functions**: In this step are loaded all the necessary functions to simulate the data. These functions are ***labelMrk*** (which creates the 'A','C','G','T' character labels), ***simHapSelection*** (which selects n different haplotypes between the total number of possible haplotypes), ***simOffspring*** (which generates n offspring by selecting randomly one haplotype from each parent), ***simOneFamily*** (which simulates one family from a population containing the haplotypes *'popHaplos'*), and ***simRecombHap*** (which simulates the recombination of haplotypes).

II. **Alleles per Marker**: The second step is the simulation of a number of alleles per marker for the entire population (if they are not supplied by the user). It is assigned an allele range per marker whether alleles are not character type and, if alleles are character type, they are repeated two times.

III. **Haplotypes in population**: Once the number of alleles per marker and the haplotypes size of the population (n) are specified, the population haplotypes are generated. In this process, n different haplotypes were selected among the total number of possible haplotypes.

IV. **Data Concatenation**: In this step, clinical and previously simulated data of all families are concatenated.

V. **Data Labelling**: The fifth step is the labelling of the previous concatenated data (*"famID"*, *"indID"*, *"patID"*, *"matID"*, **"sex"**, *"phen"*, *"markers"*, *"recombNr"*, *"ParentalHap"*, *"MaternalHap"*).

VI. **Data Conversion**: In this step, the conversion of the previous generated data into the most suitable type (integer and/or character) is performed.

VII. **Missing Data Generation**: The seventh step is the insertion of missing values in the previously generated dataset (only when users require it). The missing values may be generated taking into account four different factors: *missParProb* (probability of parents' missing genotype), *missOffProb* (probability of offspring' missing genotype), *ungenotPars* (proportion of ungenotyped parents) and *ungenotOffs* (proportion of ungenotyped offspring).

VIII. **Function Output**: The last step is the creation of a list containing two different data.frames, for genotype and haplotypes respectively. This may be useful in order to compare simulated haplotypes with later reconstructed haplotypes.

Full examples of alleSimulator function are provided by Section B.2.1 of Appendix B.

10.3.2. Data Loading Stage

The package can be used with either simulated or real data, and can handle or not genetic missing information. As it has mentioned at Section B.1, .ped files are the default input format of alleHap, and although its loading process is quite simple, it is important to note that a file containing **a large number of markers could slow down the process**. Furthermore, to avoid the preceding, it is highly recommended that users *split the data into non-recombinant chunks*, where each chunk should be later loaded by the alleLoader function.

Previously to the loading process, users should check what kind of missing values has the intended PED file. If those values are different than "-9" or "-99", the parameter *"missingValues"* of alleLoader has to be updated with the corresponding value. Per example, if the PED file

has been codified with zeros as missing values, it has to be specified *'missingValues = 0'*.

alleHap has been tested with the T1DGC database [101]. This database consisted of over 3000 families and 20 markers (16 numeric and 4 character type: "A", "C", "G" or "T"), as described in Section 4.5.3 of Chapter 4.

10.3.2.1. alleLoader Function

The alleLoader function tries to read family data from an R data frame or an external file, to later pass it into the alleImputer or alleHaplotyper functions. For this purpose, this function has been developed in five steps:

I. **Internal Function**: In this step is loaded the auxiliary function *recodeNA*, which recodes pre-specified missing data as NA values.

II. **Extension check and data read**: In this step the extension file is checked and if it has a .ped type the dataset is loaded into R as a data.frame. Should this not occur, the message *"The file must have a .ped extension"* is returned, and the data will not be loaded. Then, if the file extension is appropriate, data is loaded, and missing values (by default *-9* or *-99*) are recoded as NAs (although any other value may be supplied by the user).

III. **Data check**: The third step counts the number of families, individuals, parents, children, males, females and markers of the dataset, as well as, it checks the ranges of *Paternal IDs*, *Maternal IDs*, *genotypes* and *phenotype* values.

IV. **Missing data count**: This step counts the missing/unknown data that may exist in genetic data or subjects' identifiers.

V. **Function output**: In the final step, the dataset is returned as an R data.frame, with the same structure as a PED file, with the variables renamed, and the missing values correctly identified and coded. If *'dataSummary = TRUE'* a summary of previous data counting, ranges, and missing values is printed into the screen.

The intended datasets must conform to the specifications of a PED file: in each row the first six variables correspond to *family ID*, *subject ID*, *paternal ID*, *maternal ID*, *sex*, and affection status (*phenotype*). The

rest of the variables are the observed genotypes in each marker, being each one composed by other two fields (main and alternate allele).

Full examples of alleLoader function are provided at Section B.3.1.1 of the Appendix B.

10.3.3. Data Imputation Stage

This package stage tries to impute previously simulated/loaded datasets by analyzing all possible combinations of a parent-offspring pedigree in which parental and/or offspring genotypes may be missing; as long as one child was genotyped, in certain cases it is possible an unequivocal imputation of missing genotypes both in parents and children.

At this stage, firstly is conducted a simple quality control of data, and secondly, it imputes alleles marker by marker in those cases where possible. The way these tasks are carried out is described below:

10.3.3.1. Imputation Quality Control

This subsection will describe the all possible conditions (cases) where the genotype imputation of missing data and haplotype reconstruction are possible. Such conditions are verified **for every marker**, taking into account the following features:

1. There cannot be more than two different homozygous children in a family.
2. If there are two different homozygous children in the family, there cannot be a different allele in any other family member.
3. Considering all the individuals of a family, there can be at most four different alleles in a marker.
4. If a family has four different alleles in a marker, no child can be homozygous.
5. If there are three (or more) unique heterozygous children, they cannot share a common allele.
6. There cannot be more than four genotypically different children per marker and family.
7. If a child has the same alleles as one of his/her parent, there can only be at most three different alleles in the family.
8. When parental alleles are not missing:

a) Each child must have at least a common allele with each parent.
b) A child cannot have an allele not present in any of the parents.

In previous cases, when referring to families comprising by more than one child, it has been considered that children have different genotypes each other. This is because of two or more children share the same genotype in all the markers. For the purposes of genotype identification, he/she will count as an only child (since will not provide different or new information).

10.3.3.2. Imputation Procedure

Throughout this procedure, a "simple" allele imputation (marker by marker) is performed as follows:

1. **Imputation in children**. It is identified which allele has been inherited from the father and which from the mother. If a parent has a homozygous genotype, the corresponding allele is imputed to all children with missing alleles (which do not already have this allele). Moreover, if both parents are homozygous, all children with missing alleles are readily imputed.
2. **Imputation in parents**. Given a reference child, it is determined which allele has been transmitted to such child:
   a) If a child has homozygous genotype, the allele is imputed to that parent that do not already have this allele.
   b) If a parent has missing alleles and the other not, and there are heterozygous children, the alleles present in those children (which are not located in the non-missing parent) are imputed to the parent with missing alleles.

10.3.3.3. alleImputer Function

The alleImputer function sorts the alleles for each family marker (when possible) and then imputes the missing values. For this purpose, the operation of this function can be reduced to the following steps:

I. **Internal Functions**: In this step all the necessary functions to impute the data are loaded. The most important ones are:
   - mkrImputer, which performs the imputation of a marker. This function firstly receives as input data the alleles of one marker in one family, then applies the quality control, and imputation processes described above and finally returns the (possible) imputed markers' alleles.

- famImputer, which sequentially applies mkrImputer to impute all markers in one family.
- famsImputer, which applies famImputer to all families of the given data frame, returning a dataset with the same format and dimensions as the input data (with imputed values in those alleles where imputation has been possible).

II. **Data Loading**: The second step tries to read genotypic data and the families information into a fully compatible format by means of the alleLoader function. If this process is successful, data are stored in an R data.frame with the same structure as a PED file.

III. **Imputation**: This is the most important step of the alleImputer function. Firstly, marker by marker and then family by family, the imputation of the corresponding missing alleles is performed by the mkrImputer function in two stages: children imputation and parent imputation. Given a marker with missing values, these can be imputed only either the genotypes of a parent and/or a child are homozygous. If in a marker, one parent has missing alleles and the other not, and the heterozygous alleles of children are not present in the complete parent, those alleles are imputed to the other parent.

IV. **Data Summary**: Once the imputation is done, a summary of the imputed data are collected. This summary gathers information about the imputation process, i.e. number of imputed alleles, identified incidences (number of canceled markers due to problems detected in the quality control process), imputation rate (quotient of the imputed alleles to the number of initially missing alleles) and time consumed in the process.

V. **Data Storing**: In this step, the imputed data are stored in the same path where the PED file was located, as long as, data have been read from an external file. The generated new file will have the same name and extension as the original, ending as *'imputed.ped'*.

VI. **Function Output**: In this final step, if *'dataSummary = TRUE'* the imputation summary may be printed out. Imputed data can be directly returned as an R data.frame (with the same structure and dimensions as the input dataset), whether *'invisibleOutput'* is deactivated. Incidence messages can be shown, if they are detected at the *quality control*, described in Subsection 10.3.3.1.

Full examples of alleImputer function are provided at Section B.3.2.1 of Appendix B.

10.3.4. Data Haplotyping Stage

At this stage, the corresponding haplotypes of the biallelic pedigree databases are generated. To accomplish this, based on the user's knowledge of the intended genomic region to analyse, it is necessary to **slice the data into non-recombinant chunks** in order to perform the haplotype reconstruction to each of them.

Users may choose among several icons in order to specify the non-identified and missing values in the haplotypes, as well as, it is also possible to define the character that will be used as a separator between alleles when generating the corresponding haplotypes.

This stage intends to reconstruct haplotypes from a dataset containing genotypic familial data. Genotypic markers are supposed to be part of the same haplotype, i.e. there is no recombination between markers. For this purpose, we have considered that when missing data occurs in a subject's marker, missingness affects to both alleles (i.e. each marker is fully missing for the given subject); but if the subject is a child and a parent is homozygous at the same marker (say *G/G*), one and only allele will be imputed to such child by alleImputer function. Thus, the child's genotype would be *G/NA* (where NA stands for missing value). The same occurs if a fully missing marker is located in a parent and there is a homozygous child in that marker.

10.3.4.1. Haplotyping Procedure

The ***haplotyping procedure*** begins by considering only the offspring, trying to identify/sort the alleles in each marker in such a way that the allele in the first row of the matrix A_i is the one inherited from the father (see Subsection 10.2.2), and the allele in the second row be the inherited from the mother. So, if all the markers are sorted this way, the first row of the matrix A_i would inherit the first haplotype from the father and the second one from the mother. Once these haplotypes have been found in children, they can be readily identified in parents. What complicates this idea and makes difficult its direct application is the fact that, in some cases, both parents and a child can share the same genotype (say *G/T* for the three subjects), and therefore it is not possible to know which allele has been inherited from which parent.

Also, there may be missing alleles in parents or children that prevent to specify the origin of some alleles in some markers. In particular, if both parents have all the alleles missing in a marker, it is impossible to determine the provenance of the alleles of that marker in the children, at least if there are less than three children in the family. As we will see in Subsection 10.3.4.2 when the family has three or more children, if there are no missing alleles in at least three children, deterministic phasing can be carried out even when parents are completely missing. Also, in Subsection 10.3.4.2 we show that in the particular case of having only two children, if parental alleles are available in some markers and the other markers are entirely missing, it is possible (under certain conditions) to determine the alleles' phase in those missing markers.

10.3.4.2. Haplotyping Scenarios

We have considered four scenarios for the haplotyping procedure. In the first one, there are no complete missing markers in parents, and children may be completely without missing alleles or may have full or partially missing data. In the second one, we have taken into account that all of the parental markers are entirely missing, and there are at least three children in the family without missing alleles. The third scenario is a mixture of the previous two: some markers have parents with completely missing alleles and, at least, three complete children (without missing alleles). Some markers have non-missing alleles in parents, with some missing values in children, and some markers may have no missing values in parents nor children. Finally, in the fourth scenario, we show the conditions in which alleles can be deterministically phased with only two children when parents have completely missing markers.

Scenario 1: There are no completely missing markers in parents

Regarding the genotypic data of **one** family, the haplotyping process to deterministically phase the corresponding alleles is the following:

I. Set $i = 3$, $k = 1$ (recall that family members are indexed so that $i = 1$ is the father, $i = 2$ the mother and $i = 3, 4, \ldots$ are the children).

II. Given the marker k in the i^{th} member of the family ($i \geq 3$, so only children are considered), check if it is possible to determine unequivocally (for each marker k) which allele has been inherited

from the father and which one from the mother. This can be readily done under the following circumstances:

a) If a child is homozygous in that marker k, the two alleles are identical and so it is trivial to assign one copy to each parent.

b) If a child has, at least, one allele that is present only in one parent (and at most there is only one missing allele in one of the parents); that allele is assigned to that parent and the other allele to the other one.

c) If a child has one missing allele, the other allele has been imputed from a homozygous parent, and so, the provenance of that allele will be known.

d) If a parent is homozygous, the allele has been necessarily transmitted to all the children, so this allele in the child is assigned to that parent, even if the other parent has missing alleles.

If the two alleles in the child are present in both parents (for example, the child has alleles *T/G* and both parents also have *T/G*), it is not possible to determine the provenance of the alleles.

III. If the origin of the alleles has been unequivocally determined, place the allele inherited from father in the first row of the matrix A_i and the allele inherited from the mother in the second row. Put the values of $IDS_{1ki} = IDS_{2ki} = 1$.

IV. Repeat steps 2 and 3 for all the markers in all the children in the family.

V. Compute the row sums of the IDS matrices for each child. Let c_1 be the child with the greatest value in the sum of the first IDS_i row, and c_2 the child with the highest value in the sum of the second row. Also, let $m_{11}, m_{12}, \ldots, m_{1l_1}$ be the set of markers with IDS=1 in the child c_1, and $m_{21}, m_{22}, \ldots, m_{2l_2}$ the set of markers with IDS=1 in the child c_2. These markers are already phased.

VI. In the father, sort the alleles in the markers $m_{11}, m_{12}, \ldots, m_{1l_1}$ in such a way that the first row of the matrix A_1 (allele matrix of the father) be equal to the first row of the matrix A_{c_1} (allele matrix of the child c_1) in the columns $m_{11}, m_{12}, \ldots, m_{1l_1}$. In the first row of IDS_1 put $IDS_{1k1} = 1$ for $k \in \{m_{11}, m_{12}, \ldots, m_{1l_1}\}$.

In the second row of IDS_1, for $k \in \{m_{11}, m_{12}, \ldots, m_{1l_1}\}$ put $IDS_{2k1} = 1$ if allele A_{2k1} is not missing and $IDS_{2k1} = 0$ if allele A_{2k1} is missing.

VII. In the mother, sort the alleles in the markers $m_{21}, m_{22}, \ldots, m_{2l_2}$ in such a way that the first row of the matrix A_2 (allele matrix of mother) be equal to the second row of the matrix A_{c_2} (allele matrix of the child c_2) in the columns $m_{21}, m_{22}, \ldots, m_{2l_2}$. In the first row of IDS_2 put $IDS_{1k2} = 1$ for $k \in \{m_{21}, m_{22}, \ldots, m_{2l_2}\}$. In the second row of IDS_2, for $k \in \{m_{21}, m_{22}, \ldots, m_{2l_2}\}$ put $IDS_{2k2} = 1$ if allele A_{2k2} is not missing and $IDS_{2k2} = 0$ if allele A_{2k2} is missing.

VIII. If all values in the matrices IDS_i are equal to 1, STOP. All the genotypes are phased. In other case proceed to update iteratively the matrices A_i and IDS_i following the procedure described below until there is no change in these matrices between two successive iterations. The objective of this procedure is essentially locating those alleles that are phased in children but not in parents and vice versa, and moving that information from one another.

a) Create the matrix *idHaps*, with two columns and as many rows as children in the family (n)

$$\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ \vdots & \vdots \\ h_{n1} & h_{n2} \end{bmatrix} \tag{10.4}$$

where, being $j = 1$ the father and $j = 2$ the mother:

$$h_{ij} = \begin{cases} 0 & \text{if haplotype } j \text{ provenance in child } i \text{ cannot be decided} \\ 1 & \text{if haplotype } j \text{ of child } i \text{ is the } \mathbf{1}^{st} \text{ haplotype in parent } j \\ 2 & \text{if haplotype } j \text{ of child } i \text{ is the } \mathbf{2}^{nd} \text{ haplotype in parent } j \end{cases}$$

This matrix is initially created comparing the alleles in A_i matrices with *IDS=1* (between parents and offspring). In the initial step it is expected that not all haplotypes could be unequivocally matched between parents and children and, so this matrix would contain some zeros.

b) Beginning with child 1 and proceeding to child n, for every child i:

1) For the row j ($j = 1, 2$) in the matrix IDS_i computes the difference vector $D_j = (D_{j1}, D_{j2}, \ldots, D_{jK})$ between such row and the row h_{ij} in the IDS matrix of the $j - th$ parent (IDS_j):
   - A value of -1 in the k position of D indicates that for marker j the corresponding parent haplotype has the allele correctly identified and phased, but not the child.
   - A value of 1 indicates that is the child who has that allele correctly identified and phased, but not the parent.
   - A value of 0 indicates that the allele is identified and phased in both (parent and child), or in none of them.
2) For those markers where $D_{jk} \neq 0$:
   - If $D_{jk} = -1$: update marker k in A_i (child allele matrix) and IDS_i (child IDS matrix) from the parent j allele matrix. This updating consists of checking if the actual allele in position jk of the matrix A_i (child) coincides with the allele in position k of the row h_{ij} in A_j (parent). Let a_c the child's allele, and a_p the parental allele. Let also be b_c the other child's allele in that marker.
   - If $D_{jk} = 1$ follow the same procedure as for $D_{jk} = -1$ but interchanging the roles of parent and child.
3) If, as a result of the previous step, the parents IDS matrices change, and there is any zero in the matrix *idHaps*, then this matrix must be revised to check if the unmatched haplotypes in children can be matched now with parental haplotypes, updating *idHaps* in consequence.

Within this scenario , all markers not in **Case 5** (see the description of configuration 5 in subsection 10.2) will be phased.

Note also that in step 2 in this last procedure, alleles **not previously imputed** by alleImputer may be imputed to parents and children simultaneously within the haplotyping process.

Scenario 2: There are missing markers in parents

In this scenario, new difficulties arise as previous processes depend on iteratively making constant comparisons (between the alleles present in parents and children). When parents have all their alleles missing this procedure cannot be addressed. But if there are at least three genotypically different children, it is possible to determine the haplotypes in children, it is not possible to know which haplotype comes from the father and which from the mother.

To understand how the procedure works, we must take into account that all haplotype combinations in parents (for which there may be at least three genotypically "unique" children) are the shown in Table 10.2. This table is similar to cases 5, 6 and 7 in Table 10.1, but while in such table ***genotypes*** were considered, now A, B, C and D are different ***haplotypes***. **All this is assuming that initially there are not missing alleles in children**.

	Case I	Case II	Case III
Parent 1	A/B	A/B	A/B
Parent 2	A/B	A/C	C/D
Possible Offspring	A|A A|B B|B	A|A A|B A|C B|C	A|C A|D B|C B|D

Table 10.2: Haplotype configurations in a parent-offspring pedigree when at least three genotypical different offspring are possible.

It is easy to see that for any combination of three children in either of the above cases, there is always, at least, a common shared haplotype. The idea for identifying haplotypes without knowing the parents begins with the identification of the shared haplotype between two children. Once this haplotype is identified, we would have also identified the complementary haplotypes. If the two selected children are heterozygous there are two different complementary haplotypes, so we would have identified a total of three different haplotypes. If one of the two children was homozygous, we had finished having two different haplotypes. In either of the cases, one of the identified haplotypes must be present also in the third child (as can be easily seen if we select any pair of children in Table 10.2).

In this way, the process to find haplotypes from three children is the following:

1. Given three children genotypically different (1, 2 and 3), for each marker $k = 1, \ldots, K$ find the common alleles between child 1 and 2, 1 and 3, and 2 and 3. When such common alleles exist, compose the corresponding haplotypes, and their corresponding complementary haplotypes. Denote by $H_{ij}^{(m)}$ a set of haplotypes found this way from children i and j. Note that in many cases, depending on the number of alleles in each marker, there will be more than one set of possible haplotypes (one common and one or two complementaries) derived from genotypes of children i and j. Also, sometimes if the selected children have haplotypes A/C and B/D, they will not have any common haplotype. In any case, if all $H_{ij}^{(m)}$ are empty then there is a genotyping error or a recombination, a ***Haplotype Incidence*** is then generated, and the process stops.

2. In other cases, for each $H_{ij}^{(m)} \neq \emptyset$ determine if at least one of the haplotypes, in the set $H_{ij}^{(m)}$, is also present in the third child:
   - If this condition is not fulfilled, an ***incidence*** is generated, and the process stops.
   - If there are more than one set of haplotypes that fulfill this condition (this can occur depending on the allele configuration, which can be organized conforming different haplotypes structures compatibles with 1 or more of the cases in Table 10.2), then there is no unique solution and the process stops without having phased the haplotypes.
   - If there is only one set of haplotypes H_{ij} fulfilling this condition, proceed to *step 3*.

3. If there are more than three children in the family, identify the set of haplotypes that have been found in *step 2* with one set of three offspring haplotypes in some of the cases depicted in Table 10.2 and determine the haplotypes in parent 1 and parent 2 (it is not possible to know which one is the father and which one is the mother). Match these parental haplotypes with the rest of the children, determining which particular haplotypes are present in each child. In case of the child to have missing alleles, impute them when possible.

4. Stop and return for each child his/her pair of phased haplotypes. This returns the A_i matrices of each child, with the alleles of each haplotype in one row, and set the IDS_i matrix to values equal to 1 (for those alleles that have been phased/imputed).

For proper operation of this process, it is necessary that at least three children have all their alleles not missing. The rest of the children (in case of the family having more than three) may have missing alleles. These alleles may be imputed to *step 3* if non-missing alleles in those children allow for a unique matching with the parental haplotypes. If children with missing alleles match with several of the parental haplotypes, the phase of the alleles in those children cannot be determined.

Scenario 3: Mixture of the previous scenarios

The process for this scenario is obviously more complicated than in the other two cases, and the haplotyping procedure is a mixture of the previous two.

1. If not all markers are completely missing in both parents apply the ***scenario 1***.
2. Count the total number of 1's in the matrices IDS_i for all family members. Let that number be *IDSNr*.
3. If there are markers completely missing in both parents, locate combinations of markers that have at least three children entirely genotyped without missing values, and that include at least one marker with completely missing parents. Let the set of such combinations be $S = \{S_1, S_2, \ldots, S_r\}$. All S_i such that $S_i \subseteq \bigcup_{j \neq i} S_j$ are removed from S.
4. If $S = \emptyset$ stops. If not, sort the sets S_i in decreasing order of size. For $i = 1$ to r:
   a) Apply the ***scenario 2*** to the markers in the set S_i. In particular, impute all missing alleles that are possible in children as indicated in *step 3* of such scenario. The idea is that the more complete are the markers, the better will work the processes of both scenarios.
   b) If there is one or more marker/s not completely missing in parents in S_i, use that/those marker/s for trying to *align* the parents: as we have seen, the ***scenario 2*** identifies parental

haplotypes although it does not distinguish between father and mother (not even using parental data). When it is possible to match these haplotypes with the non-missing markers, these may allow to determine which haplotype comes from the father and which one comes from the mother. If unique matching is possible, then impute the corresponding alleles in the missing markers of parents.

c) If it has not been possible to align the parents and there are phased markers outside S_i, try to match the haplotypes found for markers in S_i with the phased alleles outside S_i. If unique matching is possible, then impute the corresponding alleles in the parental missing markers.

d) If imputation has been made in *a)*, *b)* or *c)*, apply ***scenario 1*** to S_i.

5. Apply ***scenario 1*** to all markers in the family. Count the total number of 1's in the matrices IDS_i for all family members. Let *IDSNrNew* that number. If *IDSNrNew=IDSNr*, stop. In other cases, let *IDSNr=IDSNrNew* and go to *step 3*.

When this process stops the phased alleles are those with $IDS = 1$. If not all *IDS=1* the phasing process will be only partial.

Scenario 4: There are missing markers in parents and only two completely genotyped children

In this scenario, we consider that there are some markers (namely $Mk_1, Mk_2, \ldots, Mk_n$) for which parents and children have their alleles already phased. Let those identified haplotypes be (F_1, F_2) and (M_1, M_2), in the father and the mother, respectively. Consider now that, in another marker M, the genotypes of both parents are completely missing, but there are two children for whom that marker genotypes are known. Moreover, both children are heterozygote in such marker. There are four possible manners in which children could have inherited the haplotypes of the phased markers (F_1M_1, F_1M_2, F_2M_1, F_2M_2). For each one of these manners, the marker M genotypes in every child may appear in three alternative ways: ***two equal heterozygote children***, ***two heterozygote children sharing an allele*** or ***two heterozygote children without common alleles***. All the above considering that inherited haplotypes in the phased markers are the ones summarized in table 10.3.

Case	Description	Subject	Inherited haps. in phased markers	Inherited genotypes in unphased markers		
				Option 1	*Option 2*	*Option 3*
1	Children do not share any haplotype	Child 1	F_1, M_1	A/B	A/B	A/B
		Child 2	F_2, M_2	A/B	A/C	C/D
2	Children share a haplotype inherited from father	Child 1	F_1, M_1	A/B	A/B	A/B
		Child 2	F_1, M_2	A/B	A/C	C/D
3	Children share a haplotype inherited from mother	Child 1	F_1, M_1	A/B	A/B	A/B
		Child 2	F_2, M_1	A/B	A/C	C/D
4	Children share both haplotypes	Child 1	F_1, M_1	A/B	A/B	A/B
		Child 2	F_1, M_1	A/B	A/C	C/D

Table 10.3: Cases description of Scenario 4

1. If we suppose that $F_1 \neq F_2$, $M_1 \neq M_2$ and $\{F_1, F_2\} \cap \{M_1, M_2\} = \emptyset$, then:
   - In **case 1**: it is not possible to unequivocally identify the haplotypes that result from the combination of the phased markers $Mk_1, \ldots Mk_n$ with M, as each one of the haplotypes F_i y M_i has been observed once, and so any pairing between these haplotypes and the alleles in M would be compatible with the observed data.
   - In **case 2**:
     - With the ***option 1*** it is not possible to decide if F_1 is paired with A (and therefore (M_1, M_2) with B) or F_1 is paired with B (and therefore (M_1, M_2) with A). So marker M cannot be unequivocally phased.
     - With the ***option 2***, one paternal haplotype (F_1A) and two maternal haplotypes (M_1B and M_1C) are readily identified, as these are the unique haplotypes compatible with observed data.
     - ***Option 3*** cannot occur in this case without genotyping error or recombination.
   - **Case 3**: is analogous to case 2. Only with the ***option 2***, M_1A maternal haplotype and F_1B and F_1C paternal haplotypes would be compatible with observed data.

- In **Case 4**:
  - If ***option 1*** occurs, is not possible to determine one unique set of haplotypes. These could be F_1A and M_1B, as well as F_1B and M_1A.
  - ***Options 2*** and ***3*** cannot occur without genotyping error or recombination event.

2. If $F_1 = F_2$, $M_1 \neq M_2$ and $\{F_1\} \cap \{M_1, M_2\} = \emptyset$ then:
   - As before, in **case 1** is not possible an unequivocal identification of haplotypes.
   - In **case 2**, if $F_1 = F_2$, this haplotype could be accompanied by two different alleles in M, and therefore is not possible an unequivocal identification of haplotypes.
   - In **case 3**, ***option 2***, two haplotypes from the father and one from the mother can be determined: M_1A, F_1B, F_1C.
   - In **case 3**, ***option 3*** is not possible except by genotyping error or recombination.
   - As $F_1 = F_2$, **case 4** is equivalent to case 3.
3. If $F_1 \neq F_2$, $M_1 = M_2$ and $\{F_1, F_2\} \cap \{M_1\} = \emptyset$ then:
   - Again, an unequivocal identification of haplotypes is not possible for **case 1**.
   - In **case 2**, ***option 2***, the two maternal haplotypes and one in the father can be determined: F_1A, M_1B, M_2C.
   - In **case 2**, ***option 2*** is not possible except by genotyping error or recombination.
   - As $M_1 = M_2$, **case 4** is equivalent to case 2.
4. If $F_1 \neq F_2$, $M_1 \neq M_2$, $F_1 = M_1 = W_1$, $F_2 \neq M_2$, then:
   - In **case 1** it is possible to determine that the first child has the W_1A and W_1B haplotypes. Although is not possible to know which haplotype comes from the father and which one comes from the mother. This is valid for the three options.
   - In **case 2**, in the three options it is possible to identify the first child's haplotypes. In the ***option 2***, it is also possible to determine that the mother has haplotypes W_1B and M_2C, and that one of the haplotypes in the parent is W_1A. This is so because, in child 2, the haplotype W_1 is necessarily inherited from the father. When joining this haplotype with the

marker M, if the paternal haplotype in child 2 was W_1C, this would imply that, in child 1, the paternal haplotype should be W_1A or W_1B. But this is not possible because as the father has already two different haplotypes W_1 and F_2, it cannot be that there are two different paternal haplotypes beginning with W_1, as in this case counting with F_2 there would be a total of three haplotypes in father, which is impossible. Therefore necessarily the paternal haplotype must be W_1A and, by discarding maternal haplotypes, they are obtained.

- ***Option 3*** in **case 2** is not possible.
- **Case 3** is similar to case 2. In this case, besides it is possible to specify haplotypes in child 1 (although without knowing from which parent comes each one). In **option 2**, it is also possible to deduce that: one maternal haplotype is W_1A and two paternal haplotypes are F_2C and W_1B.
- In **case 4** it is feasible to determine the haplotypes of the two children, although without knowing which one comes from father and which one from mother. Moreover, only ***option 1*** is possible, as ***options 2*** and ***3*** only happen when there are genotyping errors or recombination events.

5. If $F_1 = M_1 = W_1$, $F_2 = M_2 = W_2$, $W_1 \neq W_2$ or $F_1 = F_2 = F$, $M_1 = M_2 = M$, $F = M = W$, then:
   - In the case of homozygous haplotypes for phased markers, haplotypes (including marker M) can be inferred in children but not in parents.
6. If $F_1 = F_2 = F$, $M_1 = M_2 = M$, $F \neq M$, or $F_1 = F_2 = F$, $M_1 \neq M_2$, $\{F\} \cap \{M_1, M_2\} = \emptyset$, or $F_1 \neq F_2 = F$, $M_1 = M_2 = M$, $\{F_1, F_2\} \cap \{M\} = \emptyset$, then:
   - In these conditions, complete haplotypes cannot be identified in parents nor offspring.

Summarizing this exhaustive analysis, the cases in which haplotypes can be unequivocally determined for a set of already phased markers $Mk_1, Mk_2, \ldots, Mk_n$ and a new marker M which has alleles completely missing for parents (and two children genotyped without missing values) are the shown in table 10.4. This table has been easily implemented in the form of an algorithm with several if-else conditions.

Parental haplotypes in phased markers	Inherited children's haplotypes in phased markers	Children's genotypes in unphased markers	Complete parental phased genotypes	Complete children's phased genotypes
$(F_1, F_2)\,, (M_1, M_2)$	(F_1, M_1), (F_1, M_2)	A/B, A/C	$(F_1A \mid F_2NA)$, $(M_1B \mid M_2C)$	$(F_1A \mid M_1B)$, $(F_1A \mid M_2C)$
$(F_1, F_2)\,, (M_1, M_2)$	(F_1, M_1), (F_2, M_1)	A/B, A/C	$(F_1B \mid F_2C)$, $(M_1A \mid M_2NA)$	$(F_1B \mid M_1A)$, $(F_2C \mid M_1A)$
$(F, F)\,, (M_1, M_2)$	$(F, M_1)\,, (F, M_1)$	A/B, A/C	$(FB \mid FC)$, $(M_1A \mid M_2NA)$	$(FB \mid M_1A)\,, (FC \mid M_1A)$
$(F_1, F_2)\,, (M, M)$	(F_1, M), (F_1, M)	A/B, A/C	$(F_1A \mid F_2NA)$, $(MB \mid MC)$	$(F_1A \mid MB)$, $(F_1A \mid MC)$
(W_1, F_2), (W_1, M_2)	(W_1, W_1), (W_1, M_2)	A/B, A/C	$(W_1A \mid F_2NA)$, $(W_1B \mid M_2C)$	$(W_1A \mid W_1B)$, $(W_1A \mid M_2C)$
(W_1, F_2), (W_1, M_2)	(W_1, W_1), (F_2, W_1)	A/B, A/C	$(W_1B \mid F_2C)$, $(W_1A \mid M_2NA)$	$(W_1B \mid W_1A)$, $(F_2C \mid W_1A)$

Table 10.4: Arrangement of those cases in which is possible to phase alleles in an unphased marker with parental missing data (using the information of phased adjacent markers' alleles of two children).

10.3.4.3. alleHaplotyper Function

alleHaplotyper creates the haplotypes family by family taking into account the previously imputed genotypes, along with the matrix IDS. In order to generate the haplotypes, this function has been developed in six steps:

I. **Internal Functions**: In this step, numerous functions to were implemented, being the most important ones:
   - famHaplotyper, which develops the haplotype reconstruction for each family data as follows:
     1) Receives as input data the matrix of imputed data returned by *alleImputer* for **one** family.
     2) Applies the algorithms described in scenario 3 (note that this algorithm also adapts to scenarios 1 and 2) or in scenario 4 according to the availability of children and genotypic information.
     3) Returns: *a)* a matrix equals to the input one, but with the new imputed alleles, *b)* a matrix with the same dimensions as the previous one filled with 0's an 1's. The zero value indicates a non-phased allele, and the 1 a phased one, and *c)* other matrix with two columns corresponding to the haplotypes found in each member of the family.
   - famsHaplotyper, which applies famHaplotyper sequentially to all families.
   - summarizeData, which generates a summary of the haplotyping process.

II. **Re-Imputation**: This step calls the alleImputer function which performs the imputation marker by marker and then, family by family.

III. **Haplotyping**: This part is the most important of alleHaplotyper, since it tries to solve the haplotypes when possible. The process is the following: once each family genotype has been imputed marker by marker, those markers containing two unique heterozygous alleles (both in parents and offspring) are excluded from the process. Then, an IDentified/Sorted (IDS) matrix is generated per family. Later, the internal function famHaplotyper tries to

solve the haplotypes of each family, comparing the information between parents and children in an iterative and reciprocal way. When there are not genetic data in both parents and there are two or more "unique" offspring (not twins or triplets), the internal functions *makeHapsFromThreeChildren* and *makeHapsFromTwoChildren* try to solve the remaining data. Finally, the HoMoZygosity (HMZ) matrix is updated, and the excluded markers are again included. *Even if both parental alleles are missing in each marker, it is possible to reconstruct the family haplotypes, identifying the corresponding children's haplotypes, although* ***in certain cases their parental provenance will be unknown***.

IV. **Data Summary**: Once the data haplotyping is done, a data summary is collected, containing the re-imputation rate (after the haplotyping process), the proportions of phased and non-phased alleles, the proportion of full, partial and empty reconstructed haplotypes, and the time employed in the process.

V. **Data Storing**: In this step, the re-imputed data are stored in the same path where the PED file was located, as long as, data have been read from an external file. Two new files will be generated with the same name and extension as the original, but ending as *'re-imputed.ped'* and *'haplotypes.txt'*, for the re-imputed genotypes and for the reconstructed haplotypes, respectively.

VI. **Function Output**: In this final step, a summary of the generated data may be printed out, if *dataSummary=TRUE*. All the results can be directly returned, whether *invisibleOutput* is deactivated.

The list of results contains: *imputedMkrs* (which contains the preliminary imputed marker alleles), *IDS* (which includes the resulting IDentified/Sorted matrix), *reImputedAlls* (which includes the re-imputed alleles) and *haplotypes* (which stores the reconstructed haplotypes) and *haplotypingSummary* (which shows a summary of the haplotyping process).

Incidence messages can also be shown if they are detected. These may be caused by *haplotype recombination* (detected on children), *genotyping errors* or *inheritance from non-declared parents*.

Full examples of alleHaplotyper function are provided at section B.3.3.1 of Appendix B.

Chapter 11

alleHap Package: Performance

In this chapter, a benchmarking of alleHap is performed. The tasks to evaluate were: computing times, genotype imputation rates and proportion of completely reconstructed haplotypes.

11.1. Computing Times

In the evaluation of the corresponding computational times, three factors have been taken into account: number of families, number of markers and number of different or unique *(non-repeated)* alleles per marker in each population.

