Technical University of Denmark

Human leukocyte antigen typing from next generation sequencing data

Master's Thesis

Author:

Nikolas Hallberg Thuesen

s152993

Supervisors:

Internal: Assoc. Prof. Gabriel Renaud

Internal: Assoc. Prof. Shyam Gopalakrishnan

External: Michael Schantz Klausen (Evaxion Biotech)

Hand-in date: 22nd January 2020

ECTS: 30

DTU Health Tech
Department of Health Technology

Abstract

The human leukocyte antigen (HLA) system is a group of genes coding for proteins that are central to the adaptive immune system and of great clinical importance. Identifying the specific HLA allele combination of a patient is relevant in organ donation, risk assessment of autoimmune and infectious diseases and cancer immunotherapy. The high genetic polymorphism in this region however makes HLA typing a difficult problem. With the advance of next-generation-sequencing (NGS), computational methods for NGS based HLA typing have emerged.

This project investigated the performance of the five of these NGS based HLA typing tools for the five HLA genes, HLA-A, -B, -C, -DRB1 and -DQB1 using 829 whole-exome sequencing (WES) samples from the International Genome Sample Resource. Specifically, this study looked at the typing accuracy and required computational resources of the tools and how these parameters depended on the coverage of the WES samples.

Optitype was found to have the highest typing accuracy for HLA class I genes (98.45% at 2-field resolution) and HLA*LA was found to have the highest typing accuracy for all five genes (95.72% at 2-field resolution). HLA*LA also had the highest peak memory usage (median of 31 GB) and Optitype had the lowest (median of 1.1 GB).

Furthermore, a subsampling study was carried out to evaluate the typing accuracy of the tools at differing coverages. It was found that HISAT-genotype, Kourami and STC-Seq require coverage above 100X to perform optimally while the typing accuracy of HLA*LA and Optitype only starts to drop significantly at coverages below 75X.

Contents

List of Abbreviations

ACT:	Adoptive Cell Transfer
ADO:	Allelic Dropout
ARD:	Antigen Recognition Domain
bp:	base pairs
BWT:	Burrows-Wheeler Transformation
CNV:	Copy Number Variation
CPI:	Checkpoint Inhibitor Therapy
FM index	Ferragina Manzini index
HLA:	Human Leukycote Antigen
IGSR:	Internation Genome Sample resource
HCT:	Hematopoietic Cell Transplantation
ILP:	Integer Linear Program
MHC:	Major Histocompatability Complex
MSA:	Multiple Sequence Analysis
OV:	Oncolytic Virus
PCR:	Polymerase Chain Reaction
SBT:	Sequence Based Typing
SNP:	Single Nucleotide Polymorphism
SSOP:	Sequence-Specific Oligonucleotide Probes
SSP:	Sequence-specific primers
WES:	Whole Exome Sequencing
WGS:	Whole Genome Sequencing

Acknowledgements

I want to thank my DTU supervisors, Gabriel and Shyam for guiding me through the project from start to finish through numerous Monday meetings which at their worst was mostly about my thesis and at their best hardly about my thesis.
This project was carried out with and at Evaxion Biotech. From here, I want to thank my external supervisor, Michael, for always having time to pause your own work to answer my questions, my mentor Anders and my fellow DTU student Charlotte, who first introduced me to Evaxion. To the rest of the wonderful people I have met at Evaxion I want to say thank you for lunch dates, Friday bars, runs around Kastellet, morning swims in cold water and for making me feel welcome and part of the company, even though I was just visiting.
The completion of this project means that I finish more than five years of studying at DTU. With that, I want to thank the many amazing people I have met and friends I have gained during my studies. I especially want to thank my colleagues from DTU ScienceShow for countless great memories and for making my university time fire.
Finally, I want to thank my family for their support and Ciara for necessary and unnecessary sparring, heaps of class craic and for translating my broken sentences into an "understandable" form of British English.