11.1.1. Computing Times per Number of Families

The simulations to evaluate ***computational times*** of alleHap according to the ***number of families*** have been developed considering the following factors:

```
nFams = Number of families to generate: [1,...,2000]
nChildren = Number of children per family: 2
nMarkers = Number of markers (allele pairs) to generate: 3
numAllperMrk = Number of different alleles per marker: 2
chrAlleles = Alleles expressed as characters (A,C,G,T): TRUE
nHaplos = Number of different haplotypes in the population: 1200
missParProb = Probability of parents' missing genotype: 0.25
missOffProb = Probability of children's missing genotype: 0.25
phenProb = Phenotype probability: 0.2
recombRate = Recombination rate: 0
```

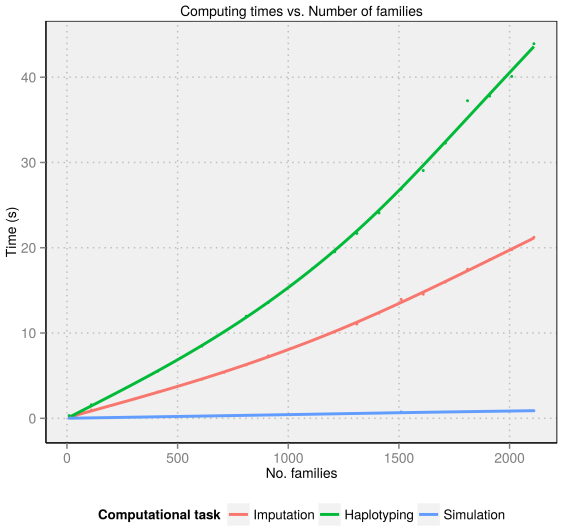


Figure 11.1: *Computing times for Simulation, Imputation and Haplotyping, depending on the number of families.* Computing times per families.

In the previous figure, it can be seen how reconstruction and imputation times grow linearly as the number of families increases while simulation times hardly grow. Therefore, it can be said that when using a small number of markers, alleHap consumes very little time, even considering a high number of individuals ($\geq$ 8000).

11.1.2. Computing Times per Number of Markers

The simulations to evaluate ***computational times*** of alleHap depending on the ***number of markers*** have been developed considering the following characteristics:

```
nFams = Number of families to generate: 1
nChildren = Number of children per family: 2
nMarkers = Number of markers (allele pairs) to generate: [1,...,5000]
numAllperMrk = Number of different alleles per marker: 2
chrAlleles = Alleles expressed as characters (A,C,G,T): TRUE
nHaplos = Number of different haplotypes in the population: 1200
missParProb = Probability of parents' missing genotype: 0.25
missOffProb = Probability of children's missing genotype: 0.25
```

```
phenProb = Phenotype probability: 0.2
recombRate = Recombination rate: 0
```

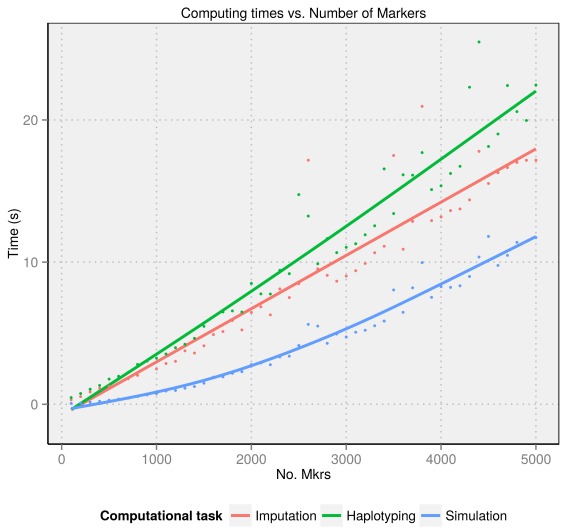


Figure 11.2: *Computing times for Simulation, Imputation and Haplotyping, depending on the number of markers.* Computing times per markers.

Above figure shows how simulation times grow almost exponentially as the number of markers increase, while imputation and reconstruction times remain linear, even considering a great number of markers ($\geq$ 5000). Although it is true that the analysis only has taken into account one family (containing four individuals).

11.1.3. Computing Times per Number of Alleles

The simulations to evaluate the alleHap ***computational times*** in function of the ***number of unique alleles per marker*** have been developed considering the following features:

```
nFams = Number of families to generate: 1000
nChildren = Number of children per family: 2
nMarkers = Number of markers (allele pairs) to generate: 3
numAllperMrk = Number of different alleles per marker: [2,...,268]
```

```
chrAlleles = Alleles expressed as characters (A,C,G,T): FALSE
nHaplos = Number of different haplotypes in the population: 1200
missParProb = Probability of parents' missing genotype: 0.25
missOffProb = Probability of children's missing genotype: 0.25
phenProb = Phenotype probability: 0.2
recombRate = Recombination rate: 0
```

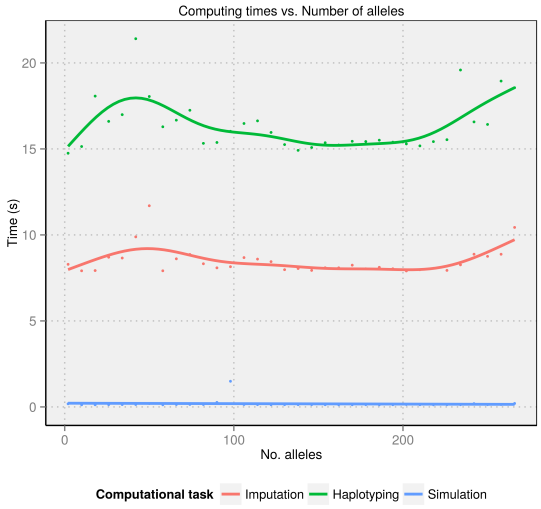


Figure 11.3: *Computing times for Simulation, Imputation and Haplotyping, depending on the number of unique alleles per marker.* Computing times per number of unique alleles.

The previous figure represents how computing times remain stable independently the number of unique alleles per marker and the computational task, either simulation, imputation or reconstruction.

After analysing all the computing times of alleHap, it can be established that the performance of the algorithm is proportional to the number of families $\{n_{Fams}\}$, as well as to the number of individuals $\{(n_{Children} + 2) * nFams\}$ and the number of markers $\{n_{Markers}\}$. So, an approximate computing time could be:

$$E_{Run\ time} = (n_{Children} + 2) \times n_{Fams} \times n_{Markers}$$

The evaluation of results reveals an optimum performance of the alleHap computational tasks, namely ***simulation***, ***imputation*** or ***haplo-***

typing, The resulting execution times were quite low, even when considering a large number of families (≤ 2000), subjects (≤ 8000) and markers (≤ 5000).

11.2. Genotype Imputation Rates

The generated imputation rates were evaluated taking into account different aspects. Firstly, a comparison between initial and final imputation rates was performed. Initial imputation rate corresponds to the imputation task carried out by alleImputer for individual markers. Final imputation rate refers to the imputation rate achieved after using alleHaplotyper, which imputes additional markers when identifies complete haplotypes. Subsequently, an analysis of the imputation rates, depending on the probability of missing genotypes, number of *different* alleles per marker and number of markers, has been developed.

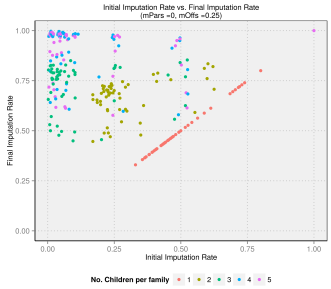
11.2.1. Initial vs. Final Imputation Rates

Initial imputation rates (generated by the alleImputer function) and ***final imputation rates*** (by the alleHaplotyper function) are compared in Figure 11.4, taking into account several characteristics, as the number of children, the number of markers, the number of alleles per marker and the probability of missing genotypes (either in parents or in offspring).

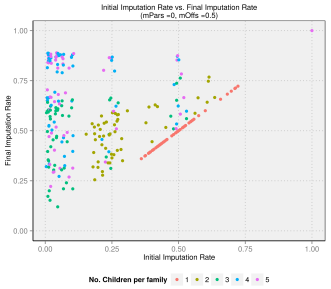
The illustrations in Figure 11.4 show how imputation rates decrease as the proportion of missing genotypes in children increases. For only one child in the family, there is no difference between initial and final imputation rates, because, in this case, the function alleHaplotyper is inactive. However, it is clear that final imputation rates improve the initial ones in those families that have more than one child, since in that case alleHaplotyper comes into action, improving its performance with the number of children. Moreover, for a probability of missing genotypes in children less than 0.25, it can be seen how a good final imputation rates ($\geq$ 0.5, and even near to 1) are achieved in families composed of two or more children.

11.2.2. Imputation Rates vs. Missing Genotypes

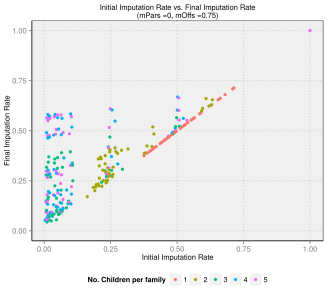
The ***final imputation rates*** as function of the ***probability of missing genotypes*** (either in parents or in offspring), depending on the number of markers, are arranged in Figures 11.5 and 11.6.



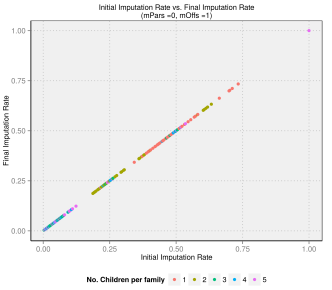
(a) mPars = 0; mOffs = 0.25



(b) mPars = 0; mOffs = 0.5

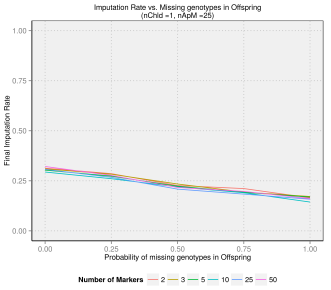


(c) mPars = 0; mOffs = 0.75

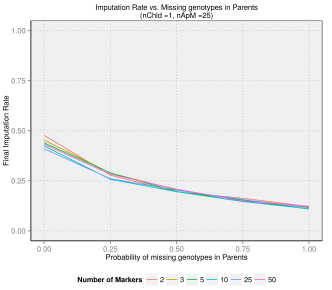


(d) mPars = 0; mOffs = 1

Figure 11.4: Initial vs. final imputation rates when the probability of missing genotypes in parents is zero, i.e. there are no parental missing values.

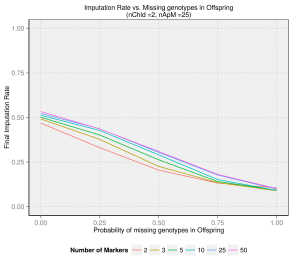


(a) nChild = 1; mPars = 0.5

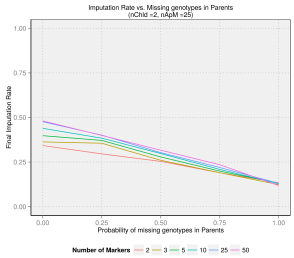


(b) nChild = 1; mOffs = 0.5

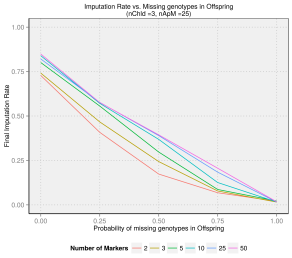
Figure 11.5: Final imputation rates vs. missing genotypes, depending on the number of children per family.



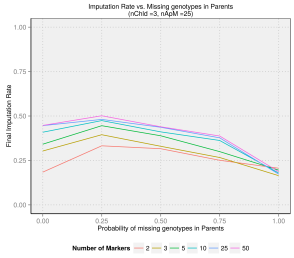
(a) nChild = 2; mPars = 0.5



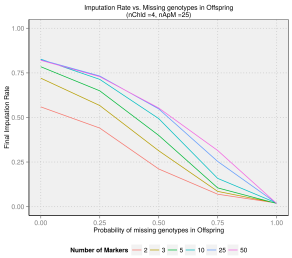
(b) nChild = 2; mOffs = 0.5



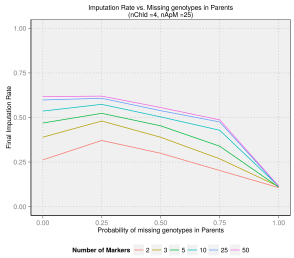
(c) nChild = 3; mPars = 0.5



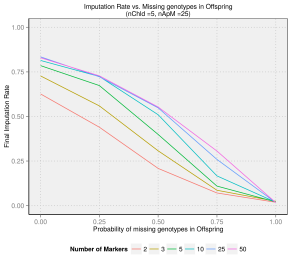
(d) nChild = 3; mOffs = 0.5



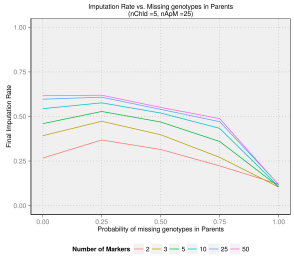
(e) nChild = 4; mPars = 0.5



(f) nChild = 4; mOffs = 0.5



(g) nChild = 5; mPars = 0.5

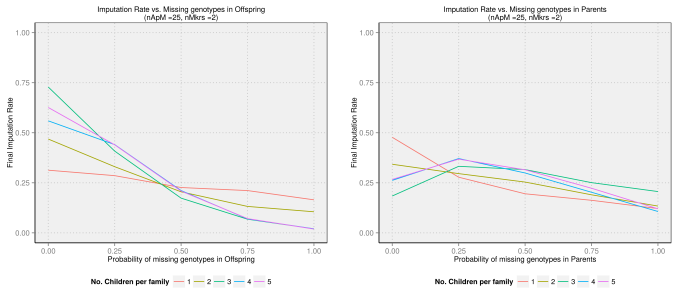


(h) nChild = 5; mOffs = 0.5

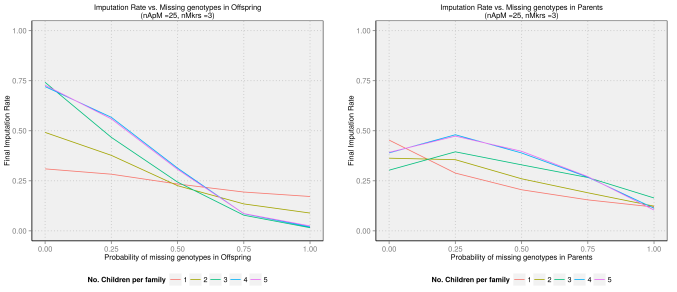
Figure 11.6: Final imputation rates vs. missing genotypes, depending on the number of children per family.

The illustrations in Figures 11.5 and 11.6 show that for a *number of markers* of 25 or more, the final imputation rates tend to stabilize, whichever the number of children per family or the proportion of missing values (either in parents or offspring) is. On the other hand, it can also be seen that the greater the proportion of missing values, the lower the rates of imputation achieved. Finally, it can be noted that when evaluating families containing 3 or more children (with a number of unique alleles per marker equals to 25) the imputation rates are better, regardless of the number of considered markers.

The ***final imputation rates*** as a function of the ***probability of missing genotypes*** (either in parents or in offspring), depending on the number of children per family, are arranged in Figures 11.7 and 11.8.

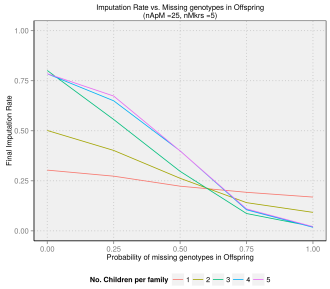


(a) nMarkers = 2; mPars = 0.5 (b) nMarkers = 2; mOffs = 0.5

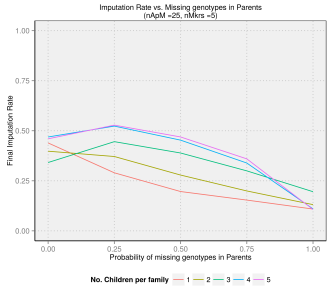


(c) nMarkers = 3; mPars = 0.5 (d) nMarkers = 3; mOffs = 0.5

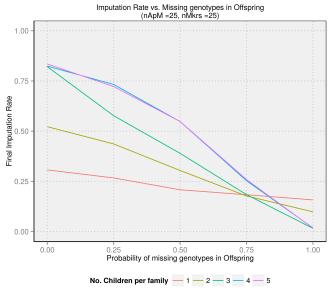
Figure 11.7: Final imputation rates vs. missing genotypes, depending on the number of markers.



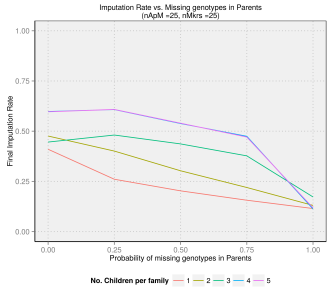
(a) nMarkers = 5; mPars = 0.5



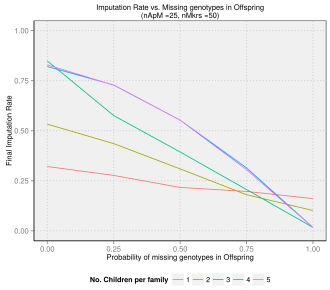
(b) nMarkers = 5; mOffs = 0.5



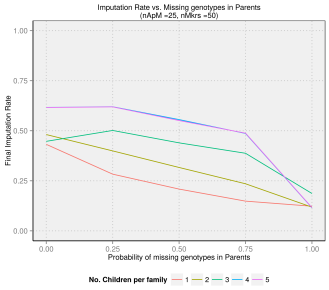
(c) nMarkers = 25; mPars = 0.5



(d) nMarkers = 25; mOffs = 0.5



(e) nMarkers = 50; mPars = 0.5



(f) nMarkers = 50; mOffs = 0.5

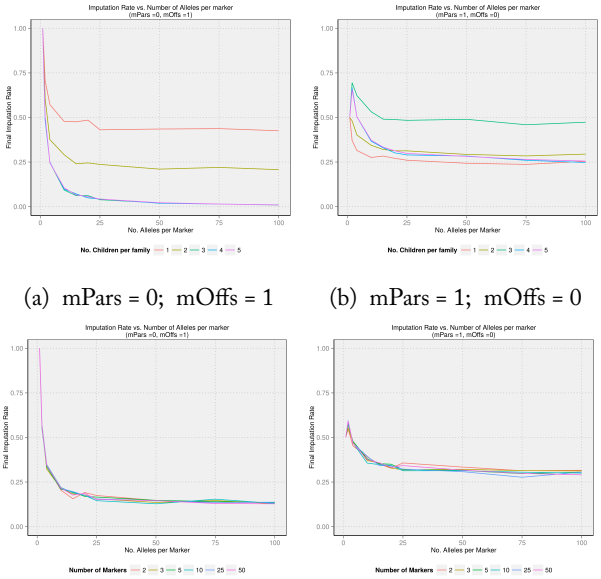
Figure 11.8: Final imputation rates vs. missing genotypes, depending on the number of markers.

The illustrations in Figure 11.7 and 11.8 show that from a *number of children per family* of 4 or more, the imputation rates tend to stabilize. Furthermore, regardless the number of markers, imputation rates also remain constant for families composed of 3 or 4 members (with 1 or 2 children), when considering a fixed proportion of missing genotypes in parents. However, when taking into account a variable proportion of parental missing values, there is a more pronounced difference among the imputation rates as the number of considered markers increase (although they become again stable from 25 markers onwards). When considering a fixed missing genotype probability in parents (0.5), figures labelled (a), (c) and (e), when the probability of missing genotypes in children is low, a better imputation rate is achieved when more children are included in the family, because alleHaplotyper combines the information in parents and children (which in this case have a low number of missing alleles) to identify the haplotypes and consequently impute new alleles. As the initial number of missing alleles is low, the imputed alleles this way represent a great proportion of the initially missing. But as the missing genotype probability grows in children, there is less information for alleHaplotyper to use, so many alleles remain without imputing. As in the initial situation there were many missing alleles, the final one experiences little improvement in proportion. When fixed missing probability is considered in children (0.5), figures labelled (b), (d) and (f), we see that whichever is the probability of missing genotype in parents, the best results are obtained for 3 or more children. This is because the algorithm can use information in three children to impute alleles in parents, find the corresponding haplotypes and (maybe) impute alleles in the fourth, fifth, and rest of the children.

11.2.3. Imputation Rates vs. Number of Alleles

The ***final imputation rates*** as a function of the ***number of alleles per marker***, depending on the number of children per family and the number of markers, are arranged in Figure 11.9.

The illustrations in Figure 11.9 show that from a *number of alleles per marker* of 25 or more, the imputation rates tend to stabilize, independently of the proportion of missing genotypes. When parental data is complete *(mPars=0)*, even with ungenotyped children *(mOffs=1)*, imputation rates of almost 0.5 and 0.25 can be achieved for families with 1 or 2 children, respectively (see Figure 11.9a). When considering ungenotyped parents *(mPars=1)* and completely genotyped offspring *(mOffs=1)*, all imputation rates are above 0.25 (see Figure 11.9d).



(a) mPars = 0; mOffs = 1 (b) mPars = 1; mOffs = 0

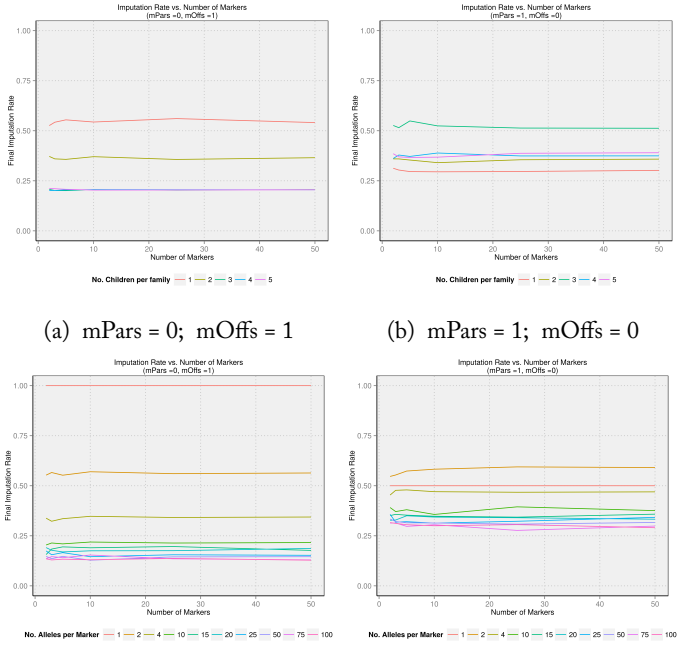
(c) mPars = 0; mOffs = 1 (d) mPars = 1; mOffs = 0

Figure 11.9: Final imputation rates vs. number of unique alleles per marker, depending on the number of children and markers.

11.2.4. Imputation Rates vs. Number of Markers

The ***final imputation rates*** as a function of the ***number of markers***, depending on the number of children per family and the number of unique alleles per marker, are arranged in Figure 11.10.

The illustrations in Figure 11.10 show that all imputation rates remain constant, independently of the number of markers. When parental data is complete *(mPars=0)*, even with ungenotyped children *(mOffs=1)*, imputation rates of more than 0.5 and 0.30 can be achieved for families with 1 and 2 children, respectively (see Figure 11.10a). With the previous proportion of missing genotypes, when all population individuals were homozygous *(nApM=1)*, maximum imputation rates *(all equal to 1)* were achieved. When population individuals were heterozygous (with two unique and different alleles per marker) more than half imputation rates *(≥ 0.5)* were obtained (see Figure 11.9c).



(a) mPars = 0; mOffs = 1

(b) mPars = 1; mOffs = 0

(c) mPars = 0; mOffs = 1

(d) mPars = 1; mOffs = 0

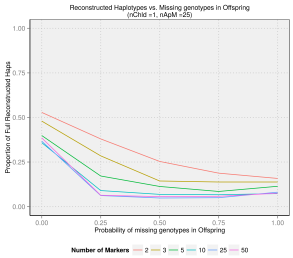
Figure 11.10: Final imputation rates vs. number of markers, depending on the number of children and unique alleles per marker.

11.3. Reconstructed Haplotypes

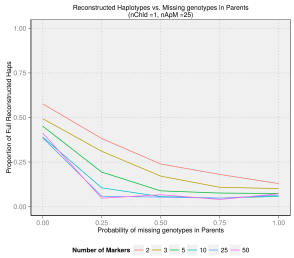
The proportion of completely reconstructed haplotypes is evaluated taking into account the probability of missing genotypes, the number of *different* alleles per marker in population and the number of markers.

11.3.1. Reconstructed Haplotypes vs. Missing Genotypes

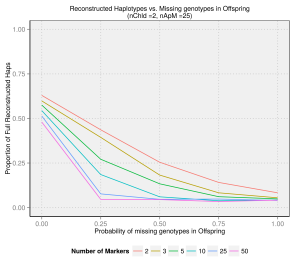
The ***proportion of completely reconstructed haplotypes*** versus the ***probability of missing genotypes*** (either in parents or in offspring), depending on the number of children per family, is depicted in Figure 11.11.



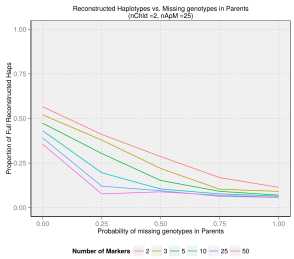
(a) nChild = 1; mPars = 0.5



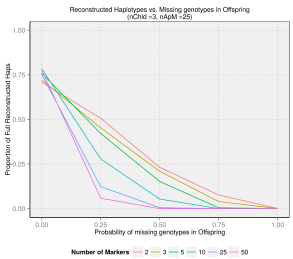
(b) nChild = 1; mOffs = 0.5



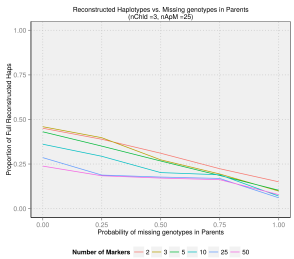
(c) nChild = 2; mPars = 0.5



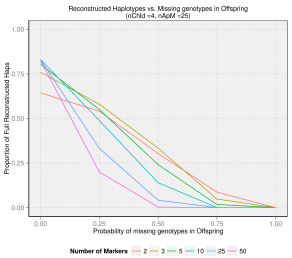
(d) nChild = 2; mOffs = 0.5



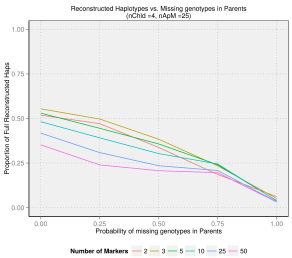
(e) nChild = 3; mPars = 0.5



(f) nChild = 3; mOffs = 0.5



(g) nChild = 4; mPars = 0.5

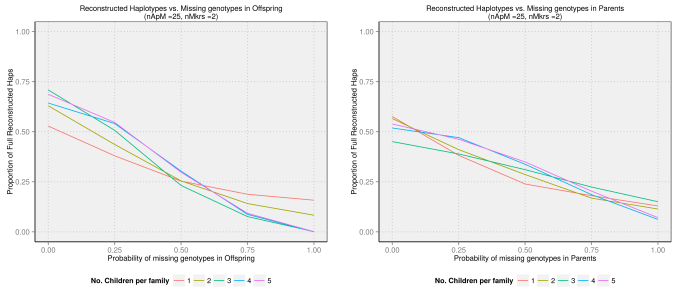


(h) nChild = 4; mOffs = 0.5

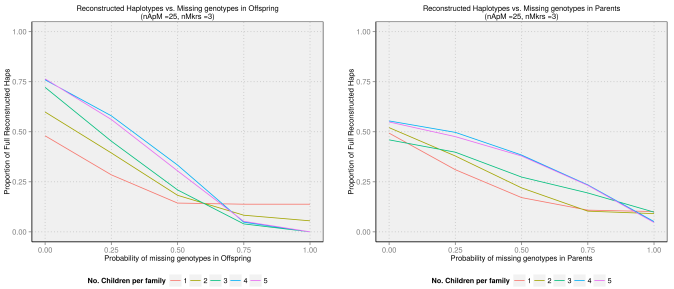
Figure 11.11: Proportion of completely reconstructed haplotypes vs. missing genotypes, depending on the number of children per family.

The illustrations in Figure 11.11 show that from a *number of children per family* of 3 or more, the same pattern is shared: an abrupt decay of the proportion of reconstructed haplotypes from 0.75 to almost 0 when varying the missing genotypes in offspring. However, if we look parental missing values, these range is smaller oscillating between 0.5 and 0.25 (corresponding to a probability of missing values of 0 and 0.75, respectively).

On the other hand, comparing the ***proportion of full reconstructed haplotypes*** regarding the ***probability of missing genotypes*** (either in parents or in offspring), depending on the number of Markers we can obtain the Figure 11.12.

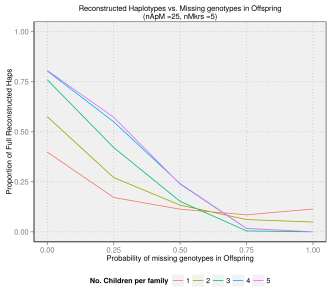


(a) nMarkers = 2; mPars = 0.5 (b) nMarkers = 2; mOffs = 0.5

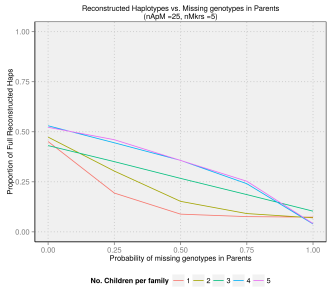


(c) nMarkers = 3; mPars = 0.5 (d) nMarkers = 3; mOffs = 0.5

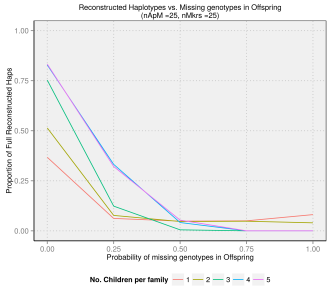
Figure 11.12: Completely reconstructed haplotypes vs. missing genotypes, depending on the number of markers.



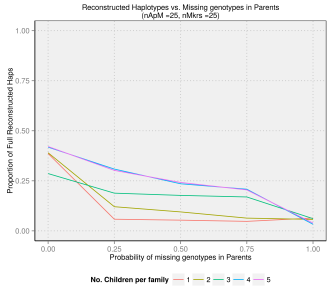
(a) nMarkers = 5; mPars = 0.5



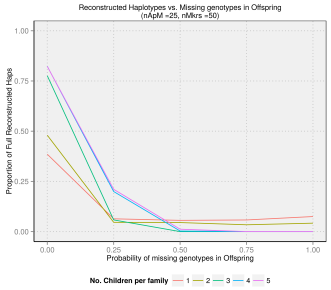
(b) nMarkers = 5; mOffs = 0.5



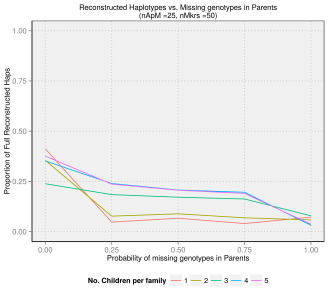
(c) nMarkers = 25; mPars = 0.5



(d) nMarkers = 25; mOffs = 0.5



(e) nMarkers = 50; mPars = 0.5



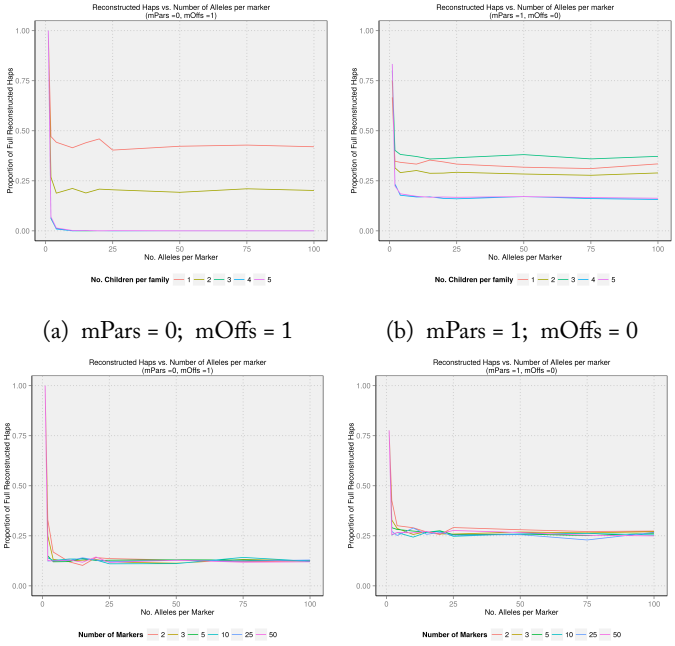
(f) nMarkers = 50; mOffs = 0.5

Figure 11.13: Completely reconstructed haplotypes vs. missing genotypes, depending on the number of markers.

The illustrations in Figure 11.12 and 11.13 and show that from a *number of children per family* of 4 or more, the proportion of reconstructed haplotypes tend to stabilize. Furthermore, there are more reconstructed haplotypes in families composed of 3 or 4 members (with 1 or 2 children) when considering a proportion of missing genotypes in offspring greater than 0.65 (for 5 or fewer markers) or 0.5 (for more than 25 markers). Moreover, it can be seen that the proportion of completely reconstructed haplotypes decays as increases the number of markers.

11.3.2. Reconstructed Haplotypes vs. Number of Alleles

The ***proportion of completely reconstructed haplotypes*** in function of the ***number of alleles per marker***, depending on the number of children per family and the number of markers, is shown in Figure 11.14.



(a) mPars = 0; mOffs = 1 (b) mPars = 1; mOffs = 0

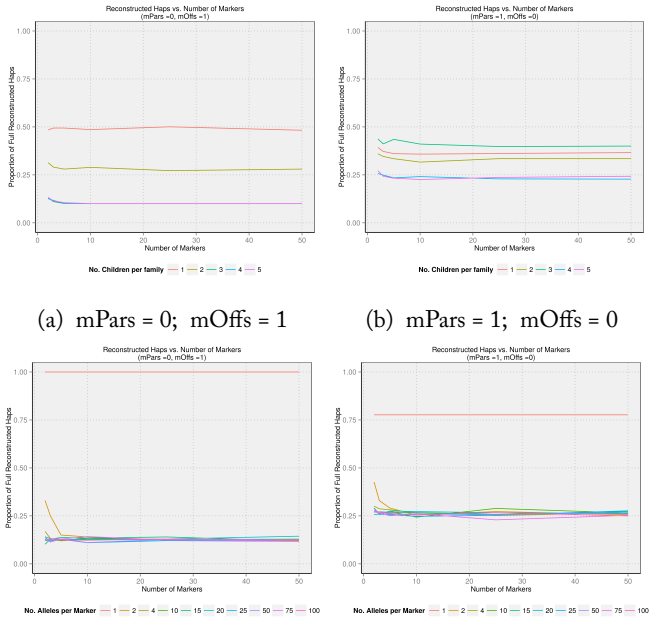
(c) mPars = 0; mOffs = 1 (d) mPars = 1; mOffs = 0

Figure 11.14: Proportion of completely reconstructed haplotypes vs. no. alleles per marker, depending on the number of children and markers.

The illustrations in Figure 11.14 show that from a regarding the *number of alleles per marker*, the proportion of reconstructed haplotypes is very stable. When parental data is complete *(mPars=0)*, even with ungenotyped children *(mOffs=1)*, haplotypes are reconstructed in a proportion about 0.4 and 0.2 can be achieved for families with 1 or 2 children, respectively (see Figure 11.14a). When considering ungenotyped parents *(mPars=1)* and completely genotyped offspring *(mOffs=1)*, all haplotypes are entirely reconstructed in a proportion of 0.25, regardless the number of considered markers (see Figure 11.14d).

11.3.3. Reconstructed Haplotypes vs. Number of Markers

The ***proportion of completely reconstructed haplotypes*** in function of the ***number of markers***, depending on the number of children per family and the number of unique alleles per marker, is depicted in Figure 11.15.



(a) mPars = 0; mOffs = 1 (b) mPars = 1; mOffs = 0

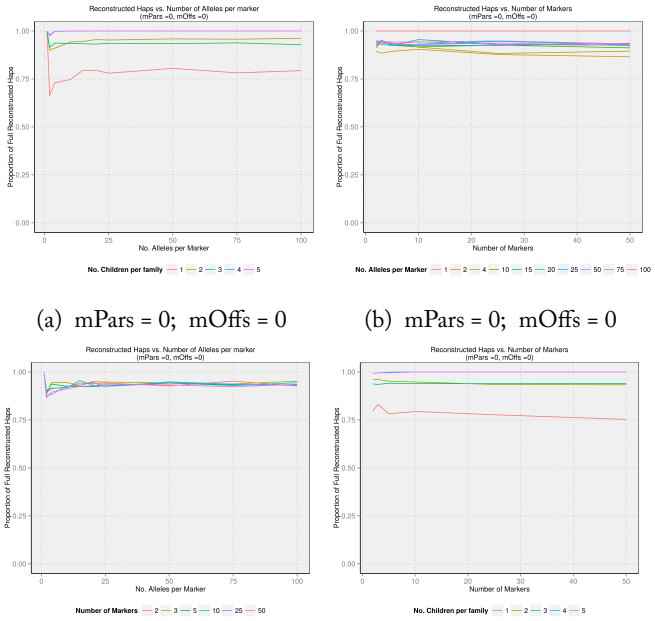
(c) mPars = 0; mOffs = 1 (d) mPars = 1; mOffs = 0

Figure 11.15: Proportion of completely reconstructed haplotypes vs. no. markers, depending on number of children and alleles per marker.