Chapter 1

Introduction and Theory

HLA typing is the determination of a person's specific allele combination in the human leukocyte antigen (HLA) region. This chapter aims to introduce the reader to the biological relevance of the HLA system, how individual HLA alleles are named and why HLA genotyping is clinically relevant. This chapter also includes a description of next-generation sequencing (NGS) as well as an overview of newly developed HLA typing workflows, where NGS data is used.

1.1 Biological Background

1.1.1 The adaptive immune system

The immune system is divided into two parts, the innate and the adaptive immune system. The innate immune system gives a quick and nonspecific response, while the adaptive immune system identifies an invading pathogen and develops a specific response. The pathogen and corresponding response are hereafter remembered so that the same pathogen can be fought more effectively in the future. The two systems work together and a response from the adaptive immune system depends on a previous innate response[1].
B lymphocytes (B cells) and T lymphocytes (T cells) are central to the adaptive immune system. These cells can distinguish foreign cells from native cells and combat the identified pathogens. The lymphocytes detect pathogens by binding to peptides derived from degraded proteins from the pathogen. These small molecules are called antigens and the part of the antigen, that binds to a receptor on a lymphocyte is called the epitope[1].
T lymphocytes express a specific T-cell receptor (TCR) for peptide recognition. The TCR recognises peptides when they are bound to a specialised antigen-presenting molecule called the major histocompatibility complex (MHC) and displayed on the cell surfaces. This is shown in figure 1.1.1. Each TCR is only able to recognise a distinct selection of peptides that matches the structure of the TCR binding pocket. This binding specificity is important to how the adaptive immune system detects pathogens and distinguishes them from healthy cells [1]. Cells can present peptides (also known as epitopes) to T-cells both from pathogens (non-self epitopes), tumour mutations (neoepitopes/neoantigens) and cells native to the body (self-epitopes). T cells are however generally able to recognise the difference. This means that a cell, which displays epitopes which indicate that the cell is, for example, infected by a virus or developing into a tumour cell, will trigger an immune response, but a healthy cell avoids doing so[2, 3, 4].

1.1.2 The HLA complex

MHC molecules are situated on the surface of cells and present peptides/antigens to the immune system, enabling the immune system to identify foreign peptides that indicate a

transformation in a cell due to, for example, infection or cancer. The MHC gene region is found in all jawed vertebrates and the genes are expressed co-dominantly. In humans, the region is called the human leukocyte antigen (HLA) complex[3, 5, 6].

HLA genes can be divided into three subclasses (class I, II and III) based on their structure and function. Generally, HLA class I proteins are found on most somatic cells and present peptides originating from intracellular proteins to $CD8^+$ cytotoxic T lymphocytes (CTLs), while HLA class II proteins only are found on antigen-presenting cells (APCs) and present exogenous peptides to $CD4^+$ helper T-cells[3]. This difference as well as some others between class I and class II molecules are shown in figure 1.1.1.

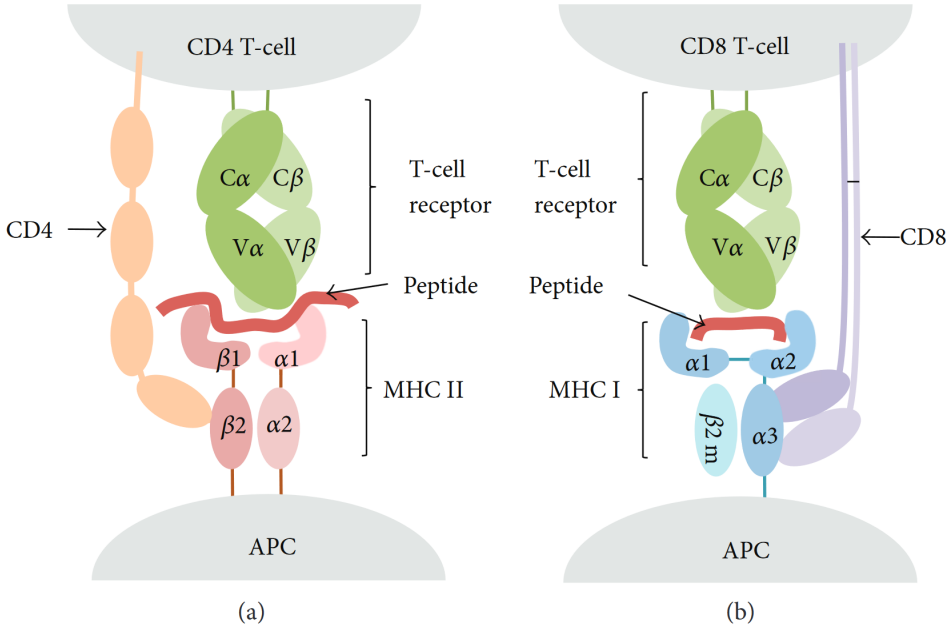

Figure 1.1.1: Epitope/peptide presentation on APCs to a $CD4^+$ helper T-cell via MHC II (a) and to a CTL ($CD8^+$ T cell) via MHC I. The two types of T-cells express either the CD4 or the CD8 coreceptor, which enables correct pairing between $CD4^+$ and MHC II/HLA II as well as between $CD8^+$ and MHC I/HLA I[1]. Figure adapted from [1] which incorrectly labeled both MHC complexes as MHC I

Both class I and class II molecules are heterodimers but they differ in the antigen recognition domains (ARD). In Class I molecules, this region is made up of the $\alpha 1$ and $\alpha 2$ domains, which are part of the same monomer and are encoded by the same gene. Conversely, the binding region of class II molecules consists of two domains ($\alpha 1$ and $\beta 1$) that belong to two different monomers and are encoded by two different genes. Class I molecules typically

bind peptides that are 9 amino acids long[7]) and the peptides binding to class II molecules are typically 13-25 amino acids long[8, 9]).
HLA class III proteins are not involved in antigen presentation but are instead associated with inflammatory responses, leukocyte maturation and the complement cascade[2, 10]. This project only focuses on HLA class I and II, as these are the antigen-presenting proteins.

The general trend of class I proteins presenting intracellular peptides and class II proteins presenting exogenous has some exceptions - one example is cross-presentation. This process can occur in situations where a pathogen or tumour cell avoids normal detection by the HLA system. In these cases, peptides from the infected cells can be transported to APCs and be presented on HLA class I molecules despite being extracellular. This can then trigger an immune response called cross-priming, where CTLs fight the pathogen or tumour cell, that originally produced the peptide[3, 11].

Several HLA class I genes exist. HLA-A, HLA-B and HLA-C are called the classic HLA-I genes (as these are important for organ transplantation) while HLA-E, -L, -J, -K, -H and -G are examples of less important, non-classical HLA-I genes. The structure of an HLA class I gene is shown in figure 1.1.2. Likewise, several types of HLA class II genes exist. HLA-DP, -DQ, and -DR are the classical HLA class II genes and additional examples are HLA-DM, -DM and -DO. Here, the first letter (D) indicates the class and the second letter indicates the family. HLA class II genes consist of both an α- and a β chain (as seen in figure 1.1.3) and several versions of these exist. Examples of HLA class II genes are therefore *HLA-DPA1, HLA-DRB1, HLA-DRB2* and *HLA-DRB3*. More examples can be seen in figure A.0.2 in appendix A.
The peptides presented by HLA molecules are not just random fragments of a degraded protein. Instead, an HLA molecule is highly selective and only recognises peptides that match the proteins binding cleft. Whether a peptide can be presented by a specific HLA molecule, therefore, depends on the structure of the binding cleft, and consequently on both the HLA gene and the specific version of the gene that is found in an individual (the specific HLA allele)[12, 13]

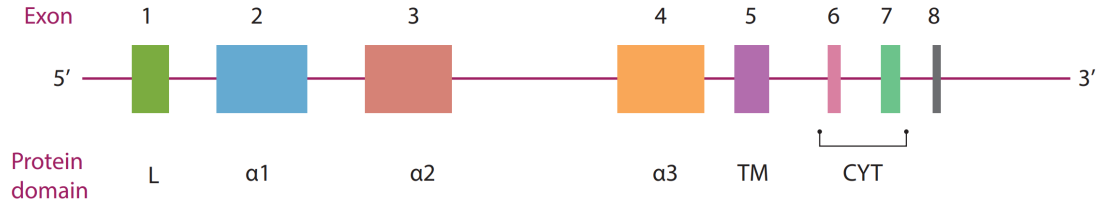

Figure 1.1.2: The exon-intron organisation of a HLA-class I gene. Exons 2 and 3 code for the ARDs[2]. Figure taken from [2].