The illustrations in Figure 11.15 show how the proportion of reconstructed haplotypes remains constant, independently of the number of markers. When parental data is complete *(mPars=0)*, even with ungenotyped children *(mOffs=1)*, this proportion about to 0.5 and 0.30 can be achieved for families with 1 and 2 children, respectively (see Figure 11.15a). With the previous proportion of missing genotypes, when all population individuals were homozygous *(nApM=1)*, all haplotypes were completely reconstructed. For the rest of the cases, this proportion seems to stabilize around the value of 0.15 (see Figure 11.14c).

11.3.4. Reconstructed Haplotypes without missing genotypes

The ***proportion of completely reconstructed haplotypes*** without ***missing values in genotypes***, depending on the number of children per family and the number of unique alleles per marker, is shown in Figure 11.16.



(a) mPars = 0; mOffs = 0 (b) mPars = 0; mOffs = 0

(c) mPars = 0; mOffs = 0 (d) mPars = 0; mOffs = 0

Figure 11.16: Proportion of completely reconstructed haplotypes vs. no. markers, depending on the number of children and alleles per marker.

In the illustrations arranged in Figure 11.16 can be noted how haplotypes are entirely reconstructed, at least, in a proportion of 0.75, independently of the considered number of markers or the number of alleles per marker (see Figure 11.16a). Furthermore, when the number of children per family is 2 or more this proportion can be greater than 0.9, i.e. at least, a 90% of the haplotypes can be completely reconstructed within families composed of two children or more (see Figures 11.16a and 11.16d).

Chapter 12

alleHap Package: Applications

12.1. alleHap into T1DGC database

This chapter will describe some applications of the alleHap package, as the analysis and identification of offspring and maternal factors associated with T1D early and childhood onset, and the comparison of the distributions of risk haplotypes between the Canary Islands and the rest of Spain and Europe.

12.1.1. Parental diabetes and other predictive factors for T1D early and childhood onset in offspring

Type 1 diabetes is a clinically heterogeneous disease, triggered by several unknown environmental factors in genetically susceptible individuals, with a course ranging from early, aggressive destruction of beta-cells to slow progression, where patients need insulin months to years after diagnosis. In Europe, the incidence rate of childhood T1D increases 3–4% per year, with a 6.3% increase for children aged 0–4 years, 3.1% for children aged 5–9 years, and 2.4% for those 10–14 years [170]. From 2005 to 2020 the number of children of <5 years at onset will double and the onset <15 years will rise by 70% [171].

There are some differences between childhood-onset and adult-onset T1D. Childhood onset of T1D is associated with more frequent ketosis/ketoacidosis, poorly preserved residual beta-cell function, strong humoral autoimmunity against islet cells and insulin, and strong HLA-defined disease susceptibility, a higher frequency of preceding infec-

tions, shorter duration of symptoms and more independence of seasonal trigger mechanisms than adult onset of the disease, pointing to a more aggressive form of diabetes, [172, 173, 174]. Male predominance is a further, unexplained feature of Type 1 diabetes in young adults according to some studies [175].

The HLA region is the most important genetic determinant of the susceptibility to type 1 diabetes [176]. According to *Gillespie et al.* [177], in the UK more than 90% of children with type 1 diabetes carry the HLA class II haplotypes DRB1*03-DQB1*02:01 (DR3-DQ2) and/or DRB1*04-DQB1*03:02 (DR4-DQ8), and the highest risk DR3-DQ2/DR4-DQ8 diplotype is present in 50% of cases of very early–onset diabetes.

Diabetes-associated autoantibodies can be used as markers of T1D for young subjects with increased genetic disease susceptibility, and an early age of onset of T1D has been found to be associated with certain high-risk HLA haplotypes, found more frequently in T1D children diagnosed < 5 years of age than in those diagnosed when older [178, 177].

Studies in pairs of twins suggest that much of the variability of the age of T1D onset is genetically determined [179], for example, by SNPs in the IL2RA gen in patients of adult onset [180]. Age of onset can be considered as an indicator of genetic susceptibility, with an earlier onset of the disease related to stronger genetic components and thus a higher risk for the first-degree relatives [177, 181]. But the T1D high increase of the very young cannot be exclusively due to changes in the population gene pool, it rather suggests an early influence of environmental factors, for example, epigenetic modifications, already during the perinatal period. Several studies analysed the influence of maternal factors on diabetes risk and demonstrated an association between parental age at birth and increased risk of childhood T1D, whereas increased birth order was associated with a significant decrease in the risk of the disease [182]. Childhood but not the adult age of onset of the father seems to affects offspring onset while only mothers with an onset before <10 years of age affect the diabetes risk of their sons (not of the daughters) [183, 184]. The onset of diabetes before or after pregnancy did not affect the diabetes risk of offspring in a different manner in mothers with adult onset [184].

The T1DGC is an international effort aimed at the study of the genetics and pathogenesis of T1D. With thousands of families with T1D included from all over the world, this collection represents an extraor-

dinary resource, not only for samples and genetic data but also of associated clinical information. The database (assembled in January 2009) contains information on 14494 subjects in 3275 families, from which 2849 includes, at least, two T1D affected siblings, and 426 contain only one affected sib. There is a total of 6271 affected siblings and 1673 unaffected ones in the database. In the parents, 194 fathers and 85 mothers are affected by T1D. Age of onset is available for all the affected sibs and some of the affected parents (130 fathers and 67 mothers).

The subjects in the T1DGC database have been recruited in four regions: Asia-Pacific (561 families, 2289 subjects), Europe (excluding the UK, 1221 families, 5502 subjects), North America (1330 families, 5967 subjects, and the UK (163 families, 736 subjects)

The database contains allele information from several markers in the MHC-HLA complex, in particular, HLA-A, HLA-B, HLA-CW, HLA-DPA, HLA-DPB, HLA-DQA, HLA-DQB, HLA-DRB, as well as CTLA4 and insulin-HPH SNPs. Alleles at these markers are complete for 12370 subjects in the database: 2215 fathers and 2651 mothers, as well as 6005, affected sibs and 1499 unaffected sibs (the remaining 2124 subjects have completely missing genotypes. We have used alleHap to identify HLA haplotypes from this dataset. Haplotypes comprising markers DRB-DQA-DQB are known to be related to the risk of T1D. Among the 12370 fully genotyped individuals, 11095 (89.7%) could be fully haplotyped for these markers, and 493 (4%), partially haplotyped. 782 (6.3%) subject could not be univocally haplotyped. Among the 2124 subjects with completely missing genotypes, 1222 (57.5%) also had completely missing haplotypes, alleHap allowed partial and complete haplotyping in 53 (2.5%) and 849 (40%) of these subjects, respectively.

The aim of our study was to find maternal factors associated with early and childhood onset of T1D such as predictors of this form of the disease in offspring in the T1DGC dataset available on the 1^{st} October 2009.

12.1.1.1. Distribution of the number of High Risk haplotypes in T1DGC database

Table 12.1 show the frequency distribution of the number of high risk DR3-DQ2 and DR4-DQ8 haplotypes depending on whether the subjects have the disease or not. As can be seen, globally 60% of the affected subjects carry two risk haplotypes versus only 35.3% in the non-

	Number of high risk haplotypes		
T1D	**0**	**1**	**2**
Global Data:			
No	1544 (19.7%)	3519 (45%)	2761 (35.3%)
Yes	620 (9.5%)	1997 (30.5%)	3933 (60%)
Asia-Pacific:			
No	311 (23.2%)	517 (38.6%)	511 (38.2%)
Yes	115 (12.2%)	281 (29.8%)	546 (58%)
Europe:			
No	544 (18.8%)	1291 (44.7%)	1051 (36.4%)
Yes	233 (9%)	794 (30.8%)	1551 (60.2%)
North America:			
No	654 (20.2%)	1511 (46.7%)	1068 (33%)
Yes	254 (9.5%)	841 (31.6%)	1569 (58.9%)
United Kingdom:			
No	35 (9.6%)	200 (54.6%)	131 (35.8%)
Yes	18 (4.9%)	81 (22.1%)	267 (73%)

Table 12.1: Distribution of the number of high risk haplotypes DR3-DQ2 and DR4-DQ8 for affected and non-affected subjects in the T1DGC database. Global and by region data are shown.

affected. In the UK case, the frequency of carriers of two risk haplotypes between the affected doubles that frequency in the non-affected. In any case, it must me taken into account that the non-affected subjects in this sample are parents and siblings of the affected ones, so a high frequency of these haplotypes should be expected among the non-affected individuals.

When onset age of the subjects is considered, Table 12.2 shows the frequency distribution of the number of high-risk DRB-DQA-DQB haplotypes (in particular DR3-DQ2 and DQ4-DQ8) in subjects from the T1DGC database, globally and by regions. It can be seen that over 93% of the subjects with an onset of T1D before the age of 5 years have at least one risk haplotype, and more than 61% have two risk haplotypes, and so, the number of high risk haplotypes is related not only to the presence of T1D but also with an earlier age of onset.

Onset Age	Number of high risk haplotypes 0	1	2
Global Data:			
[0,5)	102 (7.1%)	451 (31.4%)	885 (61.5%)
[5,10)	169 (9.3%)	585 (32.2%)	1065 (58.5%)
[10,15)	180 (10.6%)	552 (32.6%)	963 (56.8%)
15 or more	164 (10.8%)	402 (26.5%)	950 (62.7%)
No T1D	1544 (19.7%)	3519 (45%)	2761 (35.3%)
Asia-Pacific:			
[0,5)	16 (7.8%)	77 (37.4%)	113 (54.9%)
[5,10)	29 (11.2%)	80 (31%)	149 (57.8%)
[10,15)	29 (11.4%)	76 (29.9%)	149 (58.7%)
15 or more	41 (19.4%)	48 (22.7%)	122 (57.8%)
No T1D	311 (23.2%)	517 (38.6%)	511 (38.2%)
Europe:			
[0,5)	33 (7.4%)	146 (32.7%)	267 (59.9%)
[5,10)	58 (8.8%)	229 (34.8%)	371 (56.4%)
[10,15)	63 (9.9%)	214 (33.8%)	357 (56.3%)
15 or more	77 (9.5%)	201 (24.9%)	529 (65.6%)
No T1D	544 (18.8%)	1291 (44.7%)	1051 (36.4%)
North America:			
[0,5)	49 (7.2%)	207 (30.4%)	424 (62.4%)
[5,10)	76 (9.7%)	253 (32.3%)	455 (58%)
[10,15)	82 (11.7%)	236 (33.7%)	382 (54.6%)
15 or more	44 (9.4%)	142 (30.3%)	282 (60.3%)
No T1D	654 (20.2%)	1511 (46.7%)	1068 (33%)
United Kingdom:			
[0,5)	4 (3.8%)	21 (19.8%)	81 (76.4%)
[5,10)	6 (5%)	23 (19.3%)	90 (75.6%)
[10,15)	6 (5.6%)	26 (24.3%)	75 (70.1%)
15 or more	2 (6.7%)	11 (36.7%)	17 (56.7%)
No T1D	35 (9.6%)	200 (54.6%)	131 (35.8%)

Table 12.2: Distribution of the number of DR3-DQ2 and DR4-DQ8 high risk haplotypes depending on the onset age in the T1DGC database. Global and by region data are shown.

12.1.2. Analysis of maternal factors associated with T1D early and childhood onset

To assess if there is any "maternal effect" on the age of onset beyond what can be explained by the presence of the risk haplotypes we consider the following variables:

- Mother T1D status (affected/unaffected).
- Age of the mother at the time of birth of the child
- For T1D affected mothers:
  - Age of onset of the mother
  - Onset of maternal diabetes before or after the birth of the affected child.
  - Number of years since the diagnosis of maternal T1D until the time of birth of the affected child.

In order to avoid interference by family size (i.e. bias in favour of factors present in larger families), always the two first affected siblings per family (the first 2 diagnosed, present in 2849 families) were included in the analysis. Gender, antibody positivity, number of *Associated Autoimmune Disease* (AAID), number of risk HLA haplotypes, and INS and CTLA4 genotypes were included in the model as independent variables and analysed in all the families.

12.1.2.1. Analysis of maternal factors considering all mothers

We consider the linear model:

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \varepsilon$$

where:

- The dependent variable Y is the age of onset of the subject (the first and second affected children in the family).
- The independent variables X_i are:
  - *birthAgeMother:* the age of the mother at the time of the child's birth.
  - *nRiskHaps:* number of risk DR3-DQ2 and DR4-DQ8 haplotypes carried by the subject.

- *r_gad65* and *r_ia2:* GAD and IA2 antibody positivity.
- *gender:* male/female.
- *Ins_hph1:* insulin hph1 genotype carried by the subject. Reference genotype is AA and effects of TA and TT are computed.
- CTLA4: ctla4 genotype. Reference genotype is AA and effect of AG and GG are computed.
- *AIDn:* number of autoimmune diseases.
- *T1DM:* Indicator variable of the mother having T1D.
- $T1DF$: Indicator variable of the father having T1D.

The estimation of the model is shown in Table 12.3. It can be seen that the variables *ins_hph1, ctla4* and *AIDn* are not significant. Refitting the model without these variables produces the result shown in Table 12.4. The difference between both models (difference in residual sum of squares) is not significant (p=0.2304).

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.0789	0.7601	31.68	0.0000
birthAgeMother	-0.2265	0.0195	-11.60	0.0000
nRiskHaps	-0.9039	0.1491	-6.06	0.0000
r_gad65	-3.3821	0.1992	-16.98	0.0000
r_ia2	-0.6928	0.1991	-3.48	0.0005
gender.female	-0.7993	0.1981	-4.03	0.0001
ins_hph1.TA	0.0840	0.2332	0.36	0.7186
ins_hph1.TT	0.9910	0.5824	1.70	0.0889
ctla4.AG	-0.3306	0.2194	-1.51	0.1319
ctla4.GG	-0.1993	0.2828	-0.70	0.4811
AIDn	0.3251	0.2557	1.27	0.2038
T1DM.Yes	-2.1153	0.6293	-3.36	0.0008
T1DF.Yes	-0.7408	0.3918	-1.89	0.0587

Table 12.3: Estimation of the linear model for the onset age of the son/daughter.

As we can see, after adjusting for the number of Risk Haplotypes, the positivity to GAD and IA2 antibody and the gender of the subject, maternal effects (age of the mother at birth and presence of T1D in the mother) are still identified. Even a slight effect of the father having T1D is also noticeable. Indeed, age of onset is reduced, as expected,

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	23.9721	0.7379	32.49	0.0000
birthAgeMother	-0.2272	0.0195	-11.66	0.0000
nRiskHaps	-0.9025	0.1488	-6.06	0.0000
r_gad65	-3.3876	0.1991	-17.02	0.0000
r_ia2	-0.6813	0.1987	-3.43	0.0006
gender.female	-0.7632	0.1961	-3.89	0.0001
T1DM.Yes	-2.1128	0.6292	-3.36	0.0008
T1DF.Yes	-0.7393	0.3917	-1.89	0.0592

Table 12.4: Estimation of the linear model for the onset age of the son/daughter excluding non-significant predictive variables.

with the increase in the number of high risk DR3-DQ2 and DR4-DQ8 haplotypes. Lower ages of onset are also associated with GAD and IA2 positivity, and girls tend to debut before boys. Taking these factors into account, the older the mother at childbirth the more likely is the child's disease onset to occur earlier. When the mother (and maybe the father) has T1D, the age of onset of the child is also expected to be lower, which means that there are probably other genetic factors involved in the age of onset.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.0912	0.7290	33.05	0.0000
birthAgeMother	-0.2262	0.0192	-11.77	0.0000
nRiskHaps	-1.0420	0.1523	-6.84	0.0000
r_gad65	-3.3601	0.1968	-17.07	0.0000
r_ia2	-0.7213	0.1968	-3.67	0.0003
gender.female	-0.7982	0.1935	-4.12	0.0000
T1DM.Yes	-2.0788	0.6267	-3.32	0.0009
T1DF.Yes	-0.9063	0.3854	-2.35	0.0187
HLA.ACwB.A1-B8	0.6052	0.2392	2.53	0.0114
HLA.ACwB.A24-B39	-3.4473	0.9700	-3.55	0.0004

Table 12.5: Estimation of the linear model for the onset age of the son/daughter including some HLA A-Cw-B haplotypes.

The allehap package allows for the exploration of the effect of other possible haplotypes in the subject's age of onset. For example, when considering HLA class I loci A-Cw-B haplotypes, it has been found [185]

that haplotype A1-B8 (HLA A*0101-Cw*0701-B*0801) may be associated with DR3-DQ2 and alter the associated risk.

We have used alleHap to explore the haplotypes in this region and find that A1-B8 is a relatively frequent haplotype, present in 1104 subjects in the database. We have also identified the HLA A*2402-Cw*0702-B*3906 (A24-B39) haplotype as associated to lower age of onset. When these haplotypes are included in the previous model, we get the estimation shown in Table 12.5, which show a significant effect of both haplotypes.

12.1.2.2. Analysis of maternal factors considering only mothers with T1D

When only mothers with T1D are considered, the effect of the mother onset age or the evolution time of the mother at the moment of childbirth can be taken into account. As the number of mothers with T1D in the database is low (n=67) we can not expect a great resolution from the model. Indeed, if we consider the same variables as before, we arrive at the results in Table 12.6, where the only significant variable is the number of risk haplotypes.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	18.8559	4.2940	4.39	0.0000
birthAgeMother	-0.2264	0.1254	-1.81	0.0741
nRiskHaps	-3.8879	0.8426	-4.61	0.0000
r_gad65	0.9741	1.2350	0.79	0.4321
r_ia2	-0.3782	1.1688	-0.32	0.7469
sexfemale	0.1404	1.1648	0.12	0.9043
inshph1TA	2.0843	1.3626	1.53	0.1293
inshph1TT	1.2318	2.9629	0.42	0.6785
ctla4AG	-1.0317	1.3931	-0.74	0.4607
ctla4GG	-0.0495	1.7396	-0.03	0.9774
numEnfAuto	0.8142	1.6676	0.49	0.6265
T1DFYes	-1.9170	1.8754	-1.02	0.3092

Table 12.6: Estimation of the linear model for the onset age of the son/daughter using only data from families in which mother have T1D.

If we refit the model leaving only the number of risk haplotypes and the age of the mother at the childbirth (see Table 12.7) we see that this variable is still significant.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	19.7262	3.2209	6.12	0.0000
nRiskHaps	-3.8667	0.7033	-5.50	0.0000
birthAgeMother	-0.2342	0.1103	-2.12	0.0358

Table 12.7: Estimation of the linear model for the onset age of the son/daughter using only data from families in which mother is T1D, considering only the number of Risk Haplotypes in the subject and the age of the mother at childbirth.

Introducing now the variables *OnsetM* (which specifies the age of onset of the mother) and *motherEvolTime* (time of evolution of T1D in the mother before the birthchild) we obtain the results in Table 12.8 where a possible effect of collinearity between the variables is advisable, making the effect of all variables not significant.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	14.7636	3.8348	3.85	0.0002
nRiskHaps	-3.7556	0.7067	-5.31	0.0000
birthAgeMother	-0.3558	0.2180	-1.63	0.1054
onsetM	0.3191	0.2319	1.38	0.1715
motherEvolTime	0.2285	0.2580	0.89	0.3776

Table 12.8: Estimation of the linear model for the onset age of the son/daughter using only data from families in which mother is T1D, considering the number of Risk Haplotypes in the subject, the age of the mother at childbirth, the onset age of the mother, and the evolution time of T1D (in years).

Due to this collinearity, it only makes sense to fit a model excluding either maternal age of onset or years since diagnosis. The best model is that obtained when only the mother's time from her T1D onset until the childbirth moment is considered (if the onset of the mother is produced *after* the childbirth, the time since diagnosis is considered 0).

Results are shown in Table 12.9. As can be seen, the evolution time of the mother has a negative effect on the age of onset of her offspring: the longer the time since diagnosis, the more likely is the child to have an earlier onset. In any case, this result must be taken cautiously because the age of mother at birth and time since diagnosis of T1D in

the mother are confounded variables (the older the mother, the more years of evolution of T1D). The effect of both variables is difficult to separate, since for any fixed age at birth there are few data for testing the effect of time since diagnosis.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	17.2737	3.3856	5.10	0.0000
nRiskHaps	-3.6260	0.7030	-5.16	0.0000
birthAgeMother	-0.1088	0.1242	-0.88	0.3825
motherEvolTime	-0.1179	0.0564	-2.09	0.0386

Table 12.9: Estimation of the linear model for the onset age of the son/-daughter using only data from families in which mother have T1D, considering the number of Risk Haplotypes in the subject and the evolution time of T1D (in years) in the mother.

12.2. Comparison of the distribution of risk haplotypes between the Canary Islands and the rest of Spain

It is well known that in the Canary Islands there is a high prevalence of T1D, greater than in the rest of Spain. We can use the haplotypes identified by alleHap to compare the distribution of risk haplotypes between affected subjects in both territories. The sample comprises a total of 597 genotyped subjects from 149 families. In many families, the father (60) or the mother (38) was missing. In this dataset, 42 families come from the Canary Islands. 226 of the subjects had T1D in the peninsular Spain and 86 in the islands. Table 12.10 show the distribution of risk (DR3-DQ8 and DR4-DQ2) and protection (DR2, DR6, DR7 and DR11) haplotypes in the markers DRB,-DQA-DQB.

We can see that there are not great differences between the two territories; the bigger difference is in the DR3-DQ2/DR3-DQ2 diplotype. In any case, to test the significance of the observed differences, a standard chi-square test cannot be applied as data are not independent (subjects in the same family share haplotypes). So we used a bootstrap procedure consisting of a simulation of 100000 times the random selection of 42 families from the total of 149 families of Spain. For every set of 42 randomly selected families, the χ^2 statistic is computed and saved.

Haplotypes	Canary Islands	Peninsular Spain
DR2-6-7-11/DR2-6-7-11	16 (9.1%)	11 (3.1%)
DR2-6-7-11/DR3-DQ2	25 (14.2%)	51 (14.2%)
DR2-6-7-11/DR4-DQ8	26 (14.8%)	40 (11.1%)
DR2-6-7-11/other	16 (9.1%)	20 (5.6%)
DR3-DQ2/DR3-DQ2	8 (4.5%)	36 (10%)
DR3-DQ2/DR4-DQ8	42 (23.9%)	85 (23.7%)
DR3-DQ2/other	11 (6.2%)	39 (10.9%)
DR4-DQ8/DR4-DQ8	12 (6.8%)	18 (5%)
DR4-DQ8/other	16 (9.1%)	38 (10.6%)
other/other	4 (2.3%)	21 (5.8%)

Table 12.10: Frequencies of the DRB-DQA-DQB haplotypes in the Canary Islands and Peninsular Spain

The 100000 values obtained this way give us the bootstrap distribution of this statistic, from which we can compute the p-value of the test for the sample of 42 Canarian families. Specifically, for the comparison of the distribution of DRB-DQB-DQA haplotypes between the Canary Islands and peninsular Spain, the resulting value of the Chi-squared test was 23.124. The bootstrap p-value is 0.097, and thus the differences are not significant. In any case, the small sample size (only 42 families in the Canary Islands), makes the results inconclusive.

12.3. Comparison of the distribution of risk haplotypes between Spain and the rest of Europe

In the same way, we can compare the distribution of risk haplotypes between Spain and the rest of Europe.

For European data there are 4091 subjects fully genotyped, distributed in 1137 families, with 2269 subjects affected with T1D. In these families there are 439 missing fathers and 259 missing mothers. After applying alleHap the distribution of identified haplotypes is shown in Table 12.11.

Again, no great differences can be observed. Proceeding the same way as before, the chi-square statistic for this table gives a value of 21.462, for which the bootstrap p-value is 0.1729, so there are not significant differences in the distribution of HLA risk haplotypes between Spain and the rest of Europe.

Haplotypes	European	Spanish
DR2-6-7-11/DR2-6-7-11	188 (4.9%)	25 (5.1%)
DR2-6-7-11/DR3-DQ2	452 (11.9%)	67 (13.6%)
DR2-6-7-11/DR4-DQ8	641 (16.9%)	62 (12.6%)
DR2-6-7-11/other	337 (8.9%)	35 (7.1%)
DR3-DQ2/DR3-DQ2	193 (5.1%)	37 (7.5%)
DR3-DQ2/DR4-DQ8	758 (19.9%)	119 (24.1%)
DR3-DQ2/other	298 (7.8%)	48 (9.7%)
DR4-DQ8/DR4-DQ8	273 (7.2%)	30 (6.1%)
DR4-DQ8/other	476 (12.5%)	51 (10.3%)
other/other	186 (4.9%)	19 (3.9%)

Table 12.11: Frequencies of the DRB-DQA-DQB haplotypes in Spain vs. rest of European countries

PART

V

Conclusion

Chapter 13

Main conclusions

As main conclusions of this thesis we cite the following:

I. A complete revision of the state of the art of methods in Statistical Genetics and Bioinformatics has been made to evaluate the association between genome and disease.

II. Elaboration of a Genome Wide Association Tutorial that includes Quality Control, Phasing, Alignment, Imputation, and Association Analysis in population data. This tutorial contains detailed instructions for the accomplishment of this tasks.

III. Genetic variants associated with advanced diabetic nephropathy in a T2D population from the Gran Canaria island have been identified. The more significant SNPs found were in the positions 9044856, 20307083, 49552399 of chromosomes 8, 10 and 11, labelled as rs4841106, rs2358658, and rs35649357, respectively. From previous SNPs, the second one (in chromosome 8) is the only one that belongs to an identified gene, the PLXDC2 or *Plexin Domain Containing 2*.

IV. Development of an R package: alleHap. This package is capable of imputing alleles and identifying haplotypes in familial databases by crossing non-recombinant genetic information of parents and offspring. The package has been uploaded to CRAN website where is publicly and freely available.

V. Performance of alleHap package has been evaluated:

   a) Regarding the processing time, computing time grows linearly with the number of families and the number of mark-

ers. Parallelization or virtualization techniques using network resources as *Hadoop* could be useful when any of this numbers if very large.

b) Concerning the imputation rate (considering a typical situation of 4 alleles per marker and 5 to 50 markers):

1) For an extreme situation of parents with completely missing genotypes, imputation rates range from approximately 5-6% (when only one child is available and has a high percentage of missing values), to almost a 55% (when three children are available, even with a missingness rate of 25% in children).
2) For an extreme situation of a child with completely missing genotypes, imputation rate ranges from 5-6% (when parents have 75% of missing alleles) to almost 60% (when parents have no missing values).
3) For intermediate situations with parents and children having missing alleles, imputation rates up to 98% can be achieved when there are at least three children available and the missingness rate is not too low. For a missingness rate of 50% in parents and children, the rate of imputation ranges from 21% (when there is only one child) to 55% (when there are three or more children).

c) With respect to the haplotyping rate (considering again a typical situation of 4 alleles per marker and 5 to 50 markers).

1) When there are no missing alleles in parents nor children, the complete haplotyping rates range from 80% (when there is only one child available) to a 100% (in cases when there are three or more children available).
2) For an extreme situation with children having 75% of missing alleles (if all alleles in children are missing, identification of haplotypes is not possible), the complete haplotyping rates range from 2% (when parents have a 50% of missing alleles) to a 35% (when parents have no missing alleles)
3) For an extreme situation with parents having 75% of missing alleles, the complete haplotyping rates range from 2-30% (when there is only one child with a 0 to 50% of missing alleles) to 6-85% (when there are, at

least, three children with missing alleles ranging from 0 to 50%).

4) For intermediate situations with parents and children having 25% of missing alleles, the complete haplotyping rates range from 22% (when there is only one child available) to 80% (in cases of three or more children available).

VI. A manual of the alleHap package (*Allele Imputation and Haplotype Reconstruction from Pedigree Databases*) has also been made available through CRAN website.

VII. Identification of haplotypes in HLA markers included in the international T1DGC pedigree database has been accomplished. Results confirm that over 90% of affected subjects carry at least one of the DR3-DQ2 and DR4-DQ8 risk haplotypes. This proportion is slightly higher in UK sample (95%).

VIII. Results also confirm that globally 61.5% of the very early onset cases (debut of T1D before the age of 5 years) carry two risk haplotypes. This proportion is again higher in UK data (76.4%).

IX. Analysis of maternal factors associated with T1D early and childhood onset show that after adjusting by the presence of high risk haplotypes:

a) The age of the mother at childbirth associates negatively with the children's age of onset (the older the mother, the earlier the age of onset).

b) The children's age of onset is lower in mean for T1D mothers.

c) For affected mothers, the time since diagnosis of T1D associates negatively with the children's age of onset (although we must be cautious with this result due to possible confounders).

X. Comparing the frequency of risk haplotypes for the Canary Islands sample versus the rest of Spain, no significant differences have been detected. The same occurs when comparing the Spanish sample against the rest of Europe. In any case, results are not conclusive due to the low number of Canarian families in the sample.

PART

VI

Appendix

Appendix A

GWA Tutorial

This tutorial pretends to facilitate and clarify the documentation for handling most used programs for quality control, pre-phasing, imputation and statistical analysis of GWA data. Thus, this brief tutorial intends to be a practical resource that outlines the basic processes and provides specific source codes for those software tools that we used for the development of Part III of this thesis.

For above purpose, we implemented some computational codes for the analysis of Genome-wide variation, as well as for post-analytic representation. Code scripts were developed using Unix's `Bash` programming and R scripts. Graphics also were generated R scripts and using `qqman` package [186].

Other programs used and which users will need to perform the analysis are: PLINK2 [187], SHAPEIT2 [165], IMPUTE2 [121], GTOOL [188] and SNPTEST2 [154].

Our raw files (called "`nefropatias.bed`", "`nefropatias.bim`" and "`nefropatias.fam`") are in BED/BIM/FAM format (see format's description in Section 6.1.1). These files comprised information of our study subjects as well as their genetic data (see study description in Section 6.1.2). Obviously, for a personal use of this tutorial users have to rename the files "`nefropatias`" by the corresponding one (e.g. "`my.file`").

A.1. Quality Control

- *File type:* Bash script
- *File name:* `Quality_Control.R`
- *Script's purpose:* Quality control of the genomic data.

```
> #!/bin/bash

> ################################################################
> ######################### Pre-QC steps #########################
> ################################################################

> ## Pruning of individuals with missing phenotypes
> ./plink2 --bfile nefropatias --prune --make-bed --out nefropatia

> ## Exclusion of SNPs in unknown chromosomes
> ./plink2 --bfile nefropatia --not-chr 0 --make-bed --out
     nefro_not_chr0

> ################################################################
> ###################### Variant-QC steps ########################
> ################################################################

> ## Filtering of SNPs with 0.1 as limit of missingness
> ./plink2 --bfile nefro_not_chr0 --geno 0.1 --make-bed --out
     nefro_geno

> ## Filtering of SNPs with 0.0045 as limit of MAF
> ./plink2 --bfile nefro_geno --maf 0.0045 --make-bed --out nefro_maf

> ## Filtering of SNPs with 0.0001 as limit of HWE
> ./plink2 --bfile nefro_maf --hwe 0.0001 --make-bed --out nefro_hwe

> ## All Variant-QC steps together
> ./plink2 --bfile nefro_not_chr0 --maf 0.0045 --geno 0.1 --hwe 0.0001
     --make-bed --out nefro_VQC

> ################################################################
> ######################### Sample-QC steps ######################
> ################################################################

> ## Filtering samples by missingness rate
> ./plink2 --bfile nefro_VQC --mind 0.09 --make-bed --out nefro_mind

> ## Checking gender mismatches in samples
  ./plink2 --bfile nefro_mind --check-sex --out nefro_sexch

> ## Checking sample relatedness (before VQC and SQC)
> ./plink2 --bfile nefro_not_chr0 --genome --out nefro_before_VQC
```

```
> ./plink2 --bfile nefro_VQC --genome --out nefro_before_SQC

> ## Filtering related samples
> ./plink2 --bfile nefro_mind --remove indivs_to_remove.txt --make-bed
    --out nefro_SQC

> ## Checking sample relatedness (after SQC)
> ./plink2 --bfile nefro_SQC --genome --out nefro_after_SQC

> ## Checking sample heterozygosity rates (before and after SQC)
> ./plink2 --bfile nefro_VQC --het --out nefro_before_SQC
> ./plink2 --bfile nefro_SQC --het --out nefro_after_SQC
```

- *File type:* R script
- *File name:* `Quality_Control.R`
- *Script's purpose:* Representation of the QC data.

```
> setwd("~/my.path/my.folder")

> ##################################################################
> ###### Threshold selection according to missingness rates #####
> ##################################################################

> ## Individual missingness (after Variant QC)
> IMISS <- read.table("nefro_VQC.imiss", header=T, as.is=T)

> ## Individual call rate distribution
> png("VQC_Individual_call_rate_distribution.png",width=18,height=18,
    units="cm",res= 600)
> plot( (1:dim(IMISS)[1])/(dim(IMISS)[1]-1), sort(1-IMISS$F_MISS),
     main="Individual call rate cumulative distribution",
     pch=10, col="orange2", ylim=c(0.7,1), xlab="Quantile",
     ylab="Call Rate")
> grid(); dev.off()

> ## Locus missingness (before Variant QC)
> LMISS <- read.table("nefro_not_chr0.lmiss", header=T, as.is=T)

> ## SNP coverage distribution (before Variant QC)
> png("QC_SNP_coverage_distribution.png",width=18,height=18,units="cm",
    res=600)
plot( (1:dim(LMISS)[1])/(dim(LMISS)[1]-1), sort(1-LMISS$F_MISS),
     main="SNP coverage cumulative distribution",
     pch=10, col="blue4", ylim=c(0,1), xlab="Quantile", ylab="Call
          Rate")
> grid(); dev.off()

> ## Locus missingness (after Variant QC)
> LMISS <- read.table("nefro_VQC.lmiss", header=T, as.is=T)
```

```
> ## SNP coverage distribution (after Variant QC)
> png("VQC_SNP_coverage_distribution.png",width=18,height=18,units="cm"
    ,res=600)
plot( (1:dim(LMISS)[1])/(dim(LMISS)[1]-1), sort(1-LMISS$F_MISS),
      main="SNP coverage cumulative distribution",
      pch=10, col="blue4", ylim=c(0,1), xlab="Quantile", ylab="Call
          Rate")
> grid(); dev.off()

> ## Q-Q plot of HWE log-p-values of the control samples
> HWE <- read.table("nefro_hardy_ctrl.hwe", header=T, as.is=T)
> P <- HWE$P
> n <- length(P)
> PP <- -2*log(P)
> sp <- order(PP)
> PP <- PP[sp]
> qchi <- qchisq((1:n)/(n+1),2)
> m <- floor(seq(1,length(PP),length=10000))
> qc <- qchi[m]
> pc <- PP[m]
> png("QQ_plot_of_HW_control_p-values.png",width=18,height=18,units="cm
    ",res=600)
plot(qc,pc, main="QQ plot of log(P-values) in HWE",
      xlab="Expected quantile", ylab="Observed quantile")
> lines( c(0,50), c(0,50) )
> grid(); dev.off()

> ###################################################################
> ################# Relatedness Networks Analysis #################
> ###################################################################

> library("igraph")
> GEN <- read.table("nefro_related.genome", header=T, as.is=T)
> GEN$FID1 <- as.character(GEN$FID1)
> GEN$FID2 <- as.character(GEN$FID2)
> SampleID <- > levels(as.factor(c(GEN$FID1,GEN$FID2)))
> n <- length(SampleID)
> GENr <- GEN[GEN$PI_HAT>0.1875,]    #Important pairs only
> ibd <- GENr$PI_HAT
> g1 <- graph.edgelist(cbind(GENr$FID1,GENr$FID2), directed=F)
> edgewidth <- 1+(ibd>0.1875)+(ibd>0.375)+(ibd>0.75)
> png("Sample_Relateness.png",
     width=12,height=12,units="cm",res=300)
> plot(g1, layout=layout.fruchterman.reingold, vertex.label=V(g1)$name,
    vertex.label.cex=1, vertex.shape="none", edge.label=round(ibd,2),
    edge.label.cex=0.8, edge.label.color="black", edge.width=2*
        edgewidth, edge.color="red")
> dev.off()
```

```
> ##################################################################
> ########## Comparison between IBD estimates Histograms ########
> ##################################################################
> png("IBD_Histograms.png",width=22,height=10,units="cm",res=300)
> oldpar <- par(mfrow=c(1,3))
> GENbVQC <- read.table("nefro_before_VQC.genome", header=T, as.is=T)
> IBD_before_VQC <- GENbVQC$PI_HAT
> hist(IBD_before_VQC,50)
> GENbSQC <- read.table("nefro_before_SQC.genome", header=T, as.is=T)
> IBD_before_SQC <- GENbSQC$PI_HAT
> hist(IBD_before_SQC,50)
> GENaSQC <- read.table("nefro_after_SQC.genome", header=T, as.is=T)
> IBD_after_SQC <- GENaSQC$PI_HAT
> hist(IBD_after_SQC,50)
> par(oldpar)
> dev.off()

> ##################################################################
> Histogram of Heterozygosity H, and an inversely related value F
> ##################################################################
> ## Before and after SQC
> HETbeforeSQC <- read.table("nefro_before_SQC.het", header=T, as.is=T)
> HETafterSQC <- read.table("nefro_after_SQC.het", header=T, as.is=T)
> for (item in c("before","after")){
>   if (item=="before") {
>     HET <- HETbeforeSQC
>     png("Heterozygosity_Histograms_before_SQC.png",
          width=22,height=12,units="cm",res=300)
>   } else if (item=="after") {
>     HET <- HETafterSQC
>     png("Heterozygosity_Histograms_after_SQC.png",
          width=22,height=12,units="cm",res=300)
>   }
>   H <- (HET$N.NM.-HET$O.HOM.)/HET$N.NM.
>   oldpar <- par(mfrow=c(1,2))
>   hist(H,50)
>   F <- HET$F
>   hist(F,50)
>   dev.off()
> }
> outLimits <- mean(F)+c(-3,3)*sd(F)
> outSubjects <- HET[HET$F<outLimits[1]|HET$F>outLimits[2],2]
```