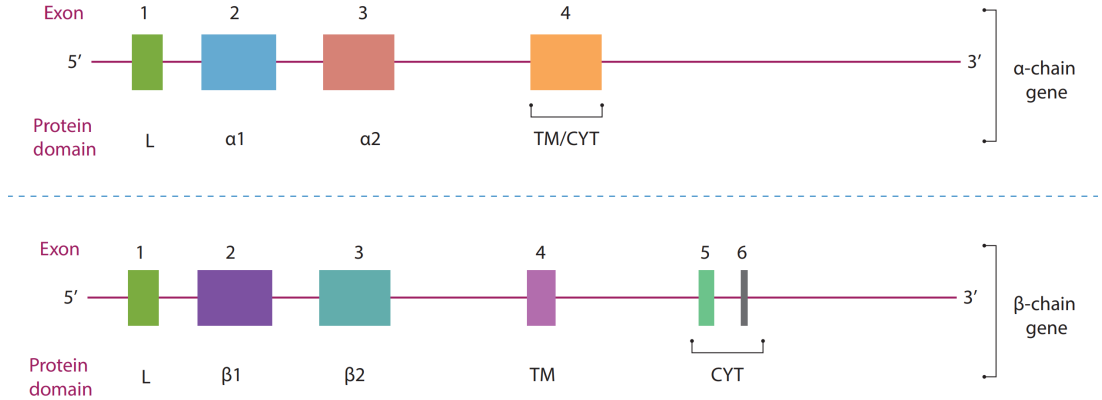

Figure 1.1.3: The exon-intron organisation of a HLA-class II gene. Both the α chain and the β chain gene code for proteins, which have a domain that is part of the ARD. This domain is encoded by exon 2 in both genes[2]. Figure taken from[2].

1.2 HLA nomenclature

The HLA region - with its 45 genes - is one of the most polymorphic regions in the human genome. Currently, over 28000 different allele sequences are known for the region and more are consistently discovered[5] (see also Appendix A).

New alleles are registered and named by the WHO Nomenclature Committee for Factors of the HLA System and stored in the IPD-IMGT/HLA database. The database is continuously updated to correct errors and includes new alleles[5].

The naming of a new HLA allele follows a 2010 naming convention[14, 15] illustrated in figure 1.2.1. An allele name is separated into four fields each consisting of 2-3 digits. The first and most important field denotes the allele group or family i.e. alleles coding for HLA molecules that can bind the same antigen or have a high degree of sequence homology to the rest of the alleles in the group[14]. The second field notes an allele within an allele family that codes for one specific protein sequence. The third field is used to distinguish between alleles that code for the same protein sequence but are different due to synonymous mutations in the exons/coding region and the fourth field is used to distinguish alleles that solely differ in introns/non-coding regions. The actual numbers in the fields 2, 3 and 4 often simply indicate the order of discovery[14].