A.2. Pre-processing

A.2.1. Split

- *File type:* R script
- *File name:* `Quality_Control.R`
- *Script's purpose:* Data partitioning into separate chromosomes.

```
> #!/bin/bash

> for chrom in {1..22}
> do
>     ./plink --nonfounders --allow-no-sex --noweb --bfile
      nefro_cleaned_maf00091 --chr ${chrom} --make-bed --out chr${chrom}

> done

> # in chromosome X
> ./plink --nonfounders --allow-no-sex --noweb --bfile
      nefro_cleaned_maf00091 --chr X --make-bed --out chrX
```

A.2.2. Alignment

- *File type:* Bash script
- *File name:* `alignments_1000G.sh`
- *Script's purpose:* Checking and alignment of each chromosome.

```
> #!/bin/bash

> # Directories
> ROOT_DIR=./
> PANEL_DIR=${ROOT_DIR}1000G/
> DATA_DIR=${PANEL_DIR}data_files_1000G/
> RESULTS_DIR=${PANEL_DIR}results_shapeit_1000G/

> # executable
> SHAPEIT_EXEC=${ROOT_DIR}shapeit

> # Chromosome to check
> CHR=$1
> # (JxT= Number of CPU cores) # J = Jobs; T = Threads
> THREAD=1

> # Reference data files
> GENMAP_FILE=${DATA_DIR}genetic_map_chr${CHR}_combined_b37.txt

> # GWAS data files in PLINK BED format
```

```
> GWASDATA_BED=${DATA_DIR}chr${CHR}.bed
> GWASDATA_BIM=${DATA_DIR}chr${CHR}.bim
> GWASDATA_FAM=${DATA_DIR}chr${CHR}.fam

> # 1000GP reference files
> GWASDATA_HAP=${DATA_DIR}1000GP_Phase3_chr${CHR}.hap.gz
> GWASDATA_LEG=${DATA_DIR}1000GP_Phase3_chr${CHR}.legend.gz
> GWASDATA_SAM=${DATA_DIR}1000GP_Phase3.sample

> # Includes "EUR" group
> GROUP_LIST=${DATA_DIR}group.list.txt

> # Main output file
> OUTPUT_LOG=${RESULTS_DIR}alignments_chr${CHR}.log

> ## Shapeit execution
> $SHAPEIT_EXEC \
  -check \
  --input-map $GENMAP_FILE \
  --input-bed $GWASDATA_BED $GWASDATA_BIM $GWASDATA_FAM \
  --input-ref $GWASDATA_HAP $GWASDATA_LEG $GWASDATA_SAM \
  --thread $THREAD \
  --include-grp $GROUP_LIST \
  --output-log $OUTPUT_LOG
```

A.2.3. Pre-phasing

- *File type:* Bash script
- *File name:* `shapeit_phasing_job_1000G.sh`
- *Script's purpose:* Pre-phasing of each chromosome.

```
> #!/bin/bash

> # directories
> ROOT_DIR=./
> DATA_DIR=${ROOT_DIR}data_files/
> RESULTS_DIR=${ROOT_DIR}results/

> # Executable
> SHAPEIT_EXEC=${ROOT_DIR}shapeit

> # Chromosome
> CHR=$1

> # GWAS data files in PLINK BED format
> GWASDATA_BED=${DATA_DIR}chr${CHR}.bed
> GWASDATA_BIM=${DATA_DIR}chr${CHR}.bim
> GWASDATA_FAM=${DATA_DIR}chr${CHR}.fam
```

```
> # reference data files
> GENMAP_FILE=${DATA_DIR}genetic_map_chr${CHR}_combined_b37.txt

> # (JxT= Number of CPU cores)
> # J = Jobs; T = Threads
> THREAD=1

> # Windows Size
> EFFECTIVE_SIZE=11418

> # Excludes non-aligned snps
> ALIGN_FILE=${RESULTS_DIR}alignments_chr${CHR}.snp.strand.exclude

> # Includes "EUR" group
> GROUP_LIST=${DATA_DIR}group.list.txt

> # Main output file
> OUTPUT_HAPS=${RESULTS_DIR}chr${CHR}.phased.haps
> OUTPUT_SAMPLE=${RESULTS_DIR}chr${CHR}.phased.sample
> OUTPUT_VCF=${RESULTS_DIR}chr${CHR}.phased.vcf
> OUTPUT_LOG=${RESULTS_DIR}chr${CHR}.phased.log

> # Phase GWAS genotypes
> $SHAPEIT_EXEC \
  --input-bed $GWASDATA_BED $GWASDATA_BIM $GWASDATA_FAM \
  --input-map $GENMAP_FILE \
  --thread $THREAD \
  --effective-size $EFFECTIVE_SIZE \
  --exclude-snp $ALIGN_FILE \
  --include-grp $GROUP_LIST \
  --output-max $OUTPUT_HAPS $OUTPUT_SAMPLE \
  --output-log $OUTPUT_LOG
```

A.3. Imputation

- *File type:* R script
- *File name:* `impute2_jobs_1000G.R`
- *Script's purpose:* Split chromosome's data into chunks.

```
> #!/usr/bin/Rscript --vanilla

> Directories
> data.dir <- paste("data_files/", sep="")
> res.dir <- paste("results/", sep="")

# Default settings; Can change when loading other scripts!!
> chr <- 21
```

```
> chunk.size <- 5e+06  # chunk size

> # Read in file with chunk boundary definitions
> genetic.map.file <- paste(data.dir,"genetic_map_chr",chr,"
    _combined_b37.txt", sep="")
> gmf.table <- read.table(genetic.map.file, head=T, as.is=T)
> chunks <- seq(gmf.table[1,1],gmf.table[nrow(gmf.table),1],chunk.size)
> chunks <- unique(c(chunks,gmf.table[nrow(gmf.table),1]+1))

> # Submit a job to the cluster for each analysis chunk on this
    chromosome
> gen.to.merge <- NULL
> sam.to.merge <- NULL
> for (i in 1:(length(chunks)-1)) {
>   syscall.subset <- paste("./shapeit_subset.sh ",chr," ",chunks[i],"
    ",chunks[i+1]-1,sep="")
>   system(syscall.subset)
>   syscall.impute2 <- paste("./imputation_job_best_guess_haps.sh ",
    chr," ",chunks[i]," ",chunks[i+1]-1,sep="")
>   system(syscall.impute2)
>   gen.to.merge <- paste(gen.to.merge,paste(res.dir,"chr",chr,".pos",
    chunks[i],"-",chunks[i+1]-1,".imputed.gen",sep=""))
>   sam.to.merge <- paste(sam.to.merge,paste(res.dir,"chr",chr,".pos",
    chunks[i],"-",chunks[i+1]-1,".phased.sample",sep=""))
> }
> syscall.merge <- paste("./gtool -M --g",gen.to.merge," --s",
                         sam.to.merge," --og ",res.dir,"chr",chr,
                         ".imputed.merged.gen"," --log ",res.dir,
                         "chr",chr,".imputed.merged.log",sep="")
> system(syscall.merge)
```

- *File type:* Bash script
- *File name:* `imputation_job_best_guess_haps_1000G.sh`
- *Script's purpose:* Imputation of each chromosome chunk.

```
#!/bin/bash

# Chromosome and chunks
> CHR=$1
> CHUNK_START=`printf "%.0f" $2`
> CHUNK_END=`printf "%.0f" $3`

> # Directories
> ROOT_DIR=./
> PANEL_DIR=${ROOT_DIR}1000G/
> DATA_DIR=${PANEL_DIR}data_files_1000G/
> RESULTS_DIR=${PANEL_DIR}results_impute_1000G/
> HAP_SAMP_DIR=${PANEL_DIR}sampled_haps/
```

```
> # Executable
> IMPUTE2_EXEC=${ROOT_DIR}impute2

> # Effective population size
> NE=11418

> # Filter by the European population
> FILTER='EUR==0'

> # Reference data files
> GENMAP_FILE=${DATA_DIR}genetic_map_chr${CHR}_combined_b37.txt
> HAPS_FILE=${DATA_DIR}1000GP_Phase3_chr${CHR}.hap.gz
> LEGEND_FILE=${DATA_DIR}1000GP_Phase3_chr${CHR}.legend.gz

> # Haplotypes from SHAPEIT phasing run
> PHASED_HAPS=${RESULTS_DIR}chr${CHR}.pos${CHUNK_START}-${CHUNK_END}.
    phased.haps

> # Main output files
> OUTPUT_FILE=${RESULTS_DIR}chr${CHR}.pos${CHUNK_START}-${CHUNK_END}.
    imputed.gen
> OUT_SUMMARY=${RESULTS_DIR}chr${CHR}.pos${CHUNK_START}-${CHUNK_END}.
    imputed.summary
> OUT_WARNINGS=${RESULTS_DIR}chr${CHR}.pos${CHUNK_START}-${CHUNK_END}.
    imputed.warnings

> ## Impute genotypes from best-guess GWAS haplotypes
> $IMPUTE2_EXEC \
  -filt_rules_l $FILTER \
  -m $GENMAP_FILE \
  -known_haps_g $PHASED_HAPS \
  -h $HAPS_FILE \
  -l $LEGEND_FILE \
  -Ne $NE \
  -int $CHUNK_START $CHUNK_END \
  -o $OUTPUT_FILE \
  -r $OUT_SUMMARY \
  -w $OUT_WARNINGS
```

A.4. GWA Analysis

- *File type:* Bash script
- *File name:* `snptest_jobs_impute_1000G.sh`
- *Script's purpose:* Statistical analysis of imputed data.

```
> #!/bin/bash

> # directories
> ROOT_DIR=./
> PANEL_DIR=${ROOT_DIR}1000G/
> IMPUT2_DIR=${PANEL_DIR}results_impute_1000G/
> RESULTS_DIR=${PANEL_DIR}results_snptest_1000G/

> # Executable
> SNPTEST_EXEC=${ROOT_DIR}snptest_v2.5

> # Chromosome
> CHR=$1

> # imputed data
> INPUT_GEN=${IMPUT2_DIR}chr${CHR}.imputed.merged.gen
> INPUT_SAMP=${IMPUT2_DIR}imputed.merged.sample

> # output data
> OUT_DATA=${RESULTS_DIR}new.chr${CHR}.imputed.merged.new.em.out

> # Method option: deals with genotype uncertainty
> METH_OPT=em

> # The five different frequentist models are coded as 1=Additive,
    2=Dominant, 3=Recessive, 4=General and 5=Heterozygote

> # Phenotype: if B is placed in the sample file,
              then a case-control test is carried out
> PHEN=plink_pheno

> ## GWAS ANALYSIS
> $SNPTEST_EXEC \
  -data $INPUT_GEN $INPUT_SAMP \
  -o $OUT_DATA \
  -method $METH_OPT \
  -frequentist 1 2 3 4 5 \
  -exclude_samples ${PANEL_DIR}excluded_indivs.list \
  -pheno $PHEN
```

A.4.1. Compilation of scripts

- *File type:* Bash script
- *File name:* `compilation_1000G.sh`
- *Script's purpose:* Compilation of previous scripts for all chromosomes.

```
> #!/bin/bash

> for chrom in {1..22}
> do
   ./alignments_1000G.sh ${chrom}
   ./shapeit_phasing_job_1000G.sh ${chrom}
   ./Rscript ~/impute2_jobs_1000G.R chr=${chrom}
   ./1000G/snptest_jobs_impute_1000G.sh ${chrom}
> done
```

A.5. Data Representation

- *File type:* R script
- *File name:* `plot_qqman_snptest_pvalues.R`
- *Script's purpose:* Results representation by Manhattan and QQ plots.

```
> #!/usr/bin/Rscript --vanilla

> ### R Libraries
> library(qqman)
> library(data.table)
> library(colorRamps)

> ### Directory
> setwd("~/my.path/CHR_imputed")

> ### Selection of plot colors
> n=8
> colors <- primary.colors(24, steps=n, no.white=TRUE)
> colors <- colors[2:23]

### Frequentist tests
seqTests <- c("add","dom","gen","het","rec")

> for (test in seqTests){

>  ### Results of test in all merged CHRs
>  mergedName <- paste("merged.",test,".",imputation,".values",sep="")
>  resultsTest <- fread(mergedName)
>  cat("Reading file",mergedName,"\n")
>  setnames(resultsTest,names(resultsTest),
            c("CHR","SNP","BP","INFO","P","INFO2"))

>  ### Manhattan Plots
>  png(paste("qqman_plots/Manhattan_snptest_",
      test,"_New.png",sep=""), width=2000, height=1000)
>  manhattan(resultsTest, main=paste("Manhattan Plot: ",test,
```

```
            " frequentist test",sep=""), cex.main=1.5, cex.axis=1.2,
                cex.lab=1.5, cex=0.6, ylim=c(1,7), col=colors)
>  dev.off()
>  print(paste("Manhattan plot of ",test," results
              has been successfully printed!",sep=""))

>  ### QQ Plots
>  png(paste("qqman_plots/QQ_snptest_",test,
            "_New.png",sep=""), width=750, height=750)
>  qq(resultsTest$P, main=paste("QQ Plot: ",test,"
     frequentist test",sep=""), xlim=c(0,8), ylim=c(0,8),
     pch=1, cex=1.2, col="blue4")
>  dev.off()
>  print(paste("QQ plot of ",test," results has been
              successfully printed!",sep=""))

}
```

Appendix B

alleHap Manual

This manual includes detailed and easily reproducible examples for each one of the different alleHap stages (previously explained in Chapter 10). Also, it described the input format that should be used with the package.

B.1. Input Format

alleHap only works with *Pedigree* (PED) files, although it can detect and adapt similar formats (with the same structure) to later load the data.

B.1.1. PED files

A PED file is a space or tab delimited file where the first six columns are mandatory and represent the following *Identifier* (ID)s: ***Family ID*** (for each family), ***Individual ID*** (for each family's member), ***Paternal ID*** (for the paternal ancestor), ***Maternal ID*** (for the maternal ancestor), ***Sex*** (genre of each individual: 1=male, 2=female, other=unknown), ***Phenotype*** (quantitative trait or affection status of each individual: -9=missing, 0=unaffected, 1=affected). The remaining columns represent the ***Genotype*** of each individual (in biallelic or coded format).

The IDs are alphanumeric: the combination of family and individual ID should uniquely identify a person. **PED files must have one and only one phenotype in the sixth column**. Genotypes (column 7 onwards) should also be white-space delimited and can be any character

(e.g. 1,2,3,4 or A,C,G,T or anything else) except 0 which is, by default, the missing genotype character. All markers should be biallelic and must have two alleles specified [130]. For example, a family composed of three individuals typed for N SNPs is represented in Table B.1.

Fam ID[a]	*Ind ID*	*Pat ID*	*Mat ID*	*Sex*	*Pheno*	*Mkr_1*		*Mkr_2*		*Mkr_3*		...	*Mkr_N*	
FAM001	1	0	0	1	0	A	A	G	G	A	C	...	C	G
FAM001	2	0	0	2	1	A	A	A	G	C	C	...	A	G
FAM001	3	0	0	1	0	A	A	G	A	A	C	...	C	A

Table B.1: Example of a Family in .ped file format

[a] No header row should be given.

B.1.2. Missing values

The missing values may be placed either in the first six colums or in genotype columns. In the genotype columns, when some values are missing either both alleles should be 0, -9, -99, NA or whatever manually introduced by users (by means of the `missingValue` parameter of alleLoader function). An example of this, using NAs as missing values, would be:

```
 famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
FAM001     1     0     0   1    0     1     2    NA    NA     1     2
FAM001     2     0     0   2    0     3     4     1     2     3     4
FAM001     3     1     2   1    0     1     3     1     2     1     3
FAM001     4     1     2   2    0    NA    NA     1     1     2     4
FAM001     5     1     2   1    0     1     4     1     1     2     4
```

B.2. Data Simulation

This part of the package simulates biallelic pedigree databases which can be performed taking into account many different factors such as number of families to generate, number of markers (allele pairs), number of different alleles per marker, type of alleles (numeric or character), number of different haplotypes in the population, probability of parent/offspring missing genotypes, proportion of missing genotypes per individual, probability of being affected by disease, and recombination rate.

B.2.1. alleSimulator Examples

Next examples show how alleSimulator works:

Example B.2.1 *Simulation of a family containing parental missing data.*

```
> simulatedFam1 <- alleSimulator(1,2,3,missParProb=0.3)

  ==========================================
  ===== alleHap package: version 0.9.6 ======
  ==========================================

  Data have been successfully loaded from:
  /home/nmr/Desktop/R-libraries/alleHap/vignettes

  ===== DATA COUNTING ======
  Number of families: 1
  Number of individuals: 4
  Number of founders: 2
  Number of children: 2
  Number of males: 2
  Number of females: 2
  Number of markers: 3
  ============================

  ======== DATA RANGES =========
  Family ID: FAM01
  Individual IDs: [1,...,4]
  Paternal IDs: [0,1]
  Maternal IDs: [0,2]
  Sex values: [1,2]
  Phenotype values: [1]
  ==============================

  ========= MISSING DATA =========
  Missing founders: 0
  Missing ID numbers: 0
  Missing paternal IDs: 0
  Missing maternal IDs: 0
  Missing sex: 0
  Missing phenotypes: 0
  Missing alleles: 4
  Markers with missing values: 2
  ================================

> # Alleles (genotypes) of the 1st simulated family
> simulatedFam1[[1]]

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
  FAM01     1     0     0   1    1     T     T     C     C  <NA>  <NA>
  FAM01     2     0     0   2    1  <NA>  <NA>     T     C     C     G
  FAM01     3     1     2   2    1     T     T     C     T     G     C
  FAM01     4     1     2   1    1     T     T     C     T     C     C
```

```
> # 1st simulated family haplotypes (without missing values)
> simulatedFam1[[2]]

  famID indID patID matID sex phen Paternal_Hap Maternal_Hap
  FAM01     1     0     0   1    1        T-C-G        T-C-C
  FAM01     2     0     0   2    1        T-T-C        T-C-G
  FAM01     3     1     2   2    1        T-C-G        T-T-C
  FAM01     4     1     2   1    1        T-C-C        T-T-C
```

Example B.2.2 *Simulation of a family containing offspring missing data..*

```
> simulatedFam2 <- alleSimulator(1,2,3,missOffProb=0.3)

  ==========================================
  ===== alleHap package: version 0.9.6 ======
  ==========================================

  Data have been successfully loaded from:
  /home/nmr/Desktop/R-libraries/alleHap/vignettes

  ===== DATA COUNTING ======
  Number of families: 1
  Number of individuals: 4
  Number of founders: 2
  Number of children: 2
  Number of males: 1
  Number of females: 3
  Number of markers: 3
  ============================

  ======== DATA RANGES =========
  Family ID: FAM01
  Individual IDs: [1,...,4]
  Paternal IDs: [0,1]
  Maternal IDs: [0,2]
  Sex values: [1,2]
  Phenotype values: [1,2]
  ==============================

  ========= MISSING DATA =========
  Missing founders: 0
  Missing ID numbers: 0
  Missing paternal IDs: 0
  Missing maternal IDs: 0
  Missing sex: 0
  Missing phenotypes: 0
  Missing alleles: 4
  Markers with missing values: 1
  ================================
```

```
> # Alleles (genotypes) of the 2nd simulated family
> simulatedFam2[[1]]

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
  FAM01     1     0     0   1    1     T     T     C     T     C     C
  FAM01     2     0     0   2    2     T     C     C     C     C     T
  FAM01     3     1     2   2    1     T     T     C     T  <NA>  <NA>
  FAM01     4     1     2   2    1     T     T     C     C  <NA>  <NA>

> # 2nd simulated family haplotypes (without missing values)
> simulatedFam2[[2]]

  famID indID patID matID sex phen Paternal_Hap Maternal_Hap
  FAM01     1     0     0   1    1        T-C-C        T-T-C
  FAM01     2     0     0   2    2        T-C-C        C-C-T
  FAM01     3     1     2   2    1        T-C-C        T-T-C
  FAM01     4     1     2   2    1        T-C-C        T-C-C
```

B.3. Workflow

The workflow of alleHap comprise mainly three stages: *Data Loading*, *Data Imputation* and *Data Phasing*. The next subsections will describe each of them.

B.3.1. Data Loading

The package can be used with either simulated or real data, and can handle or not genetic missing information. As it has mentioned in section B.1, ***.ped*** files are the default input format of alleHap, and although its loading process is quite straightforward, it is important to note that a file containing **a large number of markers could slow down the process**. Furthermore, in order to avoid the foregoing, it is highly recommended that users *split the data into non-recombinant chunks*, where each chunk should be later loaded by the alleLoader function.

B.3.1.1. alleLoader Examples

Next example depicts how alleLoader should be used:

Example B.3.1 *Loading of a dataset in .ped format with alphabetical alleles (A,C,G,T).*

```
> example1=file.path(find.package("alleHap"),"examples","example1.ped")
> # Loaded alleles of the example 1
> example1Alls <- alleLoader(example1)

  ============================================
  ===== alleHap package: version 0.9.6 ======
  ============================================

  Data have been successfully loaded from:
  /home/nmr/R/pc-linux-gnu-library/3.2/alleHap/examples/example1.ped

  ===== DATA COUNTING ======
  Number of families: 50
  Number of individuals: 227
  Number of founders: 100
  Number of children: 127
  Number of males: 118
  Number of females: 109
  Number of markers: 12
  ==========================

  ======== DATA RANGES =========
  Family IDs: [1,...,50]
  Individual IDs: [1,...,8]
  Paternal IDs: [0,1]
  Maternal IDs: [0,2]
  Sex values: [1,2]
  Phenotype values: [1,2]
  ==============================

  ========= MISSING DATA =========
  Missing founders: 0
  Missing ID numbers: 0
  Missing paternal IDs: 0
  Missing maternal IDs: 0
  Missing sex: 0
  Missing phenotypes: 0
  Missing alleles: 0
  Markers with missing values: 0
  ================================

> # Alleles of the first 10 subjects
> example1Alls[1:10,1:5]

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
      1     1     0     0   1    1     T     T     C     T     A     A
      1     2     0     0   2    1     A     T     C     G     C     C
      1     3     1     2   1    2     A     T     G     T     A     C
      1     4     1     2   2    1     A     T     C     G     A     C
      1     5     1     2   2    1     A     T     C     G     A     C
```

Example B.3.2 *Loading of a dataset in .ped format with numerical alleles.*

```
> example2=file.path(find.package("alleHap"),"examples","example2.ped")
> # Loaded alleles of the example 2
> example2Alls <- alleLoader(example2,dataSummary=FALSE)
> # Alleles of the first 7 subjects
> example2Alls[1:7,]

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
   1036     1     0     0   1    1   101  1601   101   102   501   502
   1036     2     0     0   2    1   301   401   301   501   201   301
   1036     3     1     2   1    2   301  1601   102   501   201   502
   1036     4     1     2   1    2   301  1601   102   501   201   502
   1239     1     0     0   1    1    NA    NA    NA    NA    NA    NA
   1239     2     0     0   2    1    NA    NA    NA    NA    NA    NA
   1239     3     1     2   2    2   301   401   301   501   201   302
```

B.3.2. Data Imputation

This part of the package imputes the previous simulated/loaded datasets by analyzing all possible combinations of a parent-offspring pedigree in which parental and offspring genotypes may be missing. As long as one child was genotyped, in certain cases, it is possible an unequivocal imputation of missing genotypes both in parents and children.

At this phase, each family genotype is imputed marker by marker, and alleImputer is the function implemented to do it.

B.3.2.1. alleImputer Examples

Next examples show how alleImputer works:

Example B.3.3 *Deterministic imputation for familial data containing parental missing values.*

```
> simulatedFam1 <- alleSimulator(1,2,3,missParProb=0.6)
> # Simulated alleles
> simulatedFam1[[1]]

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
  FAM01     1     0     0   1    1     A     G  <NA>  <NA>     A     G
  FAM01     2     0     0   2    1  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
  FAM01     3     1     2   1    1     G     G     G     G     G     G
  FAM01     4     1     2   2    2     A     A     T     T     A     A

> ## Genotype imputation of previously simulated data
```

```
> imputedFam1 <- alleImputer(simulatedFam1[[1]])

  ========================================
  ===== alleHap package: version 0.9.6 ======
  ========================================

  Data have been successfully loaded from:
  /home/nmr/Desktop/R-libraries/alleHap/vignettes

  ===== DATA COUNTING ======
  Number of families: 1
  Number of individuals: 4
  Number of founders: 2
  Number of children: 2
  Number of males: 2
  Number of females: 2
  Number of markers: 3
  ==========================

  ======== DATA RANGES =========
  Family ID: FAM01
  Individual IDs: [1,...,4]
  Paternal IDs: [0,1]
  Maternal IDs: [0,2]
  Sex values: [1,2]
  Phenotype values: [1,2]
  ==============================

  ========= MISSING DATA =========
  Missing founders: 0
  Missing ID numbers: 0
  Missing paternal IDs: 0
  Missing maternal IDs: 0
  Missing sex: 0
  Missing phenotypes: 0
  Missing alleles: 8
  Markers with missing values: 3
  ================================

  ===== IMPUTATION SUMMARY =====
  0 markers (0 alleles) have been
  turned into missing in 0 families
  due to familial inconsistencies.
  Alleles initially missing: 8
  Number of imputed alleles: 8
  Imputation rate: 1
  Imputation time: 0.01
  ==============================

> # Imputed alleles (markers)
```

```
> imputedFam1['imputedMkrs']

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
  FAM01     1     0     0   1    1     A     G     G     T     A     G
  FAM01     2     0     0   2    1     G     A     G     T     G     A
  FAM01     3     1     2   1    1     G     G     G     G     G     G
  FAM01     4     1     2   2    2     A     A     T     T     A     A
```

Example B.3.4 *Deterministic imputation for familial data containing offspring missing values.*

```
> simulatedFam2 <- alleSimulator(2,2,3,missOffProb=0.6)
> # Simulated alleles
> simulatedFam2[[1]]

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
  FAM01     1     0     0   1    1     C     A     A     G     C     C
  FAM01     2     0     0   2    1     C     A     A     G     C     C
  FAM01     3     1     2   2    2     A     A  <NA>  <NA>     C     C
  FAM01     4     1     2   2    1     C     C     A     A  <NA>  <NA>
  FAM02     1     0     0   1    1     A     C     A     G     C     T
  FAM02     2     0     0   2    1     A     C     A     G     C     T
  FAM02     3     1     2   2    1  <NA>  <NA>  <NA>  <NA>     C     T
  FAM02     4     1     2   1    1  <NA>  <NA>     A     A     C     C

> ## Genotype imputation of previously simulated data
> imputedFam2 <- alleImputer(simulatedFam2[[1]])

  ==========================================
  ===== alleHap package: version 0.9.6 ======
  ==========================================

  Data have been successfully loaded from:
  /home/nmr/Desktop/R-libraries/alleHap/vignettes

  ===== DATA COUNTING ======
  Number of families: 2
  Number of individuals: 8
  Number of founders: 4
  Number of children: 4
  Number of males: 3
  Number of females: 5
  Number of markers: 3
  ============================

  ======== DATA RANGES =========
  Family IDs: [FAM01,...,FAM02]
  Individual IDs: [1,...,4]
  Paternal IDs: [0,1]
```

```
  Maternal IDs: [0,2]
  Sex values: [1,2]
  Phenotype values: [1,2]
  ==============================

  ========= MISSING DATA =========
  Missing founders: 0
  Missing ID numbers: 0
  Missing paternal IDs: 0
  Missing maternal IDs: 0
  Missing sex: 0
  Missing phenotypes: 0
  Missing alleles: 4
  Markers with missing values: 1
  ================================

  ===== IMPUTATION SUMMARY =====
  0 markers (0 alleles) have been
  turned into missing in 0 families
  due to familial inconsistencies.
  Alleles initially missing: 4
  Number of imputed alleles: 0
  Imputation rate: 0
  Imputation time: 0.01
  ==============================

> # Imputed alleles (markers)
> imputedFam2['imputedMkrs']

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
  FAM01     1     0     0   1    1     C     A     A     G     C     C
  FAM01     2     0     0   2    1     C     A     A     G     C     C
  FAM01     3     1     2   2    2     A     A  <NA>  <NA>     C     C
  FAM01     4     1     2   2    1     C     C     A     A     C     C
  FAM02     1     0     0   1    1     A     C     A     G     C     T
  FAM02     2     0     0   2    1     A     C     A     G     C     T
  FAM02     3     1     2   2    1  <NA>  <NA>  <NA>  <NA>     C     T
  FAM02     4     1     2   1    1  <NA>  <NA>     A     A     C     C
```

B.3.3. Data Haplotyping

At this stage, the corresponding haplotypes of the biallelic pedigree databases are generated. To accomplish this, based on the user's knowledge of the intended genomic region to analyse, it is necessary to **slice the data into non-recombinant chunks** in order to to perform later the haplotype reconstruction to each of them. The alleHaplotyper function

firstly carries out the data imputation (using the previously explained function alleImputer) and secondly performs the haplotype generation.

The final output the alleHaplotyper is a list comprised by five elements: *imputedMkrs* (which contains the preliminary imputed marker's alleles), *IDS* (which includes the resulting IDentified/Sorted matrix), *reImputedAlls* (which includes the re-imputed alleles) and *haplotypes* (which stores the reconstructed haplotypes) and *haplotypingSummary* (which shows a summary of the haplotyping process).

B.3.3.1. alleHaplotyper Examples

Next examples depict how alleHaplotyper works:

Example B.3.5 *Haplotype reconstruction for a dataset containing parental missing data.*

```
> ## Simulation of families containing parental missing data
> simulatedFams1 = alleSimulator(2,2,6,missParProb=0.2,ungenotPars=0.4)

> ## Haplotype reconstruction of previous simulated data
> fams1List <- alleHaplotyper(simulatedFams1[[1]])

  ==========================================
  ===== alleHap package: version 0.9.6 ======
  ==========================================

  Data have been successfully loaded from:
  /home/nmr/Desktop/R-libraries/alleHap/vignettes

  ===== DATA COUNTING ======
  Number of families: 2
  Number of individuals: 8
  Number of founders: 4
  Number of children: 4
  Number of males: 4
  Number of females: 4
  Number of markers: 6
  ============================

  ======== DATA RANGES =========
  Family IDs: [FAM01,...,FAM02]
  Individual IDs: [1,...,4]
  Paternal IDs: [0,1]
  Maternal IDs: [0,2]
  Sex values: [1,2]
  Phenotype values: [1,2]
  ==============================
```

```
  ========== MISSING DATA ==========
  Missing founders: 0
  Missing ID numbers: 0
  Missing paternal IDs: 0
  Missing maternal IDs: 0
  Missing sex: 0
  Missing phenotypes: 0
  Missing alleles: 28
  Markers with missing values: 6
  ================================


  ======= IMPUTATION SUMMARY =======
  0 markers (0 alleles) have been
  turned into missing in 0 families
  due to familial inconsistencies.
  Alleles initially missing: 28
  Number of imputed alleles: 18
  Imputation rate: 0.64
  Imputation time: 0.08
  ==================================


  ========= HAPLOTYPING SUMMARY ==========
  Re-imputation rate: 0.75
  Proportion of phased alleles: 0.8333
  Proportion of non-phased alleles: 0.1667
  Proportion of missing haplotypes: 0.0625
  Proportion of partial haplotypes: 0.4375
  Proportion of full haplotypes: 0.5
  Haplotyping time: 0.027
  ========================================

> # Original data
> simulatedFams1[[1]][,-(1:6)]

Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2 Mk4_1 Mk4_2 Mk5_1 Mk5_2 Mk6_1 Mk6_2
    A     T     C     T     C     C     C     T  <NA>  <NA>     T     T
 <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
    T     A     T     T     C     T     T     T     C     C     T     T
    A     A     C     T     C     T     C     T     C     C     T     T
    A     A     C     C     C     T     C     C     C     T  <NA>  <NA>
 <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
    A     A     C     C     T     T     C     C     T     T     T     T
    A     A     C     C     C     C     C     C     C     C     C     C

# Re-imputed alleles
> fams1List['reImputedAlls'][,-(1:6)]
```

```
Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2 Mk4_1 Mk4_2 Mk5_1 Mk5_2 Mk6_1 Mk6_2
    A     T     T     C     C     C     T     C     C  <NA>     T     T
    A  <NA>     T  <NA>     T  <NA>     T  <NA>     C  <NA>     T  <NA>
    T     A     T     T     C     T     T     T     C     C     T     T
    A     A     C     T     C     T     C     T     C     C     T     T
    A     A     C     C     T     C     C     C     T     C     T     C
    A     A     C     C     T     C     C     C     T     C     T     C
    A     A     C     C     T     T     C     C     T     T     T     T
    A     A     C     C     C     C     C     C     C     C     C     C

> # Reconstructed haplotypes
> fams1List['haplotypes']

  famID indID patID matID sex phen   hap1   hap2
  FAM01     1     0     0   1    1 ?TCTCT ?CCC?T
  FAM01     2     0     0   2    2 ?TTTCT ??????
  FAM01     3     1     2   2    2 ?TCTCT ?TTTCT
  FAM01     4     1     2   1    1 A?C?CT A?T?CT
  FAM02     1     0     0   1    1 ACTCTT ACCCCC
  FAM02     2     0     0   2    1 ACTCTT ACCCCC
  FAM02     3     1     2   1    1 ACTCTT ACTCTT
  FAM02     4     1     2   2    2 ACCCCC ACCCCC
```

Example B.3.6 *Haplotype reconstruction for a dataset containing offspring missing data.*

```
> ## Simulation of families containing offspring missing data
> simulatedFams2 = alleSimulator(2,2,6,missOffProb=0.4,ungenotOffs=0.2)

> ## Haplotype reconstruction of previous simulated data
> fams2List <- alleHaplotyper(simulatedFams2[[1]],dataSummary=FALSE)

> # Original data
> simulatedFams2[[1]][,-(1:6)]

Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2 Mk4_1 Mk4_2 Mk5_1 Mk5_2 Mk6_1 Mk6_2
    C     C     A     G     T     T     C     T     T     T     G     T
    C     C     A     G     T     T     C     T     T     T     G     T
 <NA>  <NA>  <NA>  <NA>  <NA>  <NA>     C     C     T     T  <NA>  <NA>
    C     C     A     G  <NA>  <NA>     C     T  <NA>  <NA>     G     T
    T     C     G     A     C     T     C     C     C     C     T     T
    T     C     G     G     C     T     C     T     C     T     T     G
    T     C     G     G     C     T     C     T     C     T  <NA>  <NA>
 <NA>  <NA>     G     G  <NA>  <NA>     C     C     C     C  <NA>  <NA>

> # Re-imputed alleles
> fams2List['reImputedAlls'][,-(1:6)]
```

```
Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2 Mk4_1 Mk4_2 Mk5_1 Mk5_2 Mk6_1 Mk6_2
    C     C     A     G     T     T     C     T     T     T     G     T
    C     C     A     G     T     T     C     T     T     T     G     T
    C     C  <NA>  <NA>     T     T     C     C     T     T  <NA>  <NA>
    C     C     A     G     T     T     C     T     T     T     G     T
    T     C     G     A     C     T     C     C     C     C     T     T
    T     C     G     G     C     T     T     C     T     C     T     G
    T     C     G     G     C     T     C     T     C     T     T  <NA>
 <NA>  <NA>     G     G  <NA>  <NA>     C     C     C     C     T  <NA>

> # Reconstructed haplotypes
> fams2List['haplotypes']

  famID indID patID matID sex phen   hap1   hap2
  FAM01     1     0     0   1    2 C?TCT? C?TTT?
  FAM01     2     0     0   2    1 C?TCT? C?TTT?
  FAM01     3     1     2   2    1 C?TCT? C?TCT?
  FAM01     4     1     2   1    1 C?T?T? C?T?T?
  FAM02     1     0     0   1    1 ?G?CCT ?A?CCT
  FAM02     2     0     0   2    1 ?G?TT? ?G?CC?
  FAM02     3     1     2   2    1 ?G?CCT ?G?TT?
  FAM02     4     1     2   2    2 ?G?CCT ?G?CC?
```

Example B.3.7 *Haplotype reconstruction of a family containing parental and offspring missing data from a PED file.*

```
> ## PED file path
> family3path <- file.path(find.package("alleHap"),"examples","example3
    .ped")
> ## Loading of the ped file placed in previous path
> family3Alls <- alleLoader(family3path,dataSummary=FALSE)

> ## Haplotype reconstruction of previous loaded data
> family3List <- alleHaplotyper(family3Alls,dataSummary=FALSE)

> # Original data
> family3Alls

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
      1     1     0     0   2    1     C     T  <NA>  <NA>  <NA>  <NA>
      1     2     0     0   2    1  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
      1     3     1     2   1    2     C     C     A     G     A     T
      1     4     1     2   1    2     C     T     A     C  <NA>  <NA>
      1     5     1     2   1    2     C     T     A     G     C     T
      1     6     1     2   1    2     C     T     A     G     C     T
      1     7     1     2   2    1  <NA>  <NA>  <NA>  <NA>     C     G
```

```
  Mk4_1 Mk4_2 Mk5_1 Mk5_2 Mk6_1 Mk6_2 Mk7_1 Mk7_2 Mk8_1 Mk8_2
   <NA>  <NA>  <NA>  <NA>     A     C  <NA>  <NA>  <NA>  <NA>
   <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
      A     A     G     T     A     C     A     C     A     G
      A     T     C     G     C     C     C     T     C     G
   <NA>  <NA>     G     T     A     C     A     C     A     A
      A     A     G     T     A     C     A     C     A     A
      A     T     C     G  <NA>  <NA>     C     T  <NA>  <NA>

> # Re-imputed alleles
> family3List['reImputedAlls']

  famID indID patID matID sex phen Mk1_1 Mk1_2 Mk2_1 Mk2_2 Mk3_1 Mk3_2
      1     1     0     0   2    1     C     T     G     C     T     G
      1     2     0     0   2    1     C     T     A     A     A     C
      1     3     1     2   1    2     C     C     G     A     T     A
      1     4     1     2   1    2     T     C     C     A     G     A
      1     5     1     2   1    2     C     T     G     A     T     C
      1     6     1     2   1    2     C     T     G     A     T     C
      1     7     1     2   2    1     T     T     C     A     G     C
  Mk4_1 Mk4_2 Mk5_1 Mk5_2 Mk6_1 Mk6_2 Mk7_1 Mk7_2 Mk8_1 Mk8_2
      A     T     T     C     A     C     A     T     A     C
      A     A     G     G     C     C     C     C     G     A
      A     A     T     G     A     C     A     C     A     G
      T     A     C     G     C     C     T     C     C     G
      A     A     T     G     A     C     A     C     A     A
      A     A     T     G     A     C     A     C     A     A
      T     A     C     G     C     C     T     C     C     A

> # Reconstructed haplotypes
> family3List['haplotypes']

  famID indID patID matID sex phen     hap1     hap2
      1     1     0     0   2    1 CGTATAAA TCGTCCTC
      1     2     0     0   2    1 CAAAGCCG TACAGCCA
      1     3     1     2   1    2 CGTATAAA CAAAGCCG
      1     4     1     2   1    2 TCGTCCTC CAAAGCCG
      1     5     1     2   1    2 CGTATAAA TACAGCCA
      1     6     1     2   1    2 CGTATAAA TACAGCCA
      1     7     1     2   2    1 TCGTCCTC TACAGCCA
```

PART

VII

Resumen en Español

Capítulo 14

Introducción

La motivación inicial para la realización de esta tesis doctoral surgió de la necesidad de organizar y aplicar diferentes métodos bioestadísticos y/o bioinformáticos para la resolución de varios problemas planteados por el grupo de investigación en ***diabetes y endocrinología*** del *Complejo Hospitalario Universitario Insular Materno Infantil de Las Palmas de Gran Canaria*. Los problemas en cuestión estaban relacionados con la genética de la diabetes, para lo que se debían manejar datos genéticos de diferente naturaleza. Para algunas cuestiones debíamos utilizar datos de un estudio de múltiples familias (con datos genéticos de padres e hijos para un varios miles de familias). Otros de los problemas a estudiar requerían utilizar datos genéticos de un estudio de casos y controles. El proceso de análisis de estos conjuntos de datos requirió no sólo del conocimiento de métodos estadísticos e informáticos, sino también de fundamentos de la genética humana y de metodologías desarrolladas recientemente para el tratamiento avanzado de bases de datos genéticas.

La realización de estudios de asociación estadística entre datos genéticos/genómicos y la presencia de enfermedades (o fenotipos) en los individuos de una población plantea retos importantes. Por una parte los derivados el enorme número de datos que se deben manejar (fácilmente varios millones de variables, correspondientes a distintos marcadores observados a lo largo del genoma); por otro lado, los derivados de las propias técnicas de obtención de la información genética, que obligan a realizar controles de calidad sobre los datos, y a tener en cuenta características subyacentes a la composición de las poblaciones y su acervo genético. Además en muchos casos la detección de una asociación en-

tre un marcador y una enfermedad (con características hereditarias) no implica necesariamente que dicho marcador esté involucrado en el proceso causal de la enfermedad; puede ser simplemente que ese marcador se encuentre en cierta posición del genoma *ligado*, por proximidad y por la fortaleza de los enlaces químicos, al verdadero gen causal.

Otro problema frecuente es la pérdida de información en el proceso de genotipado. Este es un proceso químico que no está exento de fallos, por lo que es habitual que la base de datos que reúne la información genética de los individuos presente muchos valores perdidos. Los valores perdidos pueden, en muchos casos, determinarse mediante el adecuado uso de *técnicas de imputación*, que acuden a paneles de referencia públicamente disponibles –en la práctica auténticos mapas del contenido de nuestro genoma–, para determinar cuáles son los alelos faltantes en los marcadores con valores perdidos.

Asimismo, aún cuando se hayan observado millones de marcadores, es posible que la componente genética que se asocia a la enfermedad esté en alguna sección no directamente observada del genoma. Los métodos de imputación pueden utilizarse también para *rellenar* (*imputar*) las secciones del genoma del individuo que no han sido directamente observadas. Estos valores imputados pueden utilizarse también para evaluar posibles asociaciones con la enfermedad. Citemos, por último, que en muchas ocasiones la relación con la enfermedad tiene que ver no con marcadores más o menos dispersos a lo largo del genoma, sino con los *haplotipos* de los que dichos marcadores forman parte. Es por ello que resulta de gran interés práctico disponer de herramientas que permitan determinar los haplotipos que dan lugar a los genotipos observados.

Los métodos (algoritmos) que se utilizan habitualmente para la imputación de genotipos o la reconstrucción de haplotipos usan técnicas probabilistas para inferir el contenido de los marcadores perdidos en las bases de datos genéticas. Cuando los individuos del estudio son independientes entre sí (no comparten herencia genética más allá de la que corresponde a pertenecer a la misma especie) la única fuente de información para la imputación son los paneles de referencia, y cualquier imputación que se haga será probabilística por naturaleza: en dicho panel puede haber varias secciones distintas compatibles con la sección concreta del genoma que se quiere imputar, por lo que el valor imputado se elige como el más probable entre los posibles. Sin embargo, en el caso particular de las bases de datos familiares, cuando varios miem-

bros de la misma familia están disponibles para un estudio [1] es posible combinar la información disponible en padres, madres e hijos para determinar a partir de unos el contenido faltante en otros, teniendo en cuenta que la constitución genética de los hijos ha sido necesariamente heredada de los padres; de esta forma en estos casos la imputación de muchos valores perdidos (incluso todos), así como la identificación de los haplotipos compartidos, puede realizarse de manera determinista, disminuyendo las falsas asociaciones que pueden derivarse de los métodos de imputación probabilista.

Ciertas regiones genómicas son muy estables frente a la recombinación, al mismo tiempo que pueden llegar a ser altamente polimórficas, presentando una gran variabilidad. Una de esas regiones, que ha sido bastante bien estudiada, es la denominada *Complejo Mayor de Histocompatibilidad* (MHC). En humanos, los genes MHC conforman los denominados antígenos leucocitarios humanos (sistema HLA)[3].

Específicamente para esta región, debido a su alta tasa polimórfica, ha surgido incluso la necesidad de la creación una nomenclatura alfanumérica para facilitar los posteriores análisis. Muchos de los métodos actuales por lo general no son capaces de identificar correctamente los haplotipos en esta región, dado que la alta variabilidad hace que los métodos probabilistas que han de utilizarse cuando no se dispone de datos familiares tengan una alta tasa de error [5]. Cuando hay datos familiares disponibles la identificación de haplotipos (y la imputación de valores perdidos) es más precisa, si bien son necesarios procedimientos eficientes para ello.

Por último, cabría destacar también que aunque existe un número elevado y creciente de herramientas bioestadísticas/bioinformáticas para el procesamiento y análisis de bases de datos genómicas, la documentación relacionada con cada una de ellas es a menudo un tanto confusa y se encuentra dispersa. Particularmente para aquellos usuarios que quieren utilizar dichas herramientas por primera vez, suele resultar tediosa y complicada la puesta a puesto de los algoritmos necesarios para analizar los datos. Por lo tanto, existe una necesidad de clarificación y simplificación de la documentación relativa a los procesos que incluyen el control de calidad, la imputación de valores perdidos y/o el análisis estadístico de asociación en las bases de datos genómicas.

[1] normalmente duos formados por el padre o la madre y un hijo, tríos, con los dos padre y un hijo, o familias con dos hijos. Los estudios con más de dos hijos por familia son poco comunes

Capítulo 15

Objetivos

Los principales objetivos de esta tesis han sido:

1) **Estudio exhaustivo del estado actual de los métodos bioestadísticos/bioinformáticos** que se aplican en el manejo de bases de datos genéticas/genómicas.

2) **Aprendizaje de las técnicas y metodologías necesarias para el análisis de datos genómicos**, especificamente para las tareas de control de calidad, imputación, haplotipado, y análisis de asociación con datos poblacionales.

3) **Identificación de nuevas variantes genéticas** asociadas a la nefropatía diabética avanzada en una población de pacientes con diabetes tipo 2 de la isla de Gran Canaria.

4) Desarrollo de un paquete informático en el lenguaje R para la **imputación de genotipos perdidos y/o reconstrucción de haplotipos** en bases de datos familiares.

5) **Identificación de posibles asociaciones haplotípicas** para datos familiares, concretamente para los sujetos de la base de datos internacional T1DGC.

Capítulo 16

Planteamiento y Metodología

16.1. Planteamiento

Para el correcto desarrollo de esta tesis se planteó que tuviera dos áreas o enfoques bien diferenciados. En la primera se analizarían datos genéticos/genómicos poblacionales y en la segunda datos genéticos familiares.

16.2. Metodología

La metodología utilizada para el tratamiento y análisis de los datos genéticos poblacionales se puede dividir en los siguientes pasos:

1. Conversión de los datos *en crudo* a un formato (binario) más manejable.
2. División de los datos genómicos en cromosomas diferentes.
3. Realización del control de calidad de los datos.
4. Alineamiento y "pre-haplotipado"de las secuencias genómicas.
5. Imputación de los genotipos pertinentes.
6. Estudio de asociación estadístico.
7. Representación de los resultados del análisis de asociación.

La metodología utilizada para el tratamiento de los datos genéticos familiares se basó fundamentalmente en la implementación de un algoritmo en lenguaje R que conllevó las siguientes tareas:

1. Estudio teórico del problema de la reconstrucción de haplotipos e imputación de genotipos perdidos en familias.
2. Implementación de un simulador de datos genéticos familiares.
3. Implementación de un algoritmo para el tratamiento y carga de archivos de datos genéticos en formato *PED*.
4. Implementación de un algoritmo para imputar genotipos (marcador a marcador).
5. Implementación de un algoritmo para reconstruir haplotipos y re-imputar"genotipos (teniendo en cuenta las longitudes de los haplotipos generados).

A continuación describiremos de manera detallada los procedimientos implementados incluidos en el flujo de trabajo de alleHap, así como su implementación en funciones de R. El paquete consta de cuatro funciones principales:

- *alleLoader*: función encargada de leer datos genotípicos desde un archivo PED externo o desde un *dataframe* de R
- *alleImputer*: función encargada de imputar alelos perdidos marcador a marcador en un número indeterminado de familias.
- *alleHaplotyper*: función encargada de identificar los haplotipos a partir de los genotipos disponibles dentro de cada familia. Asimismo durante el proceso, esta función se encarga de imputar alelos perdidos a partir de la información de marcadores (completos) adyacentes.

16.2.1. AlleLoader

El cometido de esta función es leer datos genotípicos familiares desde un *dataframe* de R o desde un archivo externo, y pasar dichos datos a las funciones de imputación e identificación de haplotipos en el paquete *alleHap*. El conjunto de datos debe ajustarse a las especificaciones estándar de un archivo PED: en cada fila, las seis primeras variables corresponden a la identificación de la familia, del sujeto, del padre, de la madre, el sexo y la variable que define el fenotipo en estudio, habitualmente si el sujeto está sano o enfermo. El resto de las variables son los genotipos observados en cada marcador: cada marcador se descompone en dos variables, correspondientes a cada uno de los alelos.

La función *alleLoader* lleva a cabo el siguiente proceso:

1. Lee los datos (desde un archivo PED externo o desde un *dataframe* de R*)*

2. Identifica y recodifica como NA los valores perdidos. Estos por defecto pueden estar codificados como NA (si la entrada es un dataframe de R) y -9 o -99 (si la entrada es un archivo PED). No obstante cualquier otro valor puede ser especificado por el usuario en la llamada a la función.

3. Renombra las varaibles con los nombres esperados por las funciones de imputación e identificación de haplotipos en *alleHap*.

4. Muestra un breve resumen de los datos de entrada: número de familias, número de sujetos, número de alelos perdidos, número de fenotipos perdidos, etc.

5. Devuelve el conjunto de datos como un dataframe de R con la misma estructura que un archivo PED, con las variables renombradas y los valores perdidos correctamente identificados y codificados.

16.2.2. AlleImputer

Esta función responde a un doble propósito: primero lleva a cabo un control de calidad elemental sobre los datos genotípicos; y segundo, imputa los alelos marcador a marcador cuando sea posible. La forma en que estos procesos se llevan a cabo se describe a continuación.

16.2.2.1. AlleImputer: Control de calidad

Para cada marcador se revisa el cumplimiento de las siguientes condiciones; en caso de no cumplirse estaríamos ante un marcador con errores de genotipado, o incluso ante la posibilidad de que algún sujeto no pertenezca a la familia en que está colocado. En caso de que alguna de estas condiciones no se cumpla, el marcador completo se marca como perdido y se codifica a NA.

1. No puede haber más de dos hijos homocigotos distintos dentro de una familia.

2. Si hay dos hijos homocigotos distintos, no puede haber en ningún miembro de la familia ningún otro alelo distinto de los dos identificados en dichos hijos.

3. Considerando todos los miembros de la familia puede haber como mucho cuatro alelos distintos en un mismo marcador.

4. Si en una familia hay un total de cuatro alelos distintos, no puede haber ningún hijo homocigoto.

5. Si hay tres o más hijos heterocigotos únicos (queremos decir con ello hijos genotípicamente distintos entre sí en ese marcador) no puede haber ningún alelo común a todos los hijos.

6. En la misma familia no puede haber más de cuatro hijos genotípicamente distintos en un mismo marcador.

7. Si un hijo tiene exactamente los mismos alelos que uno de sus padres, sólo puede haber como máximo tres alelos distintos en ese marcador en la familia.

8. Cuando los alelos en los padres no están perdidos:

   a) Cada hijo debe tener al menos un alelo en común con cada progenitor.

   b) Ningún hijo puede tener alelos no presentes en al menos uno de los progenitores.

16.2.2.2. AlleImputer: Imputación de alelos marcador a marcador.

El procedimiento para llevar a cabo la imputación de alelos es el siguiente:

Paso 1: Imputación en hijos:

Si un padre es homocigoto, el alelo correspondiente es imputado en todos aquellos hijos con alelos perdidos que no tengan ya este alelo. Más aún, si ambos padres son homocigotos, todos los hijos con alelos perdidos se pueden imputar directamente.

Paso 2: Imputación en progenitores:

1. Si un hijo es homocigoto, el alelo se imputa en aquel progenitor con algún alelo perdido que no tenga ya el alelo del hijo homocigoto.
2. Si un progenitor tiene alelos perdidos y el otro no, y hay hijos heterocigotos, los alelos presentes en estos hijos y que no estén ya presentes en el padre no-perdido, se imputan al padre con alelos perdidos.

16.2.2.3. AlleImputer: implementación del algoritmo.

El núcleo de la función *alleImputer* es la función *mkrImputer*. Esta función:

1. Recibe como datos de entrado los alelos de un marcador en una familia.
2. Aplica los procedimientos de control de calidad e imputación que se acaban de describir.
3. Devuelve los marcadores ya imputados (o como estaban si no ha sido posible realizar ninguna imputación)

alleImputer tiene otras dos funciones auxiliares:

- *famImputer,* que aplica *mkrImputer* secuencialmente a todos los marcadores en una familia.
- *y famsImputer,* que aplica *famImputer* a todas las familias del dataframe.

Por tanto, el modo de operar de la función *alleImputer* puede reducirse al siguiente algoritmo:

Paso 1: Llamar a la función *alleLoader* para leer los datos genotípicos y la información familiar. Estos datos se amacenan en un dataframe de R con la misma estructura que un archivo PED.

Paso 2: Llamar a la función *famsImputer.* Esta función:

Llamar a la función *famsImputer.* Esta función:

- Identifica todas las familias en el dataframe.

- Pasa los datos de cada familia secuencialmente a la función *famImputer* función, que se encarga de realizar la imputación marcador a marcador llamando repetidamente a la función *mkrImpputer*,
- Devuelve un conjunto de datos con el mismo formato y dimensiones que el conjunto de datos de entrada, con los valores imputados en aquellos alelos donde ha sido posible llevar a cabo la imputación.

Paso 3: Opcionalmente, se mostrará un breve resumen del proceso de imputación: número de alelos imputados, incidencias detectadas (número de marcadores cancelados debido a los problemas detectados en el proceso de control de calidad), tasa de imputación (cociente de los alelos imputados sobre el número inicial de alelos perdidos) desaparecidos originalmente) y tiempo de proceso.

Paso 4: Si los datos de la familia han sido leídos desde un archivo externo, se crea un nuevo archivo con el mismo nombre, pero con extensión *imputed.ped*, que contiene el conjunto de datos devuelto por *famsImputer* en formato PED.

Paso 5: El conjunto de datos resultante de la aplicación de *famsImputer* se devuelve como un dataframe de R.

16.2.3. alleHaplotyper

Esta función tiene como objetivo identificar los haplotipos que han dado lugar a los genotipos observados en un conjunto de marcadores, en un número arbitrario de familias, cuando no hay recombinación. Consideraremos también –como de hecho sucede en la práctica– que inicialmente cuando hay datos faltantes en un marcador, faltan siempre los dos alelos (esto es, no puede haber un marcador en un sujeto con un único alelo perdido; o están perdidos los dos alelos o no está perdido ninguno). Ahora bien, si quien tiene los alelos perdidos es un hijo y uno de sus progenitores es homocigoto, por ejemplo GG, entonces una G habrá sido imputada en el hijo por *alleImputer*, con lo que el hijo presentará el genotipo G-NA. Lo mismo ocurre si el marcador completamente perdido ocurre en el padre y hay un hijo homocigoto en ese marcador.

Cuando se observan K marcadores se utilizará la siguiente notación para describir los alelos en el $i-$ésimo sujeto de la familia ($i = 1$ es el padre, $i = 2$ la madre, $i > 2$ son los descendientes):

$$\boldsymbol{A_i} = \begin{bmatrix} A_{11i} & A_{12i} & \dots & A_{1Ki} \\ A_{21i} & A_{22i} & \dots & A_{2Ki} \end{bmatrix} \tag{16.1}$$

Cada columna k de esta matriz representa un marcador, siendo (A_{1ki}, A_{2ki}) el par de alelos identificados en dicho marcador. Cada alelo (o ambos) puede estar perdido, en cuyo caso se denotaría como NA. Asociada a esta matrix se define otra matrix de *identificadores de herencia* (IDS_i) del siguiente modo:

$$\boldsymbol{IDS_i} = \begin{bmatrix} IDS_{11i} & IDS_{12i} & \dots & IDS_{1Ki} \\ IDS_{21i} & IDS_{22i} & \dots & IDS_{2Ki} \end{bmatrix} \tag{16.2}$$

donde para $h = 1, 2$, se podría afirmar que:

$$\boldsymbol{IDS_{hki}} = \begin{cases} 0 & \text{si el alelo } A_{hki} \text{ no pertence al} \\ & \text{haplotipo } h\text{, o está perdido} \\ 1 & \text{si el alelo } A_{hki} \text{ pertenece al haplotipo } h \end{cases}$$

De esta forma, si todos los términos de la matriz IDS_i son 0, la fase (haplotipo al que pertenece) de cada alelo es desconocida. A su vez, cuando todos los términos son iguales a 1, los alelos están alineados en el haplotipo al que pertenecen y las filas de la matriz A_i pueden leerse directamente como los haplotipos del $i-$ésimo miembro de la familia.

Cuando se leen los datos genotípicos de una familia, inicialmente las matrices IDS_i tienen todos sus valores idénticamente iguales a 0 para todos los miembros de la familia ya que se desconoce la fase de los genotipos. El objetivo de la función *alleHaplotyper* es ordenar los alelos A_{hki} en cada marcador de cada individuo, de tal manera que las matrices IDS_i contengan tantos valores iguales a 1 como sea posible. Cuando la fila h de IDS_i está completamente (parcialmente) rellenada con unos, la correspondiente fila h de la matriz de alelos A_i está completamente (parcialmente) ordenada, con todos sus alelos en fase.

Para lograr este objetivo, el algoritmo comienza considerando sólo los hijos, tratando de ordenar los alelos en cada marcador de tal manera que el alelo en la primera fila de la matriz A_i sea el heredado del padre, y el alelo de la segunda fila sea el heredado de la madre. De esta forma, si todos los marcadores pudieran ser ordenados de esta manera, la primera fila de la matriz A_i sería el haplotipo heredado del padre y la segunda, el

heredado de la madre. Una vez que estos haplotipos han sido identificados en los hijos, se pueden identificar fácilmente en los padres. Lo que complica esta idea y hace difícil su aplicación directa es el hecho de que en algunos casos ambos padres y un hijo comparten el mismo genotipo (digamos GT para los tres sujetos), y por lo tanto no es posible conocer qué alelo se ha heredado de qué progenitor. También puede haber alelos perdidos en los padres o hijos, lo que impide determinar la procedencia de los alelos en algunos marcadores. En particular, si ambos padres tienen todos los alelos perdidos en un marcador, es imposible determinar la procedencia de los alelos de ese marcador en los hijos, al menos si hay menos de tres hijos en la familia. Como veremos en la sección 16.2.3.1 cuando la familia tiene tres o más hijos, si no hay alelos perdidos en al menos tres hijos, es posible identificar loa haplotipos incluso cuando los progenitores está completamente perdidos. Además, en la sección 16.2.3.1 se muestra que en el caso particular de tener sólo dos hijos, si los alelos parentales están disponibles en algunos marcadores y perdidos completamente en otros marcadores, es posible bajo ciertas condiciones identificar los haplotipos que contienen los marcadores perdidos en los padres.

En las siguientes secciones consideramos cuatro escenarios para el procedimiento de identificación de haplotipos. En el primero no hay marcadores completamente perdidos en los padres, mientras que los hijos pueden no tener alelos perdidos en ningún marcador, o tener alelos perdidos completa o parcialmente en algunos marcadores; en el segundo escenario consideraremos el caso de que todos los marcadores en los padres están completamente perdidos y hay por lo menos tres hijos en la familia sin alelos perdidos; el tercer escenario es una mezcla de los dos anteriores: algunos marcadores tienen padres con alelos totalmente perdidos y por lo menos tres hijos completos (hijos sin alelos perdidos); algunos marcadores tienen alelos perdidos en los padres, y en algunos hijos; y algunos marcadores no tienen alelos perdidos en padres ni en hijos. Por último, en el cuarto escenario mostramos las condiciones en que es posible identificar parcialmente los haplotipos a partir de solo dos hijos cuando los padres tienen algunos marcadores completamente perdidos.

16.2.3.1. alleHaplotyper: Escenarios

Escenario 1: No hay marcadores con alelos completamente perdidos en los padres

El algoritmo para identificar el haplotipo al que pertenece cada alelo en este escenario es como sigue. Com los datos genotípicos de una familia:

1. Hacer $i = 3$, $k = 1$ (recuérdese que los miembros de la familia se indexan de forma que $i = 1$ el el padre, $i = 2$ la madre and $i = 3, 4, \ldots$ los hijos).

2. Dado el marcador k el el i-ésimo miembro de la familia ($i \geq 3$, por lo que solo se consideran los hijos), comprobar si es posible determinar inequívocamente para cada marcador k qué alelo ha sido heredado del padre y qué alelo de la madre. Esto puede hacerse directamente en los siguientes casos:

   a) Si el niño es homocigoto en ese marcador; los dos alelos son iguales, por lo que es trivial asignar una copia a cada padre.

   b) Si el niño tiene al menos un alelo que está presente sólo en uno de los padres (y como máximo sólo hay un alelo perdido en uno de los progenitores); ese alelo se asigna a ese progenitor y el otro alelo al otro progenitor.

   c) Si el niño tiene un alelo perdido, el otro alelo ha debido ser imputado desde un progenitor homocigoto, por lo que la procedencia de ese alelo corresponde a dicho progenitor.

   d) Si un padre es homocigoto, el alelo se ha transmitido necesariamente a todos sus hijos, por lo que este alelo en cada hijo es asignado a ese progenitor, incluso si el otro progenitor tiene alelos perdidos.

   Si los dos alelos del hijo están presentes en ambos padres (por ejemplo, el niño tiene alelos TG y ambos padres tienen también TG), no es posible determinar la procedencia de los alelos.

3. Si la procedencia de los alelos se ha determinado de manera inequívoca, colocar el alelo heredado del padre en la primera fila de la matriz A_i y el alelo heredado de la madre en la segunda fila. Hacer $IDS_{1ki} = IDS_{2ki} = 1$

4. Repetir los pasos 2 y 3 para todos los marcadores en todos los hijos de la familia.

5. Calcular las sumas por filas de las matrices IDS para cada hijo. Sea c_1 el hijo con el mayor valor de dicha suma en la primer fila de su matriz IDS_i, y sea c_2 el hijo con el mayor valor en la suma de la segunda fila. Asimismo, sea $m_{11}, m_{12}, \ldots, m_{1l_1}$ el conjunto de marcadores con IDS=1 en el hijo c_1, y $m_{21}, m_{22}, \ldots, m_{2l_2}$ el conjunto de marcadores con IDS=1 en el hijo c_2. Estos marcadores tienen ya sus alelos debidamente colocados en sus haplotipos correspondientes.

6. En el padre, ordenar los alelos en los marcadores $m_{11}, m_{12}, \ldots, m_{1l_1}$ de tal forma que la primera fila de la matriz A_1 (matriz de alelos del padre) sea igual a la primera fila de la matriz A_{c_1} (matriz de alelos del hijo c_1) en las columnas $m_{11}, m_{12}, \ldots, m_{1l_1}$. En la primera fila de IDS_1 hacer $IDS_{1k1} = 1$ para $k \in \{m_{11}, m_{12}, \ldots, m_{1l_1}\}$. En la segunda fila de IDS_1, para $k \in \{m_{11}, m_{12}, \ldots, m_{1l_1}\}$ hacer $IDS_{2k1} = 1$ si el alelo A_{2k1} no está perdido e $IDS_{2k1} = 0$ si el alelo A_{2k1} está perdido.

7. En la madre, ordenar los alelos de los marcadores $m_{21}, m_{22}, \ldots, m_{2l_2}$ de tal forma que la primera fila de la matriz A_2 (matriz de alelos de la madre) sea igual a la segunda fila de la matriz A_{c_2} (matriz de alelos del hijo c_2) en las columnas $m_{21}, m_{22}, \ldots, m_{2l_2}$. En la primera fila de IDS_2 hacer $IDS_{1k2} = 1$ para $k \in \{m_{21}, m_{22}, \ldots, m_{2l_2}\}$. En la segunda fila de IDS_2, para $k \in \{m_{21}, m_{22}, \ldots, m_{2l_2}\}$ hacer $IDS_{2k2} = 1$ si el alelo A_{2k2} no está perdido y $IDS_{2k2} = 0$ si el alelo A_{2k2} está perdido.

8. Si todos los valores en todas las matrices IDS_i son iguales a 1, **PARAR**. Todos los genotipos están identificados. En otro caso proceder a actualizar de forma iterativa las matrices A_i e IDS_i siguiendo el procedimiento descrito a continuación hasta que no hay ningún cambio en estas matrices entre dos iteraciones sucesivas. El objetivo de este procedimiento es esencialmente la localización de aquellos alelos que ya están correctamente colocados en los hijos, pero no en los padres y viceversa, y trasladar dicha información de unos a otros:

   a) Crear la matriz *idHaps*, con dos columnas y tantas filas como hijos en la familia (n)

$$\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ \vdots & \vdots \\ h_{n1} & h_{n2} \end{bmatrix} \tag{16.3}$$

donde, siendo $j = 1$ el padre y $j = 2$ la madre:

$$h_{ij} = \begin{cases} 0 & \text{si el haplotipo } j \text{ del hijo } i \text{ es el } \textbf{primer} \\ & \text{haplotipo en el progenitor } j \\ 1 & \text{si el haplotipo } j \text{ del hijo } i \text{ es el } \textbf{segundo} \\ & \text{haplotipo en el progenitor } j \\ 2 & \text{si la procedencia del haplotipo } j \text{ en el } \textbf{hijo } i \\ & \text{no puede ser determinada} \end{cases}$$

Esta matriz se crea inicialmente comparando los términos en las matrices de alelos A_i con IDS=1 entre padres e hijos. En el paso inicial es esperable que no todos los haplotipos puedan ser unívocamente determinados, por lo que esta matriz podría contener algunos ceros.

b) Comenzando en el hijo 1 y procediendo hasta el hijo n, para cada hijo i:

1) Para la fila j ($j = 1, 2$) en la matriz IDS_i calcular el vector de diferencias $D_j = (D_{j1}, D_{j2}, \ldots, D_{jK})$ entre tal fila y la fila h_{ij} en la matriz IDS del $j - esimo$ progenitor (IDS_j):
   - Un valor de -1 en la posición k de D indica que para el marcador j el correspondiente haplotipo del progenitor tiene sus alelos correctamente identificados, pero no así el hijo;
   - Un valor de 1 indica que es el hijo el que tiene los alelos correctamente identificados (colocados en sus haplotipos correspondientes), pero no así el padre.
   - Un valor de cero indica que el alelo está correctamente identificado en padre e hijo, o que no está identificado en ninguno.

2) Para aquellos marcadores donde $D_{jk} \neq 0$:
   - Si $D_{jk} = -1$: actualizar el marcador k en A_i and IDS_i a partir de la matriz de alelos A_j del padre . Esta actualización cosiste en comprobar si el alelo en la posición *jk* de la matriz A_i (hijo) coincide con el alelo en la posición k de la fila h_{ij} en A_j (progenitor). Seaa_c el alelos en el hijo y a_p el alelo en el progenitor. Sea también b_c el otro alelo en ese marcador en el hijo.
   - Si $D_{jk} = 1$ seguir el mismo procedimiento que para $D_{jk} = -1$ pero intercambiando los papeles de progenitor e hijo.
3) Si, como resultado del paso anterior, las matrices IDS de los padres cambian, y hay algún cero en la matriz *idHaps,* entonces esta matriz debe revisarse para comprobar si los haplotipos no identificados aún en hijos pueden ser ahora emparejados con alguno de los haplotipos de los padres, actualizando *idHaps* en consecuencia .

Un marcador se considera no informativo si es heterocigoto con los mismos dos alelos en todos los miembros de la familia (ya que en tales condiciones es imposible determinar qué alelo viene del padre y qué alelo de la madre). Como resultado del algoritmo anterior, todos los marcadores informativos habrán identificado cuál de sus alelos procede del padre y cuál de la madre.

Nótese también qu een el paso 2 de este último procedimiento, algunos alelos **no imputados previamente** por *alleImputer* pueden quedar imputados en padres y/o hijos al miosmo tiempo que se realiza el proceso de identificación de haplotipos. s.

Escenario 2: Alelos completamente perdidos en los progenitores

En este escenario surgen nuevas dificultades: los algoritmos anteriores se basan en realizar constantes comparaciones entre los alelos presentes en los padres y en sus hijos para lograr el objetivo final de identificar los haplotipos; cuando los padres tienen todos sus alelos perdidos tales comparaciones sencillamente no pueden realizarse. Ahora bien, si hay al menos tres descendientes genotípicamente diferentes entre sí, entonces

sí que es posible determinar los haplotipos en los hijos, aunque no será posible saber cual es el haplotipo que proviene del padre y cual proviene de la madre.

Para entender cómo funciona el procedimiento, hay que tener en cuenta que las combinaciones de haplotipos en los padres para las cuales puede haber al menos tres posibles hijos genotípicamente diferentes en la familia son las que se muestran en la tabla 16.1. **Supondremos inicialmente que no hay alelos perdidos en los hijos.**

	Caso I	**Caso II**	**Caso III**
Progenitor 1	A/B	A/B	A/B
Progenitor 2	A/B	A/C	C/D
Posible Descendencia	A|A A|B B|B	A|A A|B A|C B|C	A|C A|D B|C B|D

Tabla 16.1: Configuraciones haplotípicas cuando son posibles al menos tres hijos genotípicamente diferentes.

En la tabla 16.1 puede observarse fácilmente que para cualquier combinación de tres hijos en cualquiera de los casos anteriores, siempre hay al menos un haplotipo común compartido entre dos de ellos. La idea para la identificación de los haplotipos sin conocer los alelos de los padres comienza con la identificación del haplotipo compartido entre dos hijos. Una vez que se identifica este haplotipo, automáticamente quedan también identificados los correspondientes haplotipos complementarias. Si los dos hijos seleccionados son heterocigotos, habríamos identificado el haplotipo común y los dos complementarios, en total tres haplotipos. Si uno de los dos hijos fuese homocigoto, habíamos terminado con encontrando dos haplotipos diferentes (el haplotipo repetido que posee el hijo homocigoto y el complementario en el otro hijo). Volviendo nuevamente la tabla 16.1, observamos también que cualquiera que sea el grupo de dos hijos con el que comience el algoritmo, en el tercer hijo debe estar presente al menos uno de los haplotipos identificados en aquellos dos.

De esta manera, el algoritmo para encontrar los haplotipos a partir de tres hijos sin alelos perdidos, y sin conocer a los padres, procede como sigue:

1. Dados tres hijos genotípicamente distintos, 1, 2 y 3, para cada marcador $k = 1, 2, \ldots, K$ encontrar los alelos comunes entre los hijos 1 y 2, 1 y 3 y 2 y 3. En aquellos en que existan tales alelos comunes existan, construir los correspondientes haplotipos , así como sus haplotipos complementarios. Denotemos por $H_{ij}^{(m)}$ un conjunto de haplotipos encontrado de esta manera a partir de los hijos i y j. Nótese que, en muchos casos, dependiendo del número de alelos en cada marcador, puede haber más de un conjunto de dichos haplotipos derivados de los genotipos de los hijos i y j. También algunas veces, cuando la pareja de hijos seleccionada tenga los haplotipos A/C y B/D, no habrá ningún haplotipo común a ambos hijos. En cualquier caso, si todos los $H_{ij}^{(m)}$ están vacíos es que ha habido un error de genotipado o una recombinación. En ambos casos se genera una *incidencia* (se informa de la situación al usuario creando una entrada en una lista de incidencias) y el algoritmo se detiene.

2. En otro caso, para cada $H_{ij}^{(m)} \neq \emptyset$ determinar si al menos uno de los haplotipos del conjunto $H_{ij}^{(m)}$ está también presente en el tercer hijo:

   - Si no se cumple esta condición, se crea una incidencia y el algoritmo se detiene.
   - Si hay más de un conjunto de haplotipos que cumplen esta condición (esto puede ocurrir dependiendo de la configuración alélica, que podría ser compatible con varias estructuras de haplotipos distintas) entonces no hay una solución única y algoritmo se detiene sin haber identificado los haplotipos.
   - Si sólo hay un conjunto de haplotipos H_{ij} que cumpla esta condición, ir al paso 3.

3. *Si hay más de 3 hijos en la familia*, identificar el conjunto de haplotipos que han sido encontrado en el paso 2 con un juego de tres haplotipos compatible con alguno de los casos que aparecen en la tabla 16.1 y determinar los haplotipos en el progenitor 1 y en el progenitor 2 (no será posible saber quien es el padre y quien la madre). Emparejar estos haplotipos parentales con el resto de los hijos, determinando qué haplotipos particulares están presentes en cada hijo. En caso de que algún hijo tenga alelos perdidos, imputarlos cuando sea posible.

4. PARAR y devolver, para cada hijo, la pareja de haplotipos que se ha identificado. Esto significa devolver las matrices A_i de cada hijo, con los alelos de cada haplotipo en una fila, y poner unos en aquellas posiciones de la matriz IDS_i para aquellos alelos que han sido ubicados en su haplotipo correspondiente.

Para que este algoritmo pueda funcionar es necesario que por lo menos tres hijos tienen todos sus alelos completos, sin valores perdidos. El resto de los hijos (en el caso de que la familia tenga más de tres hijos) pueden tener alelos perdidos, que podrían resultar imputados en el paso 3 del algoritmo en caso de que sus alelos no perdidos fuesen compatibles con una única posible pareja de haplotipos parentales; si los genotipos de los hijos con alelos perdidos encajan con varias posibles parejas distintas de haplotipos de los padres, entonces no es posible determinar qué haplotipos concretos son los que portan dichos hijos.

Escenario 3: Combinación de los dos escenarios anteriores

El proceso de construcción de haplotipos en este escenario es obviamente más complicado que en los otros dos casos, y el algoritmo para la identificación de haplotipos es una mezcal de los dos anteriores. Llamaremos *algoritmo 1* al algoritmo descrito en el primer escenario, *y algoritmo 2* al segundo.

1. Si no hay ningún marcador con alelos perdidos en ambos progenitores, aplicar el algoritmo 1.

2. Contar el número de unos en las matrices IDS_i para todos los miembros de la familia. Sea *IDSNr* ese número.

3. Si hay marcadores con alelos perdidos en ambos padres, localizar combinaciones de marcadores que tengan al menos tres hijos completamente genotipados sin valores perdidos, y que incluyan al menos un marcador con padres completamente perdidos. Sea $S = \{S_1, S_2, \ldots, S_r\}$ el conjunto de tales combinaciones. Los S_i tales que $S_i \subseteq \bigcup_{j \neq i} S_j$ son eliminados de S.

4. Si $S = \emptyset$ PARAR. Si no, ordenar los conjuntos S_i en orden decreciente de su tamaño. Para $i = 1$ hasta r:

a) Aplicar el algoritmo 2 a los marcadores en el conjunto S_i. En particular, imputar todos los alelos perdidos que sea posible en hijos, como se indica en el paso 3 del algoritmo 2. La idea es que cuanto más completos estén los marcadores, mejor funcionarán ambos algorimtos.

b) Si hay uno o más marcadores no completamente perdidos en los progenitores en S_i, utilizar dichos marcadores para propcurar *alinear* a los padres; tal como hemos visto, el algoritmo 2 identifica haplotipos parentales, aunque no distingue entre padre y madre (de hecho ese algoritmo ni siquiera utiliza datos de los padres); cuando es posible emparejar tales haplotipos de forma única con los marcadores no perdidos en los padres, entonces sí que será posible determinar qué haplotipo procede de cada progenitor. Si tal único emparejamiento es posible, entonces impuar los correspondientes alelos en los marcadores perdidos de los padres.

c) Si no ha sido posible alinear los padres y hay marcadores *en fase* (ya colocados en sus haplotipos de origen) fuera de S_i, tratar de emparejar los haplotipos hallados para los marcadores en S_i con los alelos en fase fuera de S_i. Si es posible un único emparejamiento, entonces imputar los alelos correspondientes en los marcadores perdidos de los padres.

d) Si se ha realizado alguna imputación en *a)*, *b)* o *c)*, aplicar el algoritmo 1 a S_i,

5. Aplicar el algoritmo 1 a todos los marcadores de la familia. Contar el número de unos en las matrices IDS_i para todos los miembros de la familia. Sea *IDSNrNew* tal número. *Si IDSNrNew=IDSNr*, PARAR. En caso contrario, hacer *IDSNr=IDSNrNew* e ir al paso 3.

Cuando este algoritmo para, los alelos en fase son aquellos con $IDS = 1$. Si no todos los *IDS=1* solo habrá sido posible construir parcialmente los haplotipos.

Escenario 4: Identificación de haplotipos con marcadores perdidos en padres y solamente dos hijos genotipados

En este escenario se considera el caso en el que hay algunos marcadores $Mk_1, Mk_2, \ldots, Mk_n$ para los que padres e hijos tienen sus

alelos ya en fase. Sean (F_1, F_2) y (M_1, M_2) los haplotipos ya identificados en dichos marcadores, respectivamente, en el padre y en la madre. Consideremos ahora que en otro marcador M los genotipos de ambos padres están totalmente perdidos, pero que hay dos hijos para los cuales se conocen los genotipos de ese marcador, y que además esos hijos son heterocigotos en ese marcador. Hay cuatro posibles maneras en que los hijos podrían haber heredado los haplotipos de los marcadores que ya están en fase: $(F_1M_1,\ F_1M_2,\ F_2M_1,\ F_2M_2)$. Para cada una de estas maneras, los genotipos en el marcador M en cada hijo pueden aparecer de tres formas alternativas: dos hijos heterocigotos iguales, dos hijos heterocigotos compartiendo un alelo o dos hijos heterocigotos sin alelos comunes. Todas las situaciones posibles, teniendo en cuenta también los haplotipos heredados en los marcadores ya en fase se resumen en la tabla 16.2.

Caso	Descripción	Sujeto	Haps. heredados en marcadores con fase	Genotipos heredados en marcadores sin fase		
				Opción 1	*Opción 2*	*Opción 3*
1	Hijos que no comparten ningún haplotipo	Hijo 1	F_1, M_1	A/B	A/B	A/B
		Hijo 2	F_2, M_2	A/B	A/C	C/D
2	Hijos que comparten un haplotipo del padre	Hijo 1	F_1, M_1	A/B	A/B	A/B
		Hijo 2	F_1, M_2	A/B	A/C	C/D
3	Hijos comparten un haplotipo heredado de la madre	Hijo 1	F_1, M_1	A/B	A/B	A/B
		Hijo 2	F_2, M_1	A/B	A/C	C/D
4	Hijos comparten ambos haplotipos	Hijo 1	F_1, M_1	A/B	A/B	A/B
		Hijo 2	F_1, M_1	A/B	A/C	C/D

Tabla 16.2: Descripción de los casos para el escenario 4.

1. Si suponemos que $F_1 \neq F_2, M_1 \neq M_2$ y$\{F_1, F_2\} \cap \{M_1, M_2\} = \emptyset$, entonces:

    - En el caso 1: no es posible identificar inequívocamente los haplotipos que resultan de la combinación de los marcadores en fase $Mk_1, \ldots Mk_n$ con M, pues cada uno de los F_i

y M_i ha sido observado una única vez y por tanto cualquier emparejamiento entre estos haplotipos y los alelos de M sería compatible con los datos observados.

- En el caso 2:
  - Con la opción 1 no es posible decidir, ya que no puede determinarse si F_1 se empareja con A (y (M_1, M_2) con B) ó F_1 se empareja con B (y (M_1, M_2) con A). De esta forma la fase del marcador M no puede determinarse unívocamente.
  - Con la opción 2, sería posible identificar un haplotipo paterno F_1A y los dos maternos M_1B y M_1C, ya que éstos son los únicos haplotipos compatibles con los datos observados.
  - La opción 3 no puede darse en este caso; si ocurre es porque puede haber habido un error de genotipado (o bien ha habido un error de genotipado en los haplotipos construidos para $Mk_1, \ldots, Mk_n$), o puede haber habido una recombinación.
- El caso 3 es análogo al anterior:
  - Con la opción 1 no es posible decidir.
  - Con la opción 2 se identifica el haplotipo materno M_1A y los dos paternos F_1B y F_1C.
  - La opción 3 no es posible salvo error o recombinación.
- En el caso 4:
  - Con la opción 1 no es posible decidir: los haplotipos podrían ser F_1A y M_1B o bien F_1B y M_1A.
  - Las opciones 2 y 3 no serían posibles salvo error o recombinación.

2. Si $F_1 = F_2$, $M_1 \neq M_2$ y que $\{F_1\} \cap \{M_1, M_2\} = \emptyset$. Entonces:
   - Igual que antes, en el caso 1 no se puede decidir.
   - En el caso 2, si $F_1 = F_2$, dicho haplotipo podría ir acompañado de dos alelos distintos en M, por lo que tampoco se puede decidir.
   - En el caso 3, opción 2, se pueden determinar los dos haplotipos del padre y uno de la madre: M_1A, F_1B, F_1C.

- La opción 3 no es posible en este caso tampoco (por error o recombinación)
- Como $F_1 = F_2$, el caso 4 es equivalente al 3.

3. Si $F_1 \neq F_2$, $M_1 = M_2$ y que $\{F_1, F_2\} \cap \{M_1\} = \emptyset$. Entonces:

   - Igual que antes, en el caso 1 no se puede decidir.
   - En el caso 2, opción 2, se pueden determinar los dos haplotipos de la madre y uno del padre: F_1A, M_1B, M_2C.
   - La opción 3 no es posible en este caso tampoco (por error o recombinación)
   - Como $M_1 = M_2$, el caso 4 es equivalente al 2.

4. Si $F_1 \neq F_2$, $M_1 \neq M_2$, $F_1 = M_1 = W_1$, $F_2 \neq M_2$

   - En el caso 1 es posible determinar que el primer hijo tiene los haplotipos W_1A y W_1B; sin embargo no es posible saber qué haplotipo procede del padre y cuál de la madre.; esto vale para las tres opciones.
   - En el caso 2, nuevamente en las tres opciones es posible determinar los haplotipos del primer hijo; en la opción 2 además es posible determinar que los haplotipos de la madre son W_1B y M_2C, y que uno de los haplotipos del padre es W_1A (en efecto, en el hijo 2, el haplotipo W_1 viene necesariamente del padre; si al juntarlo con el otro marcador, el haplotipo paterno en el hijo 2 fueseW_1C, ello implicaría que en el hijo 1 el haplotipo paterno debe ser W_1A o W_1B; ahora bien, ello no es posible ya que como el padre ya tiene dos haplotipos distintos W_1 y F_2, no puede ser que haya dos haplotipos distintos que empiecen con W_1, ya que con el F_2 se tendrían ya tres haplotipos; por tanto, necesariamente el haplotipo paterno ha de ser W_1A, y por descarte se obtienen los haplotipos de la madre). La opción 3 del caso 2 no es posible salvo error o recombinación.
   - El caso 3 es similar al anterior; en este caso, además de poder especificar los haplotipos del hijo1, aún sin saber de qué progenitor proceden; además, en la opción 2 se puede deducir que un haplotipo de la madre es W_1A y que los haplotipos del padre son F_2C y W_1B.

- En el caso cuatro es posible determinar los haplotipos de los dos hijos, si bien no es posible especificar de cada haplotipo si procede del padre o de la madre; es más, la única opción posible sería la 1, ya que la 2 y la tres solo puede ocurrir por error de genotipado o recombinación.

5. If $F_1 = M_1 = W_1$, $F_2 = M_2 = W_2$, $W_1 \neq W_2$ or $F_1 = F_2 = F$, $M_1 = M_2 = M$, $F = M = W$

   - En los casos de haplotipo homocigoto se pueden decidir los haplotipos completos de los hijos, pero no de los padres.
   - El caso 1 opción 3 es imposible salvo error o recombinación.
   - Las tres opciones del caso 2 son posibles pero no permiten determinar haplotipos en los padres, solo en los hijos.
   - En el caso 3 las opciones 1 y 2 son posibles, pero no permiten encontrar haplotipos en los padres; la opción 3 es imposible salvo error o recombinación.

6. If $F_1 = F_2 = F$, $M_1 = M_2 = M$, $F \neq M$, or $F_1 = F_2 = F$, $M_1 \neq M_2$, $\{F\} \cap \{M_1, M_2\} = \emptyset$, or $F_1 \neq F_2 = F$, $M_1 = M_2 = M$, $\{F_1, F_2\} \cap \{M\} = \emptyset$

   - En estos casos no pueden deducirse haplotipos completos ni en padres ni en hijos.

Resumiendo este exhaustivo análisis, los casos en los que los haplotipos pueden ser unívocamente determinados para un conjunto de marcadores $Mk_1, Mk_2, \ldots, Mk_n$ ya en fase, y un nuevo M que tenga alelos completamente perdidos en los padres (y dos hijos genotipados sin valores perdidos) son los mostrados en la tabla 16.3.

Haplotipos paternos en marcadores con fase	Haplotipos heredados por hijos en marcadores con fase	Genotipos de hijos en marcadores sin fase	Genotipos con fase completos en padres	Genotipos con fase completos en padres
(F_1, F_2) , (M_1, M_2)	(F_1, M_1), (F_1, M_2)	A/B, A/C	$(F_1A \mid F_2NA)$, $(M_1B \mid M_2C)$	$(F_1A \mid M_1B)$, $(F_1A \mid M_2C)$
(F_1, F_2) , (M_1, M_2)	(F_1, M_1), (F_2, M_1)	A/B, A/C	$(F_1B \mid F_2C)$, $(M_1A \mid M_2NA)$	$(F_1B \mid M_1A)$, $(F_2C \mid M_1A)$
(F, F) ,(M_1, M_2)	(F, M_1) , (F, M_1)	A/B, A/C	$(FB \mid FC)$, $(M_1A \mid M_2NA)$	$(FB \mid M_1A)$, $(FC \mid M_1A)$
(F_1, F_2) , (M, M)	(F_1, M), (F_1, M)	A/B, A/C	$(F_1A \mid F_2NA)$, $(MB \mid MC)$	$(F_1A \mid MB)$, $(F_1A \mid MC)$
(W_1, F_2), (W_1, M_2)	(W_1, W_1), (W_1, M_2)	A/B, A/C	$(W_1A \mid F_2NA)$, $(W_1B \mid M_2C)$	$(W_1A \mid W_1B)$, $(W_1A \mid M_2C)$
(W_1, F_2), (W_1, M_2)	(W_1, W_1), (F_2, W_1)	A/B, A/C	$(W_1B \mid F_2C)$, $(W_1A \mid M_2NA)$	$(W_1B \mid W_1A)$, $(F_2C \mid W_1A)$

Tabla 16.3: Muestra detallada de aquellos casos en los que es posible obtener la fase de los alelos en marcadores sin fase con genotipos paternales perdidos (usando la información de la fase de los marcadores adyacentes de dos hijos genotípicamente diferentes).

Esta tabla puede implementarse fácilmente en forma de un algoritmo mediante una cadena de condiciones Si-Entonces.

16.2.3.2. alleHaplotyper: Implementación

El núcleo de la función *alleHaplotyper* es la función *famHaplotyper*.

Esta función:

1. Recibe como datos de entrada la matriz de datos imputados devueltos por *alleImputer* para **una** familia.
2. Aplica los algoritmos descritos en el escenario 3 anterior (téngase en cuenta que este algoritmo se adapta también a los escenarios 1 y 2) o en el escenario 4, de acuerdo a la disponibilidad de hijos y de información genotípica.
3. Devuelve:
   a) Una matriz igual a la matriz de entrada, pero con los nuevos alelos imputados
   b) Una matriz con las mismas dimensiones que la anterior llena de ceros y unos. El valor cero indica un alelo que no está en fase y el 1 que sí lo está.
   c) Una matriz con dos columnas que se corresponden con los haplotipos encontrados en cada miembro de la familia.

   Como función auxiliar, *alleHaplotyper* incluye también la función *famsHaplotyper,* encargada de aplicar *famHaplotyper* secuencialmente a todas las familias en el dataframe.

De este modo, el funcionamiento de *alleHaplotyper* puede reducirse al siguiente algoritmo:

1. Llamar a la función *alleImputer* para leer los datos familiares e imputar marcador a marcador todos aquellos alelos que sea posible.
2. Llamar a la función *famsImputer.* Esta función:
   - Identifica todas las familias en el dataframe.
   - Pasa secuencialmente los datos de cada familia a la función *famHaplotyper,* que lleva a cabo la identificación de haplotipos aplicando los algoritmos descritos en la sección anterior.

- Devuelve una lista que contiene el conjunto de datos original, los datos genotípicos imputados por *alleImputer*, los imputados por *alleHaplotyper*, la matriz IDS con ceros y unos, y los haplotipos (completos o parciales) hallados en todos los sujetos.

3. Opcionalmente, muestra un breve resumen del proceso de identificación de haplotipos, que contiene la tasa de imputación final conseguida tras el haplotipado, la proporción de alelos en fase, las proporciones de haplotipos completos, parciales y perdidos, y el tiempo empleado en todo el proceso .

El paquete alleHap , tal como se encuentra en CRAN, dispone de una *vignette* que explica su funcionamiento y contiene numerosos ejemplos que permiten ver parte de la casuística a la que nos hemos enfrentado en su desarrollo y que se ha contemplado en los algoritmos anteriores.

Capítulo 17

Resultados

17.1. Resultados del análisis de bases de datos poblacionales

Con respecto al análisis de datos poblacionales, el trabajo que abordamos consistión en identificar las variantes genéticas con mayor significación asociadas a la nefropatía diabética avanzada en la diabetes tipo 2 (T2D) en la población de la isla de Gran Canaria. Para ello dispusimos de una base de datos en la que se disponía del genotipo de algo más de 4 millones de marcadores en 110 sujetos –todos afectados con diabetes tipo 2– clasificados en casos o controles (55 sujetos en cada grupo) según que tuviesen una nefropatía avanzada o no. El objetivo final era determinar si hay condiciones genéticas diferentes entre ambos grupos que pudieran estar relacionadas con el avance de la nefropatía. Esta es la fase preliminar de un estudio más amplio; los marcadores que puedan detectarse en esta criba deberán ser posteriormente observados en otra muestra independiente que permita confirmar -o no- la validez de la asociación detectada.

El análisis de estos datos requirió la realización de varias fases: control de calidad por muestras y marcadores (eliminando los marcadores y sujetos que no superan los requisitos de calidad), imputación de valores perdidos y marcadores adyacentes a los observados, y ejecución del análisis estadístico de asociación con la base de datos resultante. Debemos señalar que el bajo tamaño de nuestra muestra dificulta la detección de posibles asociaciones, por lo que en la fase de control de calidad hemos relajado alguna de las condiciones sobre la selección de individuos para

no reducir aún más el tamaño muestral.

Describimos a continuación brevemente las distintas fases de este proceso.

17.1.1. Resultados del control de calidad

Los resultados del control de calidad obtenidos incluyen medidas de calidad por muestras (individuos) y por variantes (marcadores).

17.1.1.1. Medidas para el control de calidad de marcadores

Las medidas para el **control de calidad de marcadores** que hemos considerado son las siguientes: ***eficiencia de genotipado*** (proporción de genotipos perdidos por marcador), ***MAF*** (frecuencia del los alelos alternativos) and ***HWE*** (frecuencias alélicas (genotípicas) que permanecen constantes en una población de una generación a la siguiente).

17.1.1.1.1. Eficiencia de genotipado

La figura 17.1 se ha generado para investigar y/o analizar la **eficiencia de genotipado** (también denominada como *SNP coverage*).

Para la evaluación de la eficiencia o tasas de genotipado, en nuestro caso, y de acuerdo con la figura 17.1, hemos utilizado 0.1 como límite máximo aceptable de pérdidas. Con este umbral, han sido conservados solo aquellos SNPs con menos de un 10 % de tasa de pérdidas (o más de un 90 % de eficiencia de genotipado).

Finalmente, después del análisis de los genotipos perdidos, se obtuvo que la tasa/eficiencia de genotipado para todos los individuos fue 0.97.

17.1.1.1.2. Frecuencia de alelos alternativos

Para el evaluar las **frecuencia de los alelos alternativos** (MAF) hemos elegido un umbral dependiendo del número de de sujetos (n), si bien no la relación habitual $MAF = 10/n$, puesto el número de sujetos disponibles en el estudio era bajo, sino que hemos seleccionado el umbral resultante de la siguiente expresión:

$$MAF = \frac{1}{n \times 2} = \frac{1}{110 \times 2} = 0{,}0045 \tag{17.1}$$

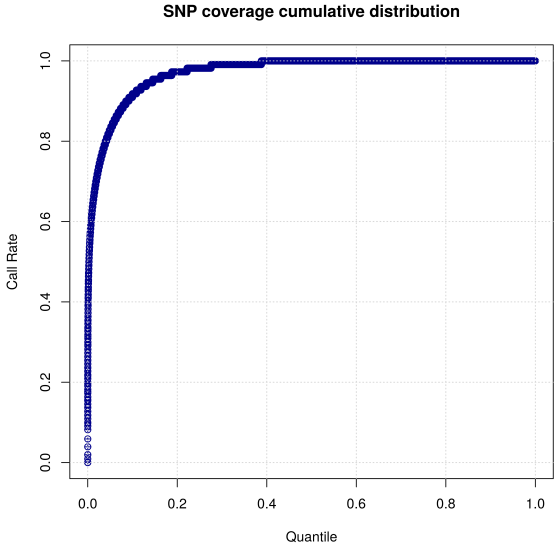


Figura 17.1: Distribución acumulativa de SNPs

Con este umbral de frecuencias alélicas alternativas, todos las variantes alélicas homocigotas (SNPs homozigotos) han sido excluidas de nuestro conjunto de datos, ya que durante análisis posteriores (específicamente en los test de asociación) estos valores no aportan información por ser iguales en casos y cotroles. Como consecuencia de este del control de calidad, se eliminaron un total de 484556 SNPs.

17.1.1.1.3. Equilibrio de Hardy-Weinberg

El **principio de Hardy-Weinberg** establece que la variación genética en una población se mantendrá constante de una generación a la siguiente, en ausencia de factores perturbadores. Ello se traduce en que la frecuencia relativa con que se observan los genotipos debe ser igual al producto de las frecuencias relativas de los alelos que los forman. Cuando un marcador no se encuentra en equilibrio de Hardy-Weinberg puede ser confundido con un marcador asociado a la enfermedad, por lo que los marcadores fuera de esta condición deben eliminarse del estudio.

Para evaluar la desviación de los sujetos controles (de nuestro estudio caso-control), hemos generado un gráfico Cuantil-Cuantil (QQ) para

poder apreciar si existe desviación en cada SNP.

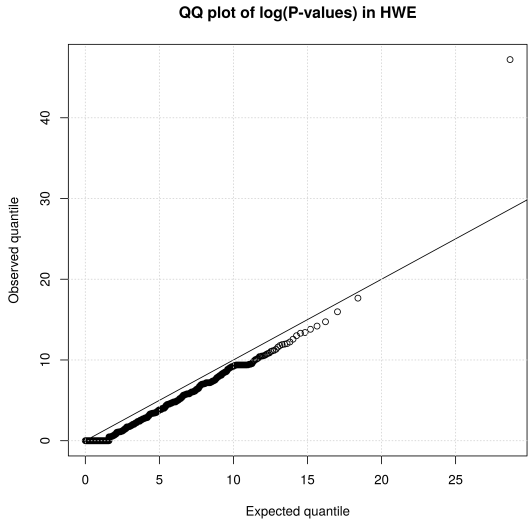


Figura 17.2: Gráfico cuantil-cuantil de sujetos controles de aquellos p-valores en Equilibrio de Hardy Weinberg.

Del gráfico anterior se puede apreciar cómo existe un SNP con un p-valor desviado considerablemente del equilibrio de Hardy Weinberg. Dicho SNP (denominado rs114833138) está localizado en la posición 186204334 of the chromosome 4. Esta desviación tan grande podría indicar un error de genotipado.

17.1.1.2. Medidas para el control de calidad de muestras

Las medidas para el **control de calidad de muestras** (sujetos) que hemos considerado son: ***tasa de pérdidas*** (proporción de genotipos perdidos por sujeto), ***discordancias de género*** (comprobación de la concordancia entre el género que se puede extraer del análisis de los genotipos y el de la identificaión del individuo), ***estratificación de la población*** (individuos con un origen genético significativamente diferente del resto de la muestra de estudio), ***tasa de heterocigosidad*** (proporción de genotipos heterocigóticos para un individuo dado) y ***parentesco entre individuos*** (comprueba si los sujetos son familiares cercanos).

17.1.1.2.1. Tasa de pérdidas

Para el estudio de la **tasa de pérdidas por individuo**, establecimos un umbral en donde se detetaba un cambio cualitativo en la tasa de pérdida de datos.

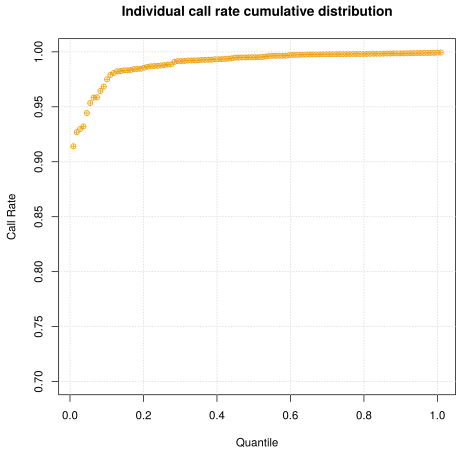


Figura 17.3: Distribución acumulativa del *Call Rate* (1 - tasa de pérdidas) por individuo.

La figura 17.3 muestra la distribución acumulativa del Call Rate (1 - tasa de pérdida) de los genotipos de los individuos, con el fin de investigar la proporción de SNPs genotipados perdidos.

Basándose en la figura 17.3, a partir de un *call rate* >91 %, todas los individups tienen una buena calidad de genotipado. Siendo más rigurosos (en términos de *call rate* por muestra), y eligiendo un umbral >97 %, tendríamos que eliminar 11 de 110 muestras de nuestro estudio, es decir, el 10 % del número total de individuos genotipados, con lo que decidimos no usar dicho umbral tan estricto.

17.1.1.2.2. Discordancias de género

Este paso del control de calidad se realiza para comprobar que el sexo declarado de los individuos coincide con el determinado por su número de cromosomas X.

Si existe un número elevado de **discordancias de género**, se puede suponer que todos los identificadores de ejemplo podrían haberse mezclado de alguna manera. En nuestro caso, sólo había una discordancia, por lo que asumimos que la mayoría de los identificadores de los sujetos fueron asignados correctamente entre datos clínicos y los genéticos.

17.1.1.2.3. Estratificación de la población

Para el estudio de la **estratificación de la población** (o detección de valores atípicos étnicos) cabe destacar que, cuando las muestras de estudio comprenden múltiples grupos de individuos que difieren sistemáticamente en tanto ascendencia genética como en fenotipos, suelen aparecer asociaciones espúrias entre las poblaciones mezcladas. Estas pueden deberse a diferencias en la ascendencia y no a una verdadera asociación genética con la enfermedad, lo que conllevaría tanto a falsos positivos como a falsos negativos [62]. Por tanto, aunque este es un paso importante en el control de calidad de datos poblacionales en muestras muy grandes donde cabe esperar mezclas étnicas, en nuestro caso decidimos que no era necesario tenerlo en cuenta, puesto que todos los sujetos genotipados pertenecían a la misma población.

17.1.1.2.4. Parentesco entre individuos

El **parentesco entre individuos** ocurre cuando parejas o grupos de sujetos están más estrechamente relacionados entre sí que la media de la población, lo que indica que son familiares cercanos [157]. Tales individuos suelen presentar correlaciones que pueden provocar asociaciones erróneas (falsos positivos o falsos negativos).

Así, de acuerdo a la figura 17.5, se puede apreciar que claramente existen dos parejas de sujetos que son familiares entre sí, con lo que uno de cada pareja tuvo que eliminarse. El criterio aplicado para decidir qué muestra eliminar, fue seleccionar aquel con la mayor proporción de datos (SNPs) perdidos.

17.1.1.2.5. Tasa de heterocigosidad

Para analizar la **tasa de heterozigosidad (H)** hemos representado en la misma figura sus correspondientes valores con los de *coeficiente de consanguinidad de Wright (F)*, donde una F positiva indicaría un exceso de

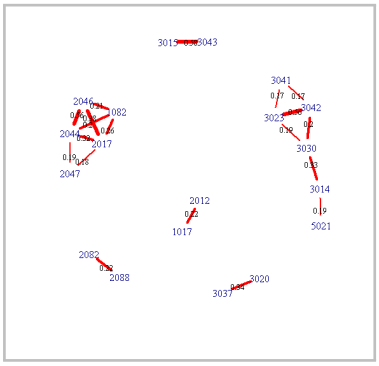


Figura 17.4: Ejemplo de una red de parentesco más compleja (aparecen varias relaciones de parentesco) [161].

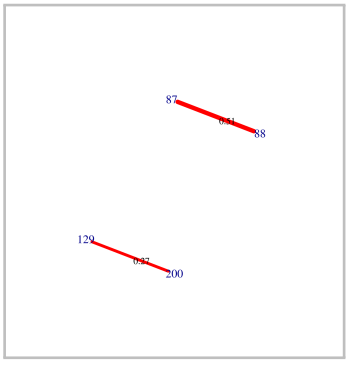


Figura 17.5: Red de parentesco entre sujetos de nuestro estudio (aparecen dos relaciones de parentesco).

homocigotos (baja heterocigosidad), y una F negativa indicaría un exceso de heterocigotos (alta heterocigosidad) [157]. La presencia de una F inusualmente alta en un individuo podría indicar que ha habido una problema de genotipado o que la muestra provenía de una población diferente, y por lo tanto debería ser eliminada.

Mediante la representación de la figura 17.6 se puede identificar la existencia de individuos con una inusual tasa de heterocigosidad.

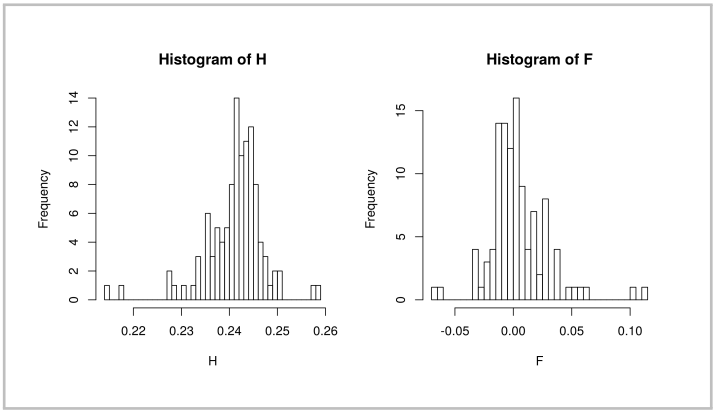


Figura 17.6: Histogramas de heterocigosidad H, y valores inversamente proporcionales F (antes del control de calidad de muestras).

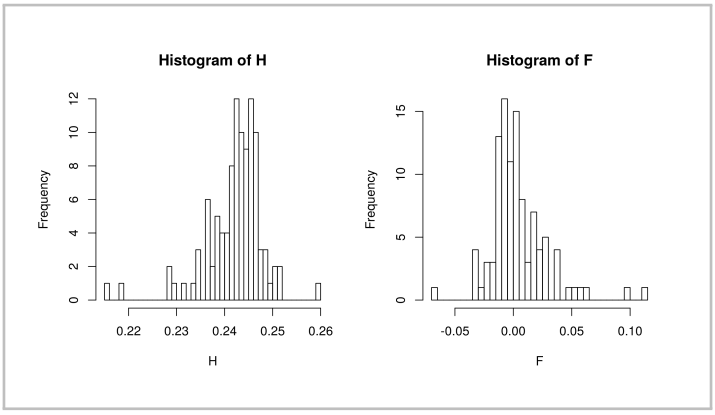


Figura 17.7: Histogramas de heterocigosidad H, y valores inversamente proporcionales F (después del control de calidad de muestras).

Los histogramas de heterocigosidad para antes y después del control de calidad de muestras se agrupan en las figuras 17.6 y 17.7, respectivamente. En ambas figuras se puede apreciar como existen algunos valores atípicos, que superan 3 veces la desviación estándar, que es el límite habitual utilizado para analizar la heterocigosidad [158]. Aunque estos valores H y F estaban fuera de los límites, decidimos no eliminar los correspondientes sujetos, ya que esto hubiera desiquilibrado el número de muestras de casos y controles.

17.1.2. Imputación

Una vez realizada la fase de control de calidad, se ha procedido a la imputación, tanto de los alelos perdidos en los marcadores observados como de los alelos presentes en marcadores no observados, que fue posible añadir a nuestra base de datos mediante el uso de paneles de referencia. Como panel de referencia hemos utilizado el proporcionado por el proyecto 1000 Genomas (1000G), y para llevar a cabo esta tarea hemos utilizado dos programas, IMPUTE2 y MINIMAC3.

17.1.3. Resultados del estudio de asociación usando 1000G como panel de referencia

En este apartado ilustramos los log-p-valores resultantes de las test de asociación utilizando un modelo aditivo. Hemos elegido los tipos de

gráficos Manhattan y Cuantil-Cuantil con el fin de mostrar las ilustraciones más representativas de datos.

Previo a estas representaciones analizamos los resultados de imputación de los programas MINIMAC3 [124] y IMPUTE2 [121] con el fin de detectar si la utilización de un método de imputación u otro podría tener influencia en los resultados.

17.1.3.1. Gráficos tipo Manhattan y Cuantil-Cuantil a partir de los resultados de la imputación con MINIMAC3

Una vez realizada la imputación con MINIMAC3, los log-p-valores resultantes del test de asociación con el modelo aditivo (utilizando 1000G como panel de referencia) están representados en la figura 17.8.

Además, hemos ilustrado los log-p-valores observados en comparación con los esperados resultantes del test de asociación con el modelo aditivo obteniéndo así el gráfico Cuantil-Cuantil de la figura 17.9.

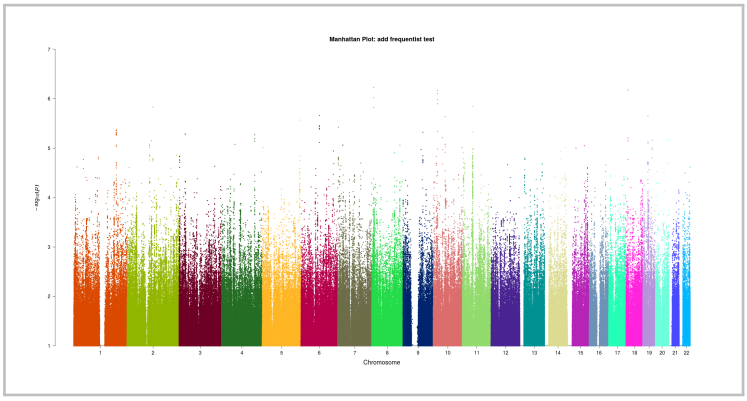


Figura 17.8: Gráfico Manhattan resultante del test de asociación con el modelo aditivo usando 1000 Genomas como panel de referencia y MINIMAC3 como software de imputacion.

17.1.3.2. Gráficos tipo Manhattan y Cuantil-Cuantil a partir de los resultados de la imputación con IMPUTE2

Una vez realizada la imputación con MINIMAC3, los log-p-valores resultantes del test de asociación con el modelo aditivo (utilizando 1000G como panel de referencia) se representan en la figura 17.10.

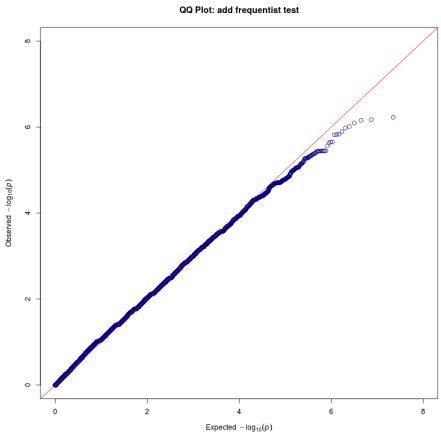


Figura 17.9: Gráfico Cuantil-Cuantil resultante del test de asociación con el modelo aditivo usando 1000G como panel de referencia y MINIMAC3 como software de imputacion.

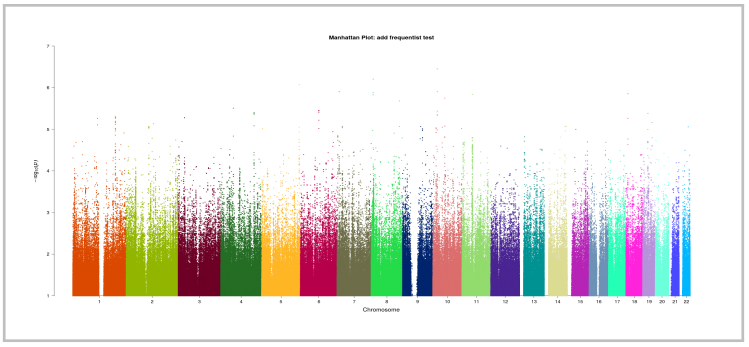


Figura 17.10: Gráfico Manhattan resultante del test de asociación con el modelo aditivo usando 1000 Genomas como panel de referencia y IMPUTE2 como software de imputacion.

Los log-p-valores observados y esperados resultantes del test de asociación con el modelo aditivo (usando 1000G como panel de referencia) también han sido representados en la figura 17.11.

Finalmente, podemos concluir que la cantidad total de valores significativos (eligiendolos como significativos aquellos marcadores que pre-

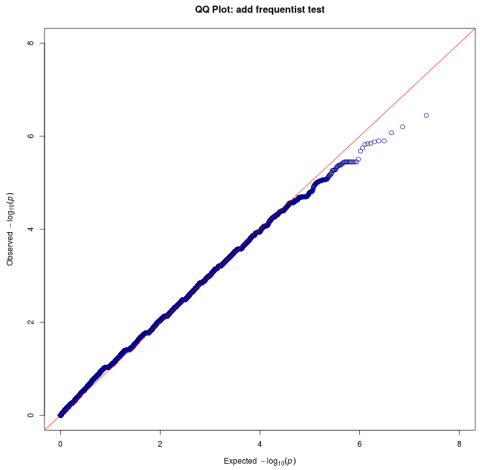


Figura 17.11: Gráfico Cuantil-Cuantil resultante del test de asociación con el modelo aditivo usando 1000G como panel de referencia e IMPUTE2 como software de imputacion.

sentan un p-valor por debajo del umbral de 10^{-5}) es mayor utilizando el software de imputación IMPUTE2 que con el programa MINIMAC3.

17.1.3.3. Selección de resultados significativos

El análisis de asociación correspondiente se realizó con diversos modelos de la posible relación entre el genotipo y la enfermedad (aditivo, dominante, general, recesivo y heterocigoto) y fue llevado a cabo utilizando todos los resultados de la imputación. Finalmente hemos analizado y representado los resultados del test aditivo con y sin el ajuste por la co-variable retinopatía"(es decir, si los sujetos estaban afectados o no por la enfermedad de retinopatía; hay estudios que apuntan a una posible base genética de la retinopatía diabética; dado que muchos de los sujetos del estudio estaban afectados por dicha enfermedad, ésta podía actuar como factor de confusión, llevándonos a detectar marcadores asociados con la retinopatía y no con la nefropatía). En la realización del filtrado y selección de los resultados más significativos, decidimos agrupar las variantes genómicas en regiones que contuvieran al menos 3 SNPs significativos (con p-valores inferiores a 10^{-5}). Por otra parte,

también sólo selecionamos aquellos SNPs con una medida de la calidad de imputación (*INFO*) mayor o igual 0.99. Los tres SNPs con los p-valores más significativos los localizamos en los cromosomas 8, 10 y 11, en las posiciones 9044765, 20304625, 49552399, respectivamente. Dichas variantes se denominan: rs4841106, rs2358658 y rs35649357, y sus representaciones corresponden a las figuras 17.12, 17.13 y 17.14.

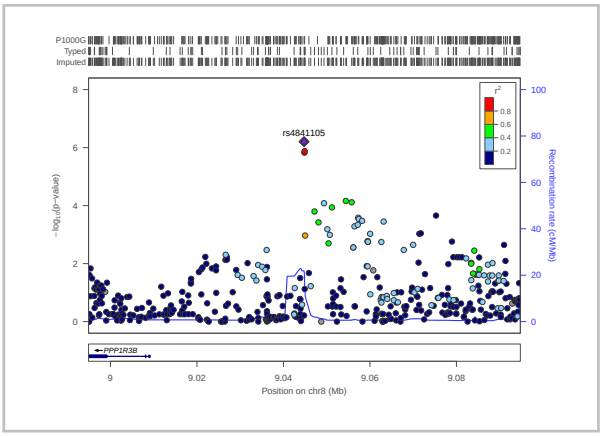


Figura 17.12: Primera región con SNPs genotipados e imputados, con el SNP rs4841106 en el centro.

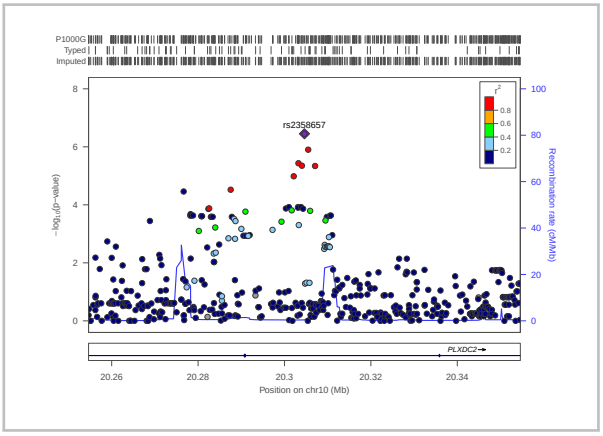


Figura 17.13: Segunda región con SNPs genotipados e imputados, con el SNP rs2358658 en el centro.

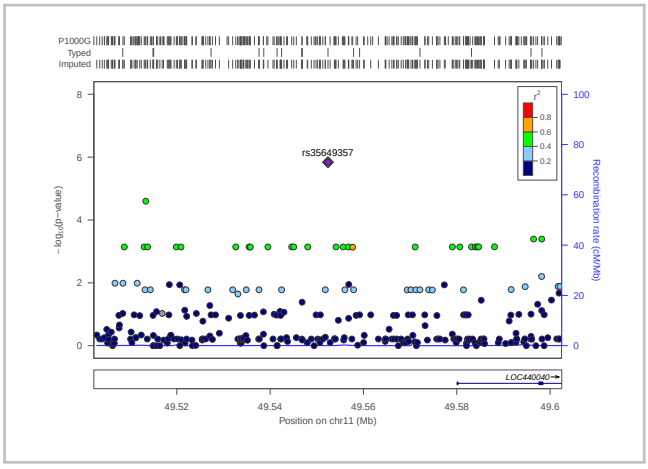


Figura 17.14: Tercera región con SNPs genotipados e imputados, con el SNP rs35649357 en el centro.

17.2. Resultados del análisis de bases de datos familiares

Hemos selleccionado como resultados del análisis de bases de datos familiares algunas aplicaciones prácticas de nuestro paquete alleHap, concretamente el análisis e identificación de factores predictivos asociados al inicio temprano e infantil de la diabetes tipo 1 (T1D) y la evaluación de la distribución de los haplotipos de riesgo entre sujetos canarios con respecto a los del resto de España y/o del mundo.

17.2.1. Diabetes en los padres y otros factores predictivos del inicio temprano de la diabetes tipo 1

17.2.1.1. Introducción

La diabetes tipo 1 es una enfermedad clínicamente heterogénea, provocada por varios factores ambientales desconocidos en individuos genéticamente susceptibles, cuya evolución puede consistir en una muy temprana y agresiva destrucción de las células beta, o en una progresión lenta a lo largo del tiempo, con los los pacientes requiriendo el uso de insulina de meses a años después del diagnóstico. En Europa, la tasa de incidencia de la diabetes de tipo 1 en la infancia aumenta globalmente en un 3-4 % por año, siendo el aumento del 6,3 % para los niños de 0 a

4 años, del 3,1 % para los niños de 5 a 9 años de edad, y un 2,4 % para los de 10 a 14 años [170]. Se espera que de 2005 a 2020 se duplque el número de niños que debutan con menos de 5 años y que el número de los que debutan antes de los 15 años aumente en un 70 % [171].

Hay grandes diferencias entre la diabetes T1D con inicio en la infancia y la T1D en la que el debut se produce cuando el individuo es adulto. El inicio de T1D en la infancia se asocia con cetosis/cetoacidosis más frecuentes, con una severa descompensación metabólica, con una mala función de las células beta residuales, fuerte autoinmunidad humoral contra las células de los islotes y la insulina, una mayor frecuencia de infecciones, una duración más corta de los síntomas y más independencia de los mecanismos de activación estacional, todo ello en mayor medida que en aquellos sujetos cuyo debut en T1D se produce a la edad adulta, lo que apunta a una forma más agresiva de la diabetes, [172], [173], [189], [174]. De acuerdo con algunos estudios [175], la mayor frecuencia de la enfermedad en hombres es otra característica aún no explicada de la diabetes tipo 1 en adultos los jóvenes.

La región HLA es el determinante genético más importante de la susceptibilidad a la diabetes tipo 1 [176]. De acuerdo con *Gillespie et al* [177], en el Reino Unido más de 90 % de los niños con diabetes tipo 1 portan haplotipos HLA de clase II: DRB1 * 03-DQB1 * 02: 01 (DR3-DQ2) y / o DRB1 * 04-DQB1 * 03: 02 (DR4-DQ8), y el diplotipo de mayor riesgo, DR3-DQ2 / DR4-DQ8, está presente en el 50 % de los casos de diabetes de inicio muy precoz.

Los autoanticuerpos asociados a la diabetes pueden ser utilizados como marcadores de T1D para sujetos jóvenes con mayor susceptibilidad genética a la enfermedad, y se ha encontrado asociación entre una edad precoz de aparición de T1D con la presencia de ciertos haplotipos HLA de alto riesgo, que se encuentran con mayor frecuencia en niños T1D diagnosticados antes de los 5 años de edad que en los diagnosticados cuando son mayores [178, 177].

Los estudios en parejas de gemelos sugieren que gran parte de la variabilidad de la edad de inicio T1D está determinada genéticamente [179]. La edad de inicio puede ser considerada como un indicador de la susceptibilidad genética, estando un inicio más temprano de la enfermedad relacionado con una componente genética más fuerte y por lo tanto con un mayor riesgo para los familiares en primer grado [177, 181]. Ahora bien, el incremento que se registra en la incidencia de T1D en la población joven no puede ser exclusivamente debido a cambios en el acervo genético de la población, sino que más bien sugiere una influencia

temprana del medio ambiente, por ejemplo modificaciones epigenéticas que se producen ya durante el periodo perinatal. Varios estudios han analizado la influencia de factores maternales en el riesgo de diabetes y han mostrado la existencia de una asociación entre la edad de la madre en el parto y un mayor riesgo que T1D se inicie en la infancia; asimismo, el orden de nacimiento ha mostrado también cierta asociación con una disminución significativa en el riesgo de la enfermedad (Sumnik et al. 2004, Cardwell et al. 2005, Haynes et al. 2007) (Bingley et al. 2000). El debut en la infancia (pero no en la edad adulta) del padre parece asociarse con la edad de debut de los hijos, mientras que solo las madres con una edad de debut anterior a los 10 años parece afectar a la edad de debut de sus hijos, pero no de sus hijas [183, 184]. Otros estudios indican que el hecho de que la madre debute con T1D antes o durante el embarazo no afecta al riesgo de diabetes del hijo de manera distinta a madres con edad de debut adulta [184].

Cabe destacar también que para el estudio de la genética y la patogénesis de la diabetes tipo 1, se ha desarrollado un importatne un esfuerzo internacional denominado T1DGC. Este proyecto ha sido constituido con miles de familias afectadas por la T1D, incluidas de todas partes del mundo. La colección de datos que se ha reunido representa un recurso extraordinario, no sólo por los datos genéticos, sino también por la información clínica asociada. La base de datos (de enero 2009) contiene información de 14494 sujetos en 3275 familias, de las cuales 2849 contienen al menos dos hermanos afectados con T1D y 426 contienen sólo un hijo afectado. Hay un total de 6271 hijos afectados y 1673 no afectados en esta base de datos. Entre los progenitores, 194 padres y 85 madres se están afectados por T1D. La edad de inicio está disponible para todos los hijos afectados y para algunos de los padres afectados (concretamente 130 padres y 67 madres).

Los sujetos en la base de datos T1DGC han sido reclutados en cuatro regiones: Asia-Pacífico (561 familias, 2289 sujetos), Europa (con exclusión del Reino Unido, 1221 familias, 5502 sujetos), América del Norte (1330 familias, 5967 sujetos) y Reino Unido (163 familias, 736 sujetos)

La base de datos contiene información de los alelos de varios marcadores en el complejo mayor de histocompatibilidad humano HLA, en particular HLA-A, HLA-B, HLA-CW, HLA-DPA, HLA-DPB, HLA-DQA, HLA-DQB, HLA-DRB, así como CTLA4 y el gen de la insulina-HPH SNPs. Los alelos en estos marcadores están completos para 12370 sujetos en la base de datos: 2215 padres y 2651 madres,

así como 6005 hijos afectados y 1499 hermanos no afectados (los 2124 sujetos restantes tienen los genotipos completamente perdidos, normalmente padres o hermanos que no aportaron muestras para el genotipado, aunque se reunió toda o parte de su información clínica).

Hemos utilizado nuestro paquete alleHap para identificar los haplotipos HLA de este conjunto de datos. Los haplotipos que comprenden los marcadores DRB-DQA-DQB se sabe que están relacionados con el riesgo de T1D. Entre los 12.370 individuos con genotipo completo en estos marcadores, fue posible obtener los haplotipos también completos en 11.095 (89,7 %). 493 (4 %) fueron parcialmente haplotipados y en 782 (6,3 %) no pudo hallarse ninguno de sus haplotipos de forma unívoca.

Entre los 2124 sujetos con genotipos completamente perdidos, 849 (40 %) pudieron haplotiparse completamente, 53 (2.5 %) fueron parcialmente haplotipados y en 1222 (57.5 %) no pudo obtenerse ningún haplotipo.

El objetivo de nuestro estudio fue estudiar los factores maternos asociados con el inicio precoz e infantil de la T1D, que pudieran utilizarse como predictores de esta forma de la enfermedad en la descendencia utilizando para ello el conjunto de datos disponible en T1DGC el 1 de octubre 2009.

17.2.1.2. Distribución del número de haplotipos de Alto Riesgo en la base de datos T1DGC

La tabla 17.1 muestra la distribución de frecuencias del número de haplotipos de alto riesgo DR3-DQ2 y DR4-DQ8, dependiendo de si los sujetos tienen la enfermedad, o no. Como puede verse, globalmente el 60 % de los sujetos afectados es protador de dos haplotipos de riesgo frente a sólo el 35,3 % en los no afectados. En el caso del Reino Unido la frecuencia de portadores de dos haplotipos de riesgo entre los afectados de T1D duplica a la frecuencia observada entre los no afectados. En todo caso, debe tenerse en cuenta, a la hora de interpretar estos datos, que la base de datos T1DGC no constituye una muestra aleatoria de las poblaciones estudiadas, sino una meustra compuesta por familias que tienen al menos dos hijos T1D. Por tanto los sujetos no afectados son padres y hermanos de sujetos afectados, con los que comparten su genética, por lo que es esperable una alta frecuencia de estos haplotipos incluso entre las personas no afectadas.

T1D	Número de Haplotipos de Riesgo 0	1	2
Datos globales:			
No	1544 (19.7 %)	3519 (45 %)	2761 (35.3 %)
Sí	620 (9.5 %)	1997 (30.5 %)	3933 (60 %)
Asia-Pacífico:			
No	311 (23.2 %)	517 (38.6 %)	511 (38.2 %)
Sí	115 (12.2 %)	281 (29.8 %)	546 (58 %)
Europa:			
No	544 (18.8 %)	1291 (44.7 %)	1051 (36.4 %)
Sí	233 (9 %)	794 (30.8 %)	1551 (60.2 %)
Norte América:			
No	654 (20.2 %)	1511 (46.7 %)	1068 (33 %)
Sí	254 (9.5 %)	841 (31.6 %)	1569 (58.9 %)
Reino Unido:			
No	35 (9.6 %)	200 (54.6 %)	131 (35.8 %)
Sí	18 (4.9 %)	81 (22.1 %)	267 (73 %)

Tabla 17.1: Distribución del número haplotipos de riesgo DR3-DQ2 y DR4-DQ8 para sujetos afecto y no afectos en la base de datos T1DGC. Se muestran datos regionales y globales.

Cuando se considera la edad de inicio de los sujetos, la tabla 17.2 muestra la distribución de frecuencias del número de haplotipos de altoriesgo DRB-DQA-DQB (en particular DR3-DQ2 y DQ8 DQ4) en sujetos de la base de datos T1DGC, a nivel mundial y por regiones. Puede observarse que más del 93 % de los sujetos con un inicio de la diabetes tipo 1 antes de la edad de 5 años tienen al menos un haplotipo de riesgo, y más de 61 % tienen dos haplotipos de riesgo. Por lo tanto, el número de haplotipos de alto riesgo se relaciona no sólo con la presencia de T1D, sino también con una edad más temprana de inicio de la enfermedad.

17.2.1.3. Factores maternos asociados con la aparición temprana y en la infancia de T1D

Para evaluar si hay algún tipo de efecto materno en la edad de aparición de T1D más allá de lo que puede explicarse por la presencia de los haplotipos de riesgo consideramos las siguientes variables:

Edad de debut	Número de Haplotipos de Riesgo 0	1	2
Datos globales:			
[0,5)	102 (7.1 %)	451 (31.4 %)	885 (61.5 %)
[5,10)	169 (9.3 %)	585 (32.2 %)	1065 (58.5 %)
[10,15)	180 (10.6 %)	552 (32.6 %)	963 (56.8 %)
15 o más	164 (10.8 %)	402 (26.5 %)	950 (62.7 %)
Sin T1D	1544 (19.7 %)	3519 (45 %)	2761 (35.3 %)
Asia-Pacifico:			
[0,5)	16 (7.8 %)	77 (37.4 %)	113 (54.9 %)
[5,10)	29 (11.2 %)	80 (31 %)	149 (57.8 %)
[10,15)	29 (11.4 %)	76 (29.9 %)	149 (58.7 %)
15 o más	41 (19.4 %)	48 (22.7 %)	122 (57.8 %)
Sin T1D	311 (23.2 %)	517 (38.6 %)	511 (38.2 %)
Europa:			
[0,5)	33 (7.4 %)	146 (32.7 %)	267 (59.9 %)
[5,10)	58 (8.8 %)	229 (34.8 %)	371 (56.4 %)
[10,15)	63 (9.9 %)	214 (33.8 %)	357 (56.3 %)
15 o más	77 (9.5 %)	201 (24.9 %)	529 (65.6 %)
Sin T1D	544 (18.8 %)	1291 (44.7 %)	1051 (36.4 %)
Norte América:			
[0,5)	49 (7.2 %)	207 (30.4 %)	424 (62.4 %)
[5,10)	76 (9.7 %)	253 (32.3 %)	455 (58 %)
[10,15)	82 (11.7 %)	236 (33.7 %)	382 (54.6 %)
15 o más	44 (9.4 %)	142 (30.3 %)	282 (60.3 %)
Sin T1D	654 (20.2 %)	1511 (46.7 %)	1068 (33 %)
Reino Unido:			
[0,5)	4 (3.8 %)	21 (19.8 %)	81 (76.4 %)
[5,10)	6 (5 %)	23 (19.3 %)	90 (75.6 %)
[10,15)	6 (5.6 %)	26 (24.3 %)	75 (70.1 %)
15 o más	2 (6.7 %)	11 (36.7 %)	17 (56.7 %)
Sin T1D	35 (9.6 %)	200 (54.6 %)	131 (35.8 %)

Tabla 17.2: Distribución del número haplotipos de riesgo DR3-DQ2 y DR4-DQ8 dependiendo de la edad de debut en la base de datos. Se muestran datos regionales y globales.

- Estado de la Madre (tiene T1D/ no tiene T1D).
- La edad de la madre en el momento del nacimiento del hijo.
- Para madres afectadas con T1D:
    - La edad de inicio de la madre.
    - Si el inicio de la diabetes materna se produce antes o después del nacimiento del niño afectado.
    - Número de años desde el diagnóstico de la T1D materna hasta el momento del nacimiento del niño afectado (si procede).

Con el fin de evitar efectos indeseados del tamaño de la familia (es decir, el sesgo en favor de factores presentes en las familias más grandes), se incluyeron en el análisis sólo los dos primeros hermanos afectados en cada familia (2849 familias). El género del sujeto, la positividad de anticuerpos, el número de enfermedades autoinmunes asociadas, AAID, el número de haplotipos HLA de riesgo, y los genotipos INS y CTLA4 fueron incluidos en el modelo como variables independientes y se analizaron en todas las familias.

17.2.1.4. Factores maternos, considerando todas las madres de la muestra

Considerando el modelo lineal:

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \varepsilon$$

dónde:

- La variable dependiente Y es la edad de inicio de la T1D (en los dos primeros hijos afectados de cada familia).
- Las variables independientes X_i son:
    - *BirthAgeMother*: la edad de la madre en el momento del nacimiento del niño.
    - *NRiskHaps*: número de haplotipos de riesgo DR3-DQ2 y DR4-DQ8 del individuo.
    - *R_gad65* y *r_ia2*: TAG y IA2 positividad de anticuerpos.
    - *Género*: Masculino o Femenino

- *Ins_hph1*: genotipo del gen de la insulina HPH1. El genotipo de referencia es AA y el modelo analiza los efectos de TA y TT en comparación con AA.
- *CTLA4*: genotipo CTLA4. El genotipo de referencia es AA, con respecto al cual se analizan los efectos de AG y GG.
- *AIDn*: número de enfermedades autoinmunes.
- *T1DM*: Variable indicadora de si la madre tiene T1D.
- *T1DF*: Variable indicadora de si el padre tiene T1D.

La estimación del modelo se muestra en la tabla 17.3. Se puede apreciar que las variables *ins_hph1, CTLA4* y *AIDn* no son significativas. El reajuste de este modelo sin estas variables produce el resultado se muestra en la tabla 17.4. La diferencia entre ambos modelos (diferencia en suma residual de cuadrados) no es significativa (p =0.2304).

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.0789	0.7601	31.68	0.0000
birthAgeMother	-0.2265	0.0195	-11.60	0.0000
nRiskHaps	-0.9039	0.1491	-6.06	0.0000
r_gad65	-3.3821	0.1992	-16.98	0.0000
r_ia2	-0.6928	0.1991	-3.48	0.0005
gender.female	-0.7993	0.1981	-4.03	0.0001
ins_hph1.TA	0.0840	0.2332	0.36	0.7186
ins_hph1.TT	0.9910	0.5824	1.70	0.0889
ctla4.AG	-0.3306	0.2194	-1.51	0.1319
ctla4.GG	-0.1993	0.2828	-0.70	0.4811
AIDn	0.3251	0.2557	1.27	0.2038
T1DM.Yes	-2.1153	0.6293	-3.36	0.0008
T1DF.Yes	-0.7408	0.3918	-1.89	0.0587

Tabla 17.3: Estimación del modelo lineal para la edad de debut del hijo/hija.

Como podemos ver, después de ajustar por el número de haplotipos de riesgo, la positividad de anticuerpos GAD y IA2 y el género del sujeto, el efecto de variables maternas (edad de la madre al nacer el hijo y presencia de diabetes tipo 1 en la madre) es aún apreciable. Incluso es perceptible un ligero efecto de la presencia de T1D en el padre. De hecho, la edad de inicio se reduce, como se esperaba, con el aumento en el

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	23.9721	0.7379	32.49	0.0000
birthAgeMother	-0.2272	0.0195	-11.66	0.0000
nRiskHaps	-0.9025	0.1488	-6.06	0.0000
r_gad65	-3.3876	0.1991	-17.02	0.0000
r_ia2	-0.6813	0.1987	-3.43	0.0006
gender.female	-0.7632	0.1961	-3.89	0.0001
T1DM.Yes	-2.1128	0.6292	-3.36	0.0008
T1DF.Yes	-0.7393	0.3917	-1.89	0.0592

Tabla 17.4: Estimación del modelo lineal para la edad de debut del hijo/hija excluyendo variables predictivas no significativas.

número de haplotipos de alto riesgo DR3-DQ2 y DR4-DQ8; menores edades de inicio también se asocian a la positividad GAD e IA2; y las niñas tienden a debutar antes que los niños. Teniendo en cuenta estas variables, la mayor edad de la madre en el parto se asocia con un inicio más temprano de la enfermedad en el hijo. Cuando la madre (y tal vez el padre) tienen diabetes tipo 1, también se detecta una tendencia a una reducción en la edad de debut en el hijo, lo que significa que probablemente hay otros factores genéticos implicados en la edad de inicio.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.0912	0.7290	33.05	0.0000
birthAgeMother	-0.2262	0.0192	-11.77	0.0000
nRiskHaps	-1.0420	0.1523	-6.84	0.0000
r_gad65	-3.3601	0.1968	-17.07	0.0000
r_ia2	-0.7213	0.1968	-3.67	0.0003
gender.female	-0.7982	0.1935	-4.12	0.0000
T1DM.Yes	-2.0788	0.6267	-3.32	0.0009
T1DF.Yes	-0.9063	0.3854	-2.35	0.0187
HLA.ACwB.A1-B8	0.6052	0.2392	2.53	0.0114
HLA.ACwB.A24-B39	-3.4473	0.9700	-3.55	0.0004

Tabla 17.5: Estimación del modelo lineal para la edad de debut del hijo/hija incluyendo haplotipos HLA A-Cw-B.

Nuestro paquete permite la exploración del efecto de otros haplotipos posibles en la edad del sujeto de inicio. Por ejemplo, al considerar haplotipos HLA clase I en los loci A-CW-B, algunos estudios indi-

can [185] que el haplotipo A1-B8 (HLA-A*0101-Cw*0701-B*0801) puede asociarse con DR3-DQ2 y modificar el riesgo asociado a este haplotipo. Hemos utilizado *alleHap* para explorar los haplotipos en esta región y hemos encontrado que A1-B8 es un haplotipo relativamente frecuente, presente en 1104 sujetos en la base de datos.

Asimismo también hemos observado que el haplotipo A24-B39 (HLA-A*2402-CW*0702-B*3906) parece estar asociado a una menor edad de inicio. Cuando estos haplotipos se incluyen en el modelo anterior se obtiene la estimación que se muestra en la tabla 17.5, en la que se aprecia un efecto significativo de ambos haplotipos.

17.2.1.5. Factores maternos considerando sólo las madres con T1D

Cuando sólo se consideran las madres con diabetes tipo 1, pueden introducirse en el modelo los efectos de la edad de inicio de T1D de la madre, o el tiempo transcurrido desde el diagnóstico de la madre hasta el momento del parto (años de evolución de la enfermedad). Como el número de madres con diabetes tipo 1 en la base de datos es bajo (n = 67) no se puede esperar gran resolución por parte del modelo. De hecho, si tenemos en cuenta la mismas variables que antes, llegamos a los resultados de la tabla 17.6, donde la única variable significativa resulta ser el número de haplotipos de riesgo.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	18.8559	4.2940	4.39	0.0000
birthAgeMother	-0.2264	0.1254	-1.81	0.0741
nRiskHaps	-3.8879	0.8426	-4.61	0.0000
r_gad65	0.9741	1.2350	0.79	0.4321
r_ia2	-0.3782	1.1688	-0.32	0.7469
sexfemale	0.1404	1.1648	0.12	0.9043
inshph1TA	2.0843	1.3626	1.53	0.1293
inshph1TT	1.2318	2.9629	0.42	0.6785
ctla4AG	-1.0317	1.3931	-0.74	0.4607
ctla4GG	-0.0495	1.7396	-0.03	0.9774
numEnfAuto	0.8142	1.6676	0.49	0.6265
T1DFYes	-1.9170	1.8754	-1.02	0.3092

Tabla 17.6: Estimación del modelo lineal para la edad de debut del hijo/hija usando sólo datos de familias en la que la madre tiene T1D.

Si reajustamos el modelo dejando sólo el número de haplotipos de riesgo y la edad de la madre en el parto (tabla 17.7) vemos que el efecto de esta variable sigue siendo significativo.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	19.7262	3.2209	6.12	0.0000
nRiskHaps	-3.8667	0.7033	-5.50	0.0000
birthAgeMother	-0.2342	0.1103	-2.12	0.0358

Tabla 17.7: Estimación del modelo lineal para la edad de debut del hijo/hija usando sólo datos familiares procedentes de familias en las que la madre tenda T1D, considerando sólo el número de haplotipos de riesgo en el sujeto y la edad de debut de la madre en la infancia.

Introduciendo ahora las variables *OnsetM* (que especifica la edad de inicio de T1D en la madre) y *motherEvolTime* (tiempo desde el diagnóstico de la diabetes tipo 1 en la madre hasta el nacimiento del niño) se obtienen los resultados mostrados en la tabla 17.8. Teniendo en cuenta el significado de estas variable, cabe esperar fuerte colinealidad entre ellas (al fin y al cabo, la edad de la madre en el parto es igual a su edad de debut más el número de años de evolución hasta el parto), lo que produce que no se detecte significación en ninguna.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	14.7636	3.8348	3.85	0.0002
nRiskHaps	-3.7556	0.7067	-5.31	0.0000
birthAgeMother	-0.3558	0.2180	-1.63	0.1054
onsetM	0.3191	0.2319	1.38	0.1715
motherEvolTime	0.2285	0.2580	0.89	0.3776

Tabla 17.8: Estimación del modelo lineal para la edad de debut del hijo/hija usando sólo datos de familias en las madre tenga T1D, considerando el número de haplotipos de riesgo en el sujetos, la edad de debut de la madre en la infancia, la edad de debut, y el tiempo de evolución de T1D (in years).

Debido a esta colinealidad, dado que la edad de la madre en el parto es una variable que ha resultado significativa en la muestra global, tendría sentido incluir en el modelo sólo la edad de debut de la madre o sólo el número de años de evolución con T1D. Tras probar ambos modelos,

el mejor ajuste se obtiene cuando se considera el tiempo de evolución hasta el momento del parto. Los resultados se muestran en la tabla 17.9. Como puede verse, el tiempo de evolución de la madre tiene un efecto negativo sobre la edad de aparición de sus hijos: cuanto mayor es el tiempo transcurrido desde el diagnóstico, más pronto se produce el debut en el hijo. En cualquier caso, este resultado debe tomarse con cautela debido a que la edad de la madre en el parto y el tiempo transcurrido desde el diagnóstico de diabetes tipo 1 en la madre son variables confundidas (si en el momento del parto lleva muchos años de evolución, esa madre tiene también una mayor edad). El efecto de ambas variables es difícil de separar con los datos disponibles, ya que la mustra de madres diabéticas T1D es pequeña.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	17.2737	3.3856	5.10	0.0000
nRiskHaps	-3.6260	0.7030	-5.16	0.0000
birthAgeMother	-0.1088	0.1242	-0.88	0.3825
motherEvolTime	-0.1179	0.0564	-2.09	0.0386

Tabla 17.9: Estimación del modelo lineal para la edad de debut del hijo/hija usando sólo datos de familias en las madre tenga T1D, considerando el número de haplotipos de riesgo en el sujetos, y el tiempo de evolución de T1D (in years) en la madre.

17.2.2. Comparación de la distribución de haplotipos de riesgo de T1D entre Canarias y el resto de España

Es bien sabido que en Canarias hay una alta prevalencia de diabetes tipo 1, mayor que en el resto de España. Podemos utilizar los haplotipos identificados por alleHap para comparar la distribución de los haplotipos de riesgo entre los sujetos afectados en ambos territorios.

La muestra española cuenta con un total de 597 sujetos genotipados en 149 familias. Muchas familias tienen el padre (60) o de la madre (38) sin genotipar. En este conjunto de datos hay 181 individuos procedentes de las Islas Canarias, agrupados en 42 familias.

Asimismo hay en total en la muestra 226 sujetos afectados con T1D en la España peninsular y 86 en las islas. La tabla 17.10 muestra la distribución de haplotipos de riesgo (DR3-DQ8 y DR4-DQ2) y protección (DR2, DR6, DR7 y DR11) en los marcadores DRB, DQA, DQB.

Haplotipos	Islas Canarias	España peninsular
DR2-6-7-11/DR2-6-7-11	16 (9.1 %)	11 (3.1 %)
DR2-6-7-11/DR3-DQ2	25 (14.2 %)	51 (14.2 %)
DR2-6-7-11/DR4-DQ8	26 (14.8 %)	40 (11.1 %)
DR2-6-7-11/other	16 (9.1 %)	20 (5.6 %)
DR3-DQ2/DR3-DQ2	8 (4.5 %)	36 (10 %)
DR3-DQ2/DR4-DQ8	42 (23.9 %)	85 (23.7 %)
DR3-DQ2/other	11 (6.2 %)	39 (10.9 %)
DR4-DQ8/DR4-DQ8	12 (6.8 %)	18 (5 %)
DR4-DQ8/other	16 (9.1 %)	38 (10.6 %)
other/other	4 (2.3 %)	21 (5.8 %)

Tabla 17.10: Frecuencias de los haplotipos DRB-DQA-DQB de las islas Canarias con respecto a los del resto de España.

Podemos ver que no hay grandes diferencias entre los dos territorios; la diferencia más grande está en el diplotipo DR3-DQ2 / DR3-DQ2. En cualquier caso para evaluar la significación de las diferencias observadas no es posible aplicar una prueba de chi-cuadrado estándar, dado que los datos no son independientes (en muchos casos los sujetos con los mismos haplotipos pertenecen a la misma famila). En su lugar utilizaremos un procedimiento bootstrap consistente en simular 100000 veces la selección aleatoria de 42 familias del total de 149 familias de España.

Por cada grupo de 42 familias seleccionadas al azar se calcula y se guarda el estadístico chi-cuadrado. Los 100000 valores obtenidos de esta manera nos dan la distribución bootstrap de esta estadístico, a partir de la cual se puede calcular el p-valor de la prueba para la muestra de 42 familias canarias. En concreto, para la comparación de la distribución de haplotipos DRB-DQB-DQA entre Canarias y la España peninsular, el valor resultante de la prueba de Chi-cuadrado fue 23.124.

El p-valor bootstrap correspondiente resulta 0.097, por lo que las diferencias no son significativas. En cualquier caso, el tamaño de la muestra es muy reducido (sólo 42 familias en las islas Canarias), por lo que el resultado no es concluyente.

17.2.3. Comparación de la distribución de haplotipos de riesgo de T1D entre España y el resto de Europa

De la misma manera podemos comparar la distribución de los haplotipos de riesgo entre España y el resto de Europa. Para los datos europeo existen 4091 sujetos completamente genotipados, distribuidos en 1137 familias, con 2269 sujetos afectados con diabetes tipo 1; en estas familias hay 439 padres y 259 madres con los genotipos completamente perdidos. Después de aplicar alleHap, la distribución de los haplotipos identificados se muestra en la tabla 17.11.

Tampoco en este caso se observan grandes diferencias. Procediendo de la misma manera que antes, el estadístico chi-cuadrado para esta tabla da un valor de 21.462, con un p-valor bootstrap de 0.1729. Por tanto, tampoco se detectan diferencias significativas en la distribución de los haplotipos de riesgo HLA entre España y el resto de Europa.

Haplotipos	Europeos	Españoles
DR2-6-7-11/DR2-6-7-11	188 (4.9 %)	25 (5.1 %)
DR2-6-7-11/DR3-DQ2	452 (11.9 %)	67 (13.6 %)
DR2-6-7-11/DR4-DQ8	641 (16.9 %)	62 (12.6 %)
DR2-6-7-11/other	337 (8.9 %)	35 (7.1 %)
DR3-DQ2/DR3-DQ2	193 (5.1 %)	37 (7.5 %)
DR3-DQ2/DR4-DQ8	758 (19.9 %)	119 (24.1 %)
DR3-DQ2/other	298 (7.8 %)	48 (9.7 %)
DR4-DQ8/DR4-DQ8	273 (7.2 %)	30 (6.1 %)
DR4-DQ8/other	476 (12.5 %)	51 (10.3 %)
other/other	186 (4.9 %)	19 (3.9 %)

Tabla 17.11: Frecuencias de los haplotipos DRB-DQA-DQB en España con respecto a los del resto de países Europeos.

Capítulo 18

Conclusiones

Como principales conclusiones de esta tesis citamos las siguientes:

I. Se ha realizado una revisión muy completa de los métodos estadísticos en Genética y Bioinformática utilizados en la actualidad para evaluar la asociación entre el genoma y las enfermedades.

II. Se ha elaborado un tutorial con instrucciones detalladas para la realización estudios de asociación del genoma (GWAS), incluyendo procedimientos para Control de Calidad, Identificación de la fase de genotipos, Alineamiento, Imputación, y Análisis de Asociación estadístico.

III. Se han identificado las variantes genéticas con mayor significación asociadas a la nefropatía diabética avanzada en la diabetes tipo 2 (T2D) en la población de la isla de Gran Canaria. Los SNPs o regiones más significativas hallados son rs4841106, rs2358658, and rs35649357, encontradas en las posiciones 9044856, 20307083, 49552399 de los cromosomas 8, 10 y 11 respectivamente. De los SNPs anteriores sólo uno, el del cromosoma 8, se relaciona con un gen conocido, el PLXDC2 (denominado también *Plexin Domain Containing 2*).

IV. Se ha desarrollado un paquete en lenguaje R llamado alleHap, capaz de imputar alelos e identificar (reconstruir) haplotipos de manera determinista e inequívoca en bases de datos familiares mediante el cruce de la información genotípica (no recombinante) entre padres e hijos. El paquete se ha subido al repositorio oficial

de R (CRAN), donde está disponible pública y gratuitamente para que pueda utilizarlo cualquier investigador del planeta.

V. Se ha evaluado el rendimiento de las funciones del paquete alleHap, obteniendo las siguientes conclusiones específicas:

   a) En relación con el tiempo de procesamiento, tiempo de cálculo crece linealmente con el número de las familias y el número de marcadores.

   b) Con respecto a la tasa de imputación (teniendo en cuenta una situación típica de 4 alelos por marcador y de 5 a 50 marcadores):

      1) Para una situación extrema con padres totalmente perdidos, las tasas de imputación oscilan entre aproximadamente el 5-6 % cuando sólo un hijo está disponible, teniendo un alto porcentaje de valores perdidos, con casi un 55 % cuando hay disponibles tres hijos (incluso con una tasa de pérdida del 25 % de los alelos en hijos).
      2) Para la situación extrema de que un hijo tuviera el genotipo completamente perdido (que no estuviera genotipado), los rangos de tasas de imputación oscilan entre un 5-6 % cuando los padres tienen un 75 % de alelos perdidos hasta casi un 60 % cuando los padres no tienen valores perdidos.
      3) En situaciones intermedias con alelos perdidos tanto en padres como en hijos, se llegan a alcanzar tasas de imputación de hasta un 98 % cuando hay por lo menos tres hijos disponibles y la tasa de pérdida de alelos no es demasiado baja. Para una tasa de pérdida de alelos de un 50 % en padres e hijos, la tasa de imputación oscila entre el 21 % cuando sólo hay un hijo al 55 % cuando hay tres o más hijos.

   c) Con respecto a la tasas de haplotipado (teniendo en cuenta de nuevo una situación típica de 4 alelos por marcador y de 5 a 50 marcadores) resultó que:

      1) Cuando no hay alelos perdidos ni en los padres ni en los hijos, la tasa de identificación de haplotipos completos oscila entre el 80 % cuando sólo hay un hijo, y el 100 % en los casos cuando hay tres o más hijos disponibles.

2) Para una situación extrema en que los hijos tengan un 75 % de los alelos perdidos (si todos los alelos en los hijos están perdidos, la identificación de haplotipos no es posible), la tasa de identificación de haplotipos completos oscila entre 2 % (cuando los padres tienen un 50 % de alelos perdidos) a un 35 % (cuando los padres no tienen alelos perdidos).
3) Para una situación extrema en que los padres tienen un 75 % de los alelos perdidos la tasa de identificación de haplotipos completos oscila desde el 2-30 % (cuando sólo hay un hijo con de 0 a 50 % de los alelos perdidos) a entre un 6-85 % cuando hay por lo menos tres hijos, con tasas de alelos perdidos que oscilan entre 0 y 50 %.
4) Para situaciones intermedias con un 25 % de alelos perdidos en padres e hijos, la tasa de identificación de haplotipos completos oscila entre 22 % cuando sólo hay un hijo disponible al 80 % en los casos de tres o más hijos disponibles.

VI. Se ha elaborado un completo manual de usuario del paquete alleHap, el cual también está publicamente disponible para la comunidad de usuarios de R a través de la web de CRAN.

VII. Se han identificado los haplotipos de riesgo en algunos marcadores HLA incluidos en la base de datos T1DGC. Los resultados confirman que el 90 % de los sujetos afectados portan al menos uno de los haplotipos de riesgo DR3-DQ2 y DQ8-DR4. Esta proporción es ligeramente más alta en la muestra del Reino Unido (95 %).

VIII. Los resultados también confirman que, a nivel mundial, el 61,5 % de los casos con inicio precoz de diabetes (debut antes de la edad de 5 años) llevan dos haplotipos de riesgo. Esta proporción es de nuevo más alta en los sujetos del Reino Unido (76,4 %).

IX. Se ha realizado un análisis de factores maternos asociados con el desarrollo precoz y en la infancia de la diabetes tipo 1 (T1D). Los resultados muestran que después de ajustar por la presencia de haplotipos de alto riesgo:

a) La edad de la madre en el parto se asocia negativamente con la edad de inicio de los hijos (cuanto mayor es la madre en el parto, más joven debuta el hijo).

b) La edad de inicio de los hijos es menor en promedio para las madres afectadas por T1D.

c) Para las madres afectadas por T1D, se detecta cierta asociación negativa entre el tiempo de evolución de la enfermedad y la edad de inicio de la enfermedad en los hijos diabéticos (aunque hay que tener cautela con este resultado debido a los posibles factores de confusión).

X. Al comparar la frecuencia de los haplotipos de riesgo en la muestra de las Islas Canarias frente a muestras de la España peninsular, no se han detectado diferencias significativas. Lo mismo ocurre al comparar la muestra española con el resto de la muestra europea. En cualquier caso, los resultados no son concluyentes debido al bajo número de familias canarias en la muestra.

Bibliografía

[1] Sharon R Browning and Brian L Browning. Haplotype phasing: existing methods and new developments. *Nature Reviews Genetics*, 12(10):703–714, 2011.

[2] Brian L Browning and Sharon R Browning. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *The American Journal of Human Genetics*, 84(2):210–223, 2009.

[3] S. J. Mack et al. Common and well-documented hla alleles: 2012 update to the cwd catalogue. *Tissue Antigens*, 81(4):194–203, 2013.

[4] Paul IW de Bakker et al. A high-resolution hla and snp haplotype map for disease association studies in the extended human mhc. *Nature genetics*, 38(10):1166–1172, 2006.

[5] EC Castelli, CT Mendes-Junior, LC Veiga-Castelli, NF Pereira, ML Petzl-Erler, and EA Donadi. Evaluation of computational methods for the reconstruction of hla haplotypes. *Tissue Antigens*, 76(6):459–466, 2010.

[6] Virtual Medical Centre. *Genetic DNA*. © Virtual Medical Centre, 2010.

[7] Harvey F Lodish, Arnold Berk, S Lawrence Zipursky, Paul Matsudaira, David Baltimore, James Darnell, et al. *Molecular cell biology*, volume 4. WH Freeman New York, 4 edition, 2007.

[8] Neil A. Campbell, Brad Williamson, and Robin J. Heyden. *Biology: Exploring Life*. Pearson Prentice Hall, Boston, Massachusetts, 2006.

[9] Duncan C. Thomas. *Statistical methods in genetic epidemiology*. Oxford University Press, 2004.

[10] RERF. Japan-US Research Foundation. *Characteristics of chromosome groups: Karyotyping*. © Radiation Effects Research Foundation, 2007.

[11] Departament of Biology. *Chromatid Definition*. © Biology Online, 2008.

[12] PBworks Online Team Collaboration. *Online Computational Biology Textbook*. © Radiation Effects Research Foundation, 2007.

[13] Scitable. *DNA Is a Structure That Encodes Biological Information*. © Nature Education, 2009.

[14] Wikibooks Principles of Biochemistry. *Nucleic acid I: DNA and its nucleotides*. Wikibooks, 2011.

[15] Genetics Home Reference. *Base pair*. Lister Hill National Center for Biomedical Communications- U.S. National Library of Medicine, 2007.

[16] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walters. *Molecular Biology of the Cell*. New York and London: Garland Science, fourth edition, 2002.

[17] Departament of Biology. *DNA Structure*. © Penn State University, 2004.

[18] International Human Genome Sequencing Consortium et al. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004.

[19] Howard Gest. Evolution of knowledge encapsulated in scientific definitions. *Perspectives in biology and medicine*, 44(4):556–564, 2001.

[20] Indira Rajagopal. *Genome Organization*. Oregon State University, 2009.

[21] Suzanne Clancy and William Brown. *Translation: DNA to mRNA to protein*, volume 1. Nature Education, 2008.

[22] Robert C Elston, Jaya M Satagopan, and Shuying Sun. Genetic terminology. In *Statistical Human Genetics*, pages 1–9. Springer, 2012.

[23] Anne Cronin and Mary Beth Mandich. *Human development and performance throughout the lifespan*. Cengage Learning, 2015.

[24] Bruce Alberts, Dennis Bray, Karen Hopkin, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Essential cell biology*. Garland Science, 2013.

[25] Talking Glossary of Genetic Terms. *Locus*. National Human Genome Research Institute, 2009.

[26] N Malats and F Calafell. Basic glossary on genetic epidemiology. *Journal of epidemiology and community health*, 57(7):480–482, 2003.

[27] E.P. Muljadi. *Human genetics*. © Google Books, 2012.

[28] Daniel L. Hartl. *Essential genetics: A genomics perspective*. Jones & Bartlett Publishers, 5 edition, 2014.

[29] Wikipedia. *Allele - Dominant and recessive alleles*. © Wikipedia, 2013.

[30] GCSE Bitesize. *Recessive and dominant alleles*. © British Broadcasting Corporation - BBC, 2007.

[31] Scitable. *Haplotype / Haplotypes*. © Nature Education, 2009.

[32] Jeffrey Mahr. *Operating a Sex Machine - Meiosis*. OpenStax CNX, © Rice University, 2015.

[33] Lynn B. Jorde. *Genetic Variation and Human Evolution*. The American Society of Human Genetics, Inc., 2003.

[34] Monya Baker. Structural variation: the genome's hidden architecture. *Nature methods*, 9(2):133–137, 2012.

[35] RGH Cotton and CR Scriver. Proof of "disease causing" mutation. *Human mutation*, 12(1):1–3, 1998.

[36] Richard Twyman. *A variable genome*. Wellcome Trust, 2003.

[37] Jocelyn E Krebs, Benjamin Lewin, Elliott S Goldstein, and Stephen T Kilpatrick. *Lewin's essential genes*. Jones & Bartlett Publishers, 2013.

[38] Sirius Genomics. *What is a Single Nucleotide Polymorphism?* Sirius Genomics Inc., 2013.

[39] Genetics Home Reference. *What are single nucleotide polymorphisms (SNPs)?* Lister Hill National Center for Biomedical Communications- U.S. National Library of Medicine, 2007.

[40] Dustin J. Penn. *Major Histocompatibility Complex (MHC)*. Macmillan Publishers Ltd., 2002.

[41] SCL Gough and MJ Simmonds. The hla region and autoimmune disease: associations and mechanisms of action. *Current genomics*, 8(7):453, 2007.

[42] Roger Horton, Laurens Wilming, Vikki Rand, Ruth C Lovering, Elspeth A Bruford, Varsha K Khodiyar, et al. Gene map of the extended human mhc. *Nature Reviews Genetics*, 5(12):889–899, 2004.

[43] AJ Mungall, SA Palmer, SK Sims, CA Edwards, JL Ashurst, L Wilming, MC Jones, R Horton, SE Hunt, CE Scott, et al. The dna sequence and analysis of human chromosome 6. *Nature*, 425(6960):805–811, 2003.

[44] HLA Complex. *HLA Complex*. Scisco Genetics Inc., 2013.

[45] WHO Committee. *Nomenclature for Factors of the HLA System*. © Anthony Nolan Research Institute, 2010.

[46] Bhadran Bose, David W Johnson, and Scott B Campbell. *Transplantation Antigens and Histocompatibility Matching*. INTECH Open Access Publisher, 2013.

[47] Shizhong Xu. *Principles of statistical genomics*. Springer, 2013.

[48] Nicholas J Schork, Tiffany A Greenwood, and David L Braff. Statistical genetics concepts and approaches in schizophrenia and related neuropsychiatric research. *Schizophrenia bulletin*, 33(1):95–104, 2007.

[49] W Maxwell Cowan, Kathy L Kopnisky, and Steven E Hyman. The human genome project and its impact on psychiatry. *Annual Review of Neuroscience*, 25(1): 1–50, 2002.

[50] Shili Lin and Hongyu Zhao. *Handbook on Analyzing Human Genetic Data*. Springer, 2010.

[51] William S Bush and Jason H Moore. Chapter 11: Genome-wide association studies. *PLoS Comput Biol*, 8(12):e1002822, 2012.

[52] Lucia A Hindorff, Praveen Sethupathy, Heather A Junkins, Erin M Ramos, Jayashri P Mehta, Francis S Collins, and Teri A Manolio. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proceedings of the National Academy of Sciences*, 106(23):9362–9367, 2009.

[53] Frank Yates. Contingency tables involving small numbers and the χ 2 test. *Supplement to the Journal of the Royal Statistical Society*, pages 217–235, 1934.

[54] Curt Stern. The hardy-weinberg law. *Science*, 97(2510):137–138, 1943.

[55] Andrea S. Foulkes. *Applied statistical genetics with R: for population-based association studies*. Springer Science & Business Media, 2009.

[56] Stephen Turner, Loren L Armstrong, Yuki Bradford, Christopher S Carlson, Dana C Crawford, Andrew T Crenshaw, Mariza Andrade, Kimberly F Doheny, Jonathan L Haines, Geoffrey Hayes, et al. Quality control procedures for genome-wide association studies. *Current protocols in human genetics*, pages 1–19, 2011.

[57] C Andrews. The hardy-weinberg principle. *Nature Education Knowledge*, 1(8):65, 2010.

[58] Anuj Gupta. Classification of complex uci datasets using machine learning and evolutionary algorithms. *IJSTR*, 4(5):85–94, 2015.

[59] David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.

[60] Joel N Hirschhorn and Mark J Daly. Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2):95–108, 2005.

[61] David J Balding, Martin Bishop, and Chris Cannings. *Handbook of statistical genetics*, volume 1. John Wiley & Sons, 2008.

[62] Lon R Cardon and Lyle J Palmer. Population stratification and spurious allelic association. *The Lancet*, 361(9357):598–604, 2003.

[63] Richard S Spielman, Ralph E McGinnis, and Warren J Ewens. Transmission test for linkage disequilibrium: the insulin gene region and insulin-dependent diabetes mellitus (iddm). *American journal of human genetics*, 52(3):506, 1993.

[64] Richard S Spielman and Warren J Ewens. A sibship test for linkage in the presence of association: the sib transmission/disequilibrium test. *The American Journal of Human Genetics*, 62(2):450–458, 1998.

[65] D Curtis. Use of siblings as controls in case-control association studies. *Annals of Human genetics*, 61(4):319–333, 1997.

[66] Michael Boehnke and Carl D Langefeld. Genetic association mapping based on discordant sib pairs: the discordant-alleles test. *The American Journal of Human Genetics*, 62(4):950–961, 1998.

[67] Steve Horvath and Nan M Laird. A discordant-sibship test for disequilibrium and linkage: no need for parental data. *The American Journal of Human Genetics*, 63(6):1886–1897, 1998.

[68] Eden R Martin, Stephanie A Monks, Liling L Warren, and Norman L Kaplan. A test for linkage and association in general pedigrees: the pedigree disequilibrium test. *The American Journal of Human Genetics*, 67(1):146–154, 2000.

[69] SA Monks and NL Kaplan. Removing the sampling restrictions from family-based tests of association for a quantitative-trait locus. *The American Journal of Human Genetics*, 66(2):576–592, 2000.

[70] Frank Dudbridge. Pedigree disequilibrium tests for multilocus haplotypes. *Genetic epidemiology*, 25(2):115–121, 2003.

[71] ER Martin, MP Bass, JR Gilbert, MA Pericak-Vance, and ER Hauser. Genotype-based association test for general pedigrees: The genotype-pdt. *Genetic epidemiology*, 25(3):203–213, 2003.

[72] Stephen L Lake, Deborah Blacker, and Nan M Laird. Family-based tests of association in the presence of linkage. *The American Journal of Human Genetics*, 67(6): 1515–1525, 2000.

[73] Eugene V Koonin. Computational genomics. *Current Biology*, 11(5):R155–R158, 2001.

[74] BioEECS. *Computational Genomics and Proteomics*. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2012.

[75] Robert Gentleman, Vincent Carey, Wolfgang Huber, Rafael Irizarry, and Sandrine Dudoit. *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer Science & Business Media, 2006.

[76] Adi L Tarca, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. Machine learning and its applications to biology. *PLoS Comput Biol*, 3(6): e116, 06 2007.

[77] Johannes Fürnkranz, Dragan Gamberger, et al. *Foundations of rule learning*. Springer Science & Business Media, 2012.

[78] MLSB14. *Workshop on Machine Learning for Systems Biology*. European Conference on Computational Biology, 2014.

[79] Sašo Džeroski, Simon Rogers, and Guido Sanguinetti. Machine learning in systems biology. In *Proceedings of The Fourth International Workshop*, 2010.

[80] Pedro Larrañaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Iñaki Inza, José A Lozano, Rubén Armañanzas, Guzmán Santafé, Aritz Pérez, et al. Machine learning in bioinformatics. *Briefings in bioinformatics*, 7(1):86–112, 2006.

[81] Jason Weston, Christina Leslie, Eugene Ie, Dengyong Zhou, Andre Elisseeff, and William Stafford Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

[82] Pierre Baldi, Yves Chauvin, Tim Hunkapiller, and Marcella A McClure. Hidden markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences*, 91(3):1059–1063, 1994.

[83] Lawrence Rabiner. *First Hand: The Hidden Markov Model*. IEEE Global History Network, 2012.

[84] Michael Nothnagel. *Genotype Imputation*. University of Kiel, 2010.

[85] Jonathan Marchini and Bryan Howie. Genotype imputation for genome-wide association studies. *Nature Reviews Genetics*, 11(7):499–511, 2010.

[86] Ion Mandoiu and Alexander Zelikovsky. *Bioinformatics algorithms: techniques and applications*, volume 3. John Wiley & Sons, 2008.

[87] Jeff A Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[88] Juan Manuel Górriz, Elmar W Lang, and Javier Ramírez. *Recent advances in biomedical signal processing*. Bentham Science Publishers, 2011.

[89] Longbing Cao, Yong Feng, and Jiang Zhong. *Advanced Data Mining and Applications: 6th International Conference, ADMA 2010, Chongqing, China, November 19-21, 2010, Proceedings*, volume 6440. Springer, 2010.

[90] Todd K Moon. The expectation-maximization algorithm. *Signal processing magazine, IEEE*, 13(6):47–60, 1996.

[91] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[92] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[93] Laurent Excoffier and Montgomery Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular biology and evolution*, 12(5):921–927, 1995.

[94] Montgomery Slatkin and Laurent Excoffier. Testing for linkage disequilibrium in genotypic data using the expectation-maximization algorithm. *Heredity*, 76: 377–383, 1996.

[95] Richard A Gibbs, John W Belmont, Paul Hardenbol, Thomas D Willis, Fuli Yu, Huanming Yang, Lan-Yang Ch'ang, Wei Huang, Bin Liu, Yan Shen, et al. The international hapmap project. *Nature*, 426(6968):789–796, 2003.

[96] Gudmundur A Thorisson, Albert V Smith, Lalitha Krishnan, and Lincoln D Stein. The international hapmap project web site. *Genome research*, 15(11):1592–1593, 2005.

[97] International HapMap Consortium et al. A haplotype map of the human genome. *Nature*, 437(7063):1299–1320, 2005.

[98] 1000 Genomes Project Consortium et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

[99] 1000 Genomes Project Consortium. *1000 Genomes - A Deep Catalog of Human Genetic Variation*. © 1000 Genomes, 2012.

[100] Peter H Sudmant, Tobias Rausch, Eugene J Gardner, Robert E Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, Markus Hsi-Yang Fritz, et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81, 2015.

[101] Stephen S Rich, Patrick Concannon, Henry Erlich, Cecile Julier, Grant Morahan, Jorn Nerup, Flemming Pociot, and John A Todd. The type 1 diabetes genetics consortium. *Annals of the New York Academy of Sciences*, 1079(1):1–8, 2006.

[102] NIDDK Central Repository. *Type 1 Diabetes Genetics Consortium*. © The National Institute of Diabetes and Digestive and Kidney Diseases, 2010.

[103] Josyf C Mychaleckyj, Janelle A Noble, Priscilla V Moonsamy, Joyce A Carlson, Michael D Varney, Jeff Post, Wolfgang Helmberg, June J Pierce, Persia Bonella, Anna Lisa Fear, et al. Hla genotyping in the international type 1 diabetes genetics consortium. *Clinical Trials*, 7(1 suppl):S75–S87, 2010.

[104] Technical Note: DNA Analysis. *Imputation Estimates Genotypes at Un-Genotyped Loci*. © Illumina, Inc., 2013.

[105] Bryan Howie, Christian Fuchsberger, Matthew Stephens, Jonathan Marchini, and Gonçalo R Abecasis. Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. *Nature genetics*, 44(8):955–959, 2012.

[106] Sebastian Zöllner and Jonathan K Pritchard. Coalescent-based association mapping and fine mapping of complex trait loci. *Genetics*, 169(2):1071–1092, 2005.

[107] Mark J Minichiello and Richard Durbin. Mapping trait loci by use of inferred ancestral recombination graphs. *The American Journal of Human Genetics*, 79(5): 910–922, 2006.

[108] Brian L Browning and Sharon R Browning. Efficient multilocus association testing for whole genome association studies using localized haplotype clustering. *Genetic epidemiology*, 31(5):365–375, 2007.

[109] Zhan Su, Niall Cardin, Wellcome Trust Case Control Consortium, Peter Donnelly, Jonathan Marchini, et al. A bayesian method for detecting and characterizing allelic heterogeneity and boosting signals in genome-wide association studies. *Statistical Science*, pages 430–450, 2009.

[110] Stephen Leslie, Peter Donnelly, and Gil McVean. A statistical method for predicting classical hla alleles from snp data. *The American Journal of Human Genetics*, 82(1):48–56, 2008.

[111] Chris C Spencer, Zhan Su, Peter Donnelly, and Jonathan Marchini. Designing genome-wide association studies: sample size, power, imputation, and the choice of genotyping chip. *PLoS Genet*, 5(5):e1000477, 2009.

[112] Jonathan Marchini, Bryan Howie, Simon Myers, Gil McVean, and Peter Donnelly. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature genetics*, 39(7):906–913, 2007.

[113] Bertrand Servin and Matthew Stephens. Imputation-based analysis of association studies: candidate regions and quantitative traits. *PLoS Genetics*, 3(7):e114, 2007.

[114] Yun Li, Cristen Willer, Serena Sanna, and Gonçalo Abecasis. Genotype imputation. *Annual review of genomics and human genetics*, 10:387, 2009.

[115] Paul IW de Bakker, Manuel AR Ferreira, Xiaoming Jia, Benjamin M Neale, Soumya Raychaudhuri, and Benjamin F Voight. Practical aspects of imputation-driven meta-analysis of genome-wide association studies. *Human molecular genetics*, 17(R2):R122–R128, 2008.

[116] Eleftheria Zeggini and Andrew Morris. *Analysis of complex disease association studies: a practical guide*. Academic Press, 2010.

[117] Matthew Stephens and Peter Donnelly. Inference in molecular population genetics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4): 605–635, 2000.

[118] Paul Fearnhead and Peter Donnelly. Estimating recombination rates from population genetic data. *Genetics*, 159(3):1299–1318, 2001.

[119] Na Li and Matthew Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165 (4):2213–2233, 2003.

[120] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[121] Bryan N Howie, Peter Donnelly, and Jonathan Marchini. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genet*, 5(6):e1000529, 2009.

[122] Yun Li, Cristen J Willer, Jun Ding, Paul Scheet, and Gonçalo R Abecasis. Mach: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic epidemiology*, 34(8):816–834, 2010.

[123] Christian Fuchsberger, Gonçalo R Abecasis, and David A Hinds. minimac2: faster genotype imputation. *Bioinformatics*, 31(5):782–784, 2015.

[124] Sayantan Das. *Minimac3*. Center for Statistical Genetics, University of Michigan, 2015.

[125] Sharon R Browning. Multilocus association mapping using variable-length markov chains. *The American Journal of Human Genetics*, 78(6):903–913, 2006.

[126] Paul Scheet and Matthew Stephens. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *The American Journal of Human Genetics*, 78(4):629–644, 2006.

[127] Yongtao Guan and Matthew Stephens. Practical issues in imputation-based association mapping. *PLoS Genet*, 4(12):e1000279, 2008.

[128] Gillian CL Johnson, Laura Esposito, Bryan J Barratt, Annabel N Smith, Joanne Heward, Gianfranco Di Genova, Hironori Ueda, Heather J Cordell, Iain A Eaves, Frank Dudbridge, et al. Haplotype tagging for the identification of common disease genes. *Nature genetics*, 29(2):233–237, 2001.

[129] David M Evans, Lon R Cardon, and Andrew P Morris. Genotype prediction using a dense map of snps. *Genetic epidemiology*, 27(4):375–384, 2004.

[130] Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.

[131] DY Lin, Y Hu, and BE Huang. Simple and efficient analysis of disease association with missing genotype data. *The American Journal of Human Genetics*, 82(2):444–452, 2008.

[132] Frank Dudbridge. Likelihood-based association analysis for nuclear families and unrelated subjects with missing genotype data. *Human heredity*, 66(2):87–98, 2008.

[133] Dan L Nicolae. Testing untyped alleles (tuna)—applications to genome-wide association studies. *Genetic epidemiology*, 30(8):718–727, 2006.

[134] Autumn Laughbaum. *Comparing BEAGLE, IMPUTE2, and Minimac Imputation Methods for Accuracy, Computation Time, and Memory Usage*. Golden Helix, Inc., 2013.

[135] Romeo Rizzi, Vineet Bafna, Sorin Istrail, and Giuseppe Lancia. Practical algorithms and fixed-parameter tractability for the single individual snp haplotyping problem. In *Algorithms in Bioinformatics*, pages 29–43. Springer, 2002.

[136] Russell Schwartz et al. Theory and algorithms for the haplotype assembly problem. *Communications in Information & Systems*, 10(1):23–38, 2010.

[137] Derek Aguiar and Sorin Istrail. Hapcompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data. *Journal of Computational Biology*, 19 (6):577–590, 2012.

[138] Derek Aguiar and Sorin Istrail. Haplotype assembly in polyploid genomes and identical by descent shared tracts. *Bioinformatics*, 29(13):i352–i360, 2013.

[139] Bjarni V Halldórsson, Vineet Bafna, Nathan Edwards, Ross Lippert, Shibu Yooseph, and Sorin Istrail. A survey of computational methods for determining haplotypes. *Lecture Notes in Computer Science*, 2983:26–47, 2004.

[140] Olivier Delaneau, Jonathan Marchini, and Jean-François Zagury. A linear complexity phasing method for thousands of genomes. *Nature methods*, 9(2):179–181, 2012.

[141] Olivier Delaneau, Jean-Francois Zagury, and Jonathan Marchini. Improved whole-chromosome phasing for disease and population genetic studies. *Nature methods*, 10(1):5–6, 2013.

[142] Jared O'Connell, Deepti Gurdasani, Olivier Delaneau, Nicola Pirastu, Sheila Ulivi, Massimiliano Cocca, Michela Traglia, Jie Huang, Jennifer E Huffman, Igor Rudan, et al. A general approach for haplotype phasing across the full spectrum of relatedness. *PLoS Genet*, 10(4):e1004234, 2014.

[143] Matthew Stephens, Nicholas J Smith, and Peter Donnelly. A new statistical method for haplotype reconstruction from population data. *The American Journal of Human Genetics*, 68(4):978–989, 2001.

[144] Jonathan Marchini, David Cutler, Nick Patterson, Matthew Stephens, Eleazar Eskin, Eran Halperin, Shin Lin, Zhaohui S Qin, Heather M Munro, Gonçalo R Abecasis, et al. A comparison of phasing algorithms for trios and unrelated individuals. *The American Journal of Human Genetics*, 78(3):437–450, 2006.

[145] Eric Sobel and Kenneth Lange. Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics. *American journal of human genetics*, 58(6):1323, 1996.

[146] Gonçalo R Abecasis, Stacey S Cherny, William O Cookson, and Lon R Cardon. Merlin—rapid analysis of dense genetic maps using sparse gene flow trees. *Nature genetics*, 30(1):97–101, 2002.

[147] Benjamin French, Thomas Lumley, Stephanie A Monks, Kenneth M Rice, Lucia A Hindorff, Alexander P Reiner, and Bruce M Psaty. Simple estimates of haplotype relative risks in case-control data. *Genetic epidemiology*, 30(6):485–494, 2006.

[148] Jason P Sinnwell, Daniel J Schaid, and Zhaoxia Yu. *haplo.stats: Statistical analysis of haplotypes with traits and covariates when linkage phase is ambiguous*. Mayo Foundation for Medical Education and Research, 2007.

[149] Mohammad H Ferdosi, Brian P Kinghorn, Julius HJ van der Werf, Seung Hwan Lee, and Cedric Gondro. hsphase: an r package for pedigree reconstruction, detection of recombination events, phasing and imputation of half-sib family groups. *BMC bioinformatics*, 15(1):172, 2014.

[150] Danielle Welter, Jacqueline MacArthur, Joannella Morales, Tony Burdett, Peggy Hall, Heather Junkins, Alan Klemm, Paul Flicek, Teri Manolio, Lucia Hindorff, et al. The nhgri gwas catalog, a curated resource of snp-trait associations. *Nucleic acids research*, 42(D1):D1001–D1006, 2014.

[151] Darryl Leja, Teri Manolio, Tony Burdett, Dani Welter, and Helen Parkinson. *A Catalog of Published Genome-Wide Association Studies*. European Bioinformatics Institute, 2014.

[152] Jeffrey C. Barrett. *Genotype Imputation Enables Powerful Combined Analyses of Genome-Wide Association Studies*. © Illumina, Inc., 2010.

[153] Yurii S Aulchenko, Maksim V Struchalin, and Cornelia M van Duijn. Probabel package for genome-wide association analysis of imputed data. *BMC bioinformatics*, 11(1):134, 2010.

[154] Jonathan Marchini and Gavin Band. *SNPTEST2 Program*. University of Oxford, 2014.

[155] Yu-Fang Pei, Lei Zhang, Jian Li, and Hong-Wen Deng. Analyses and comparison of imputation-based association methods. *PLoS One*, 5(5):e10827, 2010.

[156] Peter Castaldi. *Quality Control for Genomic Data*. Adapted from Merry-Lynn McDonald, 2013.

[157] Michael R Barnes and Gerome Breen. *Genetic Variation: Methods and Protocols, Methods in Molecular Biology*, volume 628. Springer Science + Business Media, 2013.

[158] Carl A Anderson, Fredrik H Pettersson, Geraldine M Clarke, Lon R Cardon, Andrew P Morris, and Krina T Zondervan. Data quality control in genetic case-control association studies. *Nature protocols*, 5(9):1564–1573, 2010.

[159] Scitable. *Hardy-Weinberg equilibrium*. © Nature Education, 2009.

[160] Jacqueline K Wittke-Thompson, Anna Pluzhnikov, and Nancy J Cox. Rational inferences about departures from hardy-weinberg equilibrium. *The American Journal of Human Genetics*, 76(6):967–986, 2005.

[161] Michael E Weale. *Notes on GWAS QC/QA*. © King's College London, 2011.

[162] Sten Wahlund. Zusammensetzung von populationen und korrelationserscheinungen vom standpunkt der vererbungslehre aus betrachtet. *Hereditas*, 11(1):65–106, 1928.

[163] File Formats Task Team. *Variant Call Format Specification*. Data Working Group Consortium, 2013.

[164] Colin Freeman and Jonathan Marchini. *GTOOL Program*. Wellcome Trust Centre for Human Genetics, 2010.

[165] Olivier Delaneau, Jonathan Marchini, and JF Zagury. *SHAPEIT Prephasing*. CNAM and University of Oxford, 2014.

[166] Olivier Delaneau, Cédric Coulonges, and Jean-François Zagury. Shape-it: new rapid and accurate algorithm for haplotype inference. *BMC bioinformatics*, 9(1): 540, 2008.

[167] David L Morris, Patricia P Ramsay, Kim E Taylor, Lindsey A Criswell, Tim J Vyse, Glenys Thomson, and Lisa F Barcellos. *Genotype SNP Imputation Methods Manual*. The Immunology Database and Analysis Portal (ImmPort), 2010.

[168] Bryan Howie and Jonathan Marchini. *Using IMPUTE2 for phasing of GWAS and subsequent imputation*. University of Chicago and University of Oxford, 2010.

[169] Tanya Y Berger-Wolf, Saad I Sheikh, Bhaskar DasGupta, Mary V Ashley, Isabel C Caballero, Wanpracha Chaovalitwongse, and S Lahari Putrevu. Reconstructing sibling relationships in wild populations. *Bioinformatics*, 23(13):49–56, 2007.

[170] Eurodiab Ace Study Group et al. Variation and trends in incidence of childhood diabetes in europe. *The Lancet*, 355(9207):873–876, 2000.

[171] C.C. Patterson, G.G. Dahlquist, E. Gyürüs, A. Green, and G. Soltész. Incidence trends for childhood type 1 diabetes in europe during 1989-2003 and predicted new cases 2005-20: a multicentre prospective registration study. *The Lancet*, 373 (9680):2027–2033, 2009. cited By 678.

[172] J. Komulainen, P. Kulmala, K. Savola, R. Lounamaa, J. Ilonen, H. Reijonen, M. Knip, and H.K. Ãkerblom. Clinical, autoimmune, and genetic characteristics of very young children with type i diabetes. *Diabetes Care*, 22(12):1950–1955, 1999. cited By 127.

[173] E. Sabbah, K. Savola, T. Ebeling, P. Kulmala, P. Vahasalo, J. Ilonen, P.I. Salmela, and M. Knip. Genetic, autoimmune, and clinical characteristics of childhood- and adult-onset type 1 diabetes. *Diabetes Care*, 23(9):1326–1332, 2000. cited By 0.

[174] A. Neu, A. Willasch, S. Ehehalt, R. Hub, M.B. Ranke, S.A. Becker, V. Berg, R. Dürr, R. Eberle-Kuntz, W.K. Ertelt, U. Faller, L. Feldhahn, J. Grulich-Henn, H. Haug, W. Hecker, A. Henzler-Le Boulanger, U. Hermann, M. Herrmann, B. Höhmann, M. Kloc, U. Krauß, M. Metzel, B. Meyburg, M. Pizard-Weyrich, K. Placzek, D. Poletaew, R. Sauter, R. Schnarz, H.-J. Schreckling, H. Schulmayer, A. Schumacher, K.O. Schwab, M. Wabitsch, M.M. Walka, C. Wettach, and J. Wissert. Ketoacidosis at onset of type 1 diabetes mellitus in children - frequency and clinical presentation. *Pediatric Diabetes*, 4(2):77–81, 2003. cited By 0.

[175] K.O. Kyvik, L. Nystrom, F. Gorus, M. Songini, J. Oestman, C. Castell, A. Green, E. Guyrus, C. Ionescu-Tirgoviste, P.A. McKinney, D. Michalkova, R. Ostrauskas, and N.T. Raymond. The epidemiology of type 1 diabetes mellitus is not the same in young adults as in children. *Diabetologia*, 47(3):377–384, 2004. cited By 0.

[176] K.M. Gillespie, R.J. Aitken, I. Wilson, A.J.K. Williams, and P.J. Bingley. Early onset of diabetes in the proband is the major determinant of risk in hla dr3-dq2/dr4-dq8 siblings. *Diabetes*, 63(3):1041–1047, 2014. cited By 0.

[177] K.M. Gillespie, E.A.M. Gale, and P.J. Bingley. High familial risk and genetic susceptibility in early onset childhood diabetes. *Diabetes*, 51(1):210–214, 2002. cited By 63.

[178] E. Tuomilehto-Wolf and J. Tuomilehto. Is the high incidence of diabetes in young children diagnosed under the age of 4 years determined by genetic factors in finland? *Diabete et Metabolisme*, 19(1 BIS):167–172, 1993. cited By 12.

[179] D. Fava, S. Gardner, D. Pyke, and R.D.G. Leslie. Evidence that the age at diagnosis of iddm is genetically determined. *Diabetes Care*, 21(6):925–929, 1998. cited By 0.

[180] M.W. Klinker, J.J. Schiller, V.L. Magnuson, T. Wang, J. Basken, K. Veth, K.I. Pearce, L. Kinnunen, V. Harjutsalo, X. Wang, J. Tuomilehto, C. Sarti, and S. Ghosh. Single-nucleotide polymorphisms in the il2ra gene are associated with age at diagnosis in late-onset finnish type 1 diabetes subjects. *Immunogenetics*, 62 (2):101–107, 2010. cited By 0.

[181] V. Harjutsalo, T. Podar, and J. Tuomilehto. Cumulative incidence of type 1 diabetes in 10,168 siblings of finnish young-onset type 1 diabetic patients. *Diabetes*, 54(2):563–569, 2005. cited By 49.

[182] P.J. Bingley, I.F. Douek, C.A. Rogers, and E.A.M. Gale. Influence of maternal age at delivery and birth order on risk of type 1 diabetes in childhood: Prospective population based family study. *British Medical Journal*, 321(7258):420–424, 2000. cited By 0.

[183] V. Harjutsalo, A. Reunanen, and J. Tuomilehto. Differential transmission of type 1 diabetes from diabetic fathers and mothers to their offspring. *Diabetes*, 55(5): 1517–1524, 2006. cited By 32.

[184] V. Harjutsalo, N. Lammi, M. Karvonen, and P.-H. Groop. Age at onset of type 1 diabetes in parents and recurrence risk in offspring. *Diabetes*, 59(1):210–214, 2010. cited By 0.

[185] JA Noble, A. Martin, and AM Valdes. Type 1 diabetes risk for human leukocyte antigen (hla)-dr3 haplotypes depends on genotypic context: association of dpb1 and hla class i loci among dr3- and dr4-matched italian patients and controls. *Human Immunology*, 69(4-5):291–300, 2008.

[186] Stephen D Turner. qqman: an r package for visualizing gwas results using qq and manhattan plots. *bioRxiv*, page 005165, 2014.

[187] Christopher C Chang, Carson C Chow, Laurent CAM Tellier, Shashaank Vattikuti, Shaun M Purcell, and James J Lee. Second-generation plink: rising to the challenge of larger and richer datasets. *ArXiv e-prints*, 2014.

[188] Colin Freeman and Jonathan Marchini. *Hardy–Weinberg principle*. Genetics Software Suite, © The University of Oxford, 2009.

[189] A. Neu, S. Ehehalt, A. Willasch, M. Kehrer, R. Hub, and M.B. Ranke. Varying clinical presentations at onset of type 1 diabetes mellitus in children - epidemiological evidence for different subtypes of the disease? *Pediatric Diabetes*, 2(4): 147–153, 2001. cited By 0.