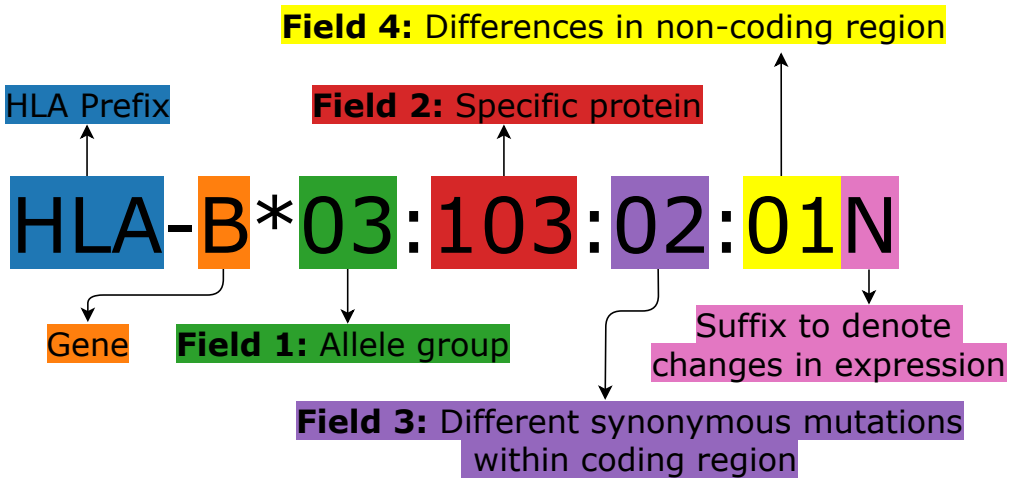

Figure 1.2.1: HLA nomenclature shown with four field (8-digit) resolution. This figure is adapted from the IPD-IMGT/HLA website (`http://hla.alleles.org/nomenclature/naming.html`) [**?**, 16, 17] and made using `diagrams.net`[18].

A specific allele has at least 2 fields in its name as the third and fourth field are only used

when two or more alleles share the two first fields and further specification is needed to resolve the ambiguity[5]. The name "HLA-A*01:01" refers to several alleles which all code for the same protein and the third and fourth field are therefore needed here - see also figure 1.2.2. In addition to the four fields, trailing letters are used to add information about expression for a specific allele or to note an allele group. An overview of the meaning behind these trailing letters is shown in table 1.2.1[5].

1.2.1 Ambiguous and Alternative allele names

The most important part of the HLA molecule is the binding region - the ARD. This domain is coded by exon 2 and 3 in class I molecules and only exon 2 in HLA class II molecules. The most important differences between alleles are therefore the ones affecting nucleotides in this region[14]. This is the basis of two alternate ways of grouping alleles - G group and P group - that exist on top of the classical naming system described in figure 1.2.1.

Alleles belonging to the same G group share identical nucleotide sequences in exons coding for the ARD while alleles belonging to the same P group have identical peptide sequences in the ARD. G group resolution focuses on the genomic level, while P group resolution focuses on the proteomic level. An example of a G group is HLA-A*02:01:01G and an example of a P group is HLA-A*11:02P

The name of a G group always includes three fields (at least six digits) preceding the 'G' since the smallest difference that separates two G groups is a synonymous mutation. P group resolution only has two fields followed by a P, as the smallest difference between two P groups is found on proteomic level.

G and P groups only have a trailing letter (G or P) if the group consists of more than one allele. The group A*01:01:01G has a trailing 'G', as it consists of the alleles A*01:01:01:01, 01:01:01:02N as well as more than 100 additional alleles. The "group" A*01:01:02 is only made up of a single allele, and is therefore lacking a trailing G. Both A*01:01:01G and A*01:01:02 would however be the correct names of alleles using G group resolution.

A special feature P group resolution is that alleles need to be able to express a functioning protein before they are included in a P group. The allele A*01:04N (a null allele) is part of the A*01:01:01G group and therefore also shares an identical nucleotide sequence across

exon 2 and 3 with all alleles in the A*01:01P group. A*01:04N is however not part of this P group because this version of the HLA-A gene does not express a functioning protein. This exclusion of null alleles is further discussed in section 2.3.1[16].

Deprecated and/or alternative names for alleles and allele resolutions than the official names described above sometimes appear in the literature. 4-field resolution is, for example, sometimes referred to as 8 digit resolution, as when fewer alleles were known, each field only consisted of two digits. P group resolution or higher is also referred to as ”high resolution” while ”low resolution” is another term for 1-field resolution. Allelic resolution is the number of fields needed to refer to one unique allele. A*01:01:01:01 and A*01:12 are both allele names in allelic resolution because they unambiguously refer to one single allele whereas A*01:01 is not a name in allelic resolution. Many alleles sharing the first two fields ”A*01:01” but only one has the first two fields ”A*01:12”[5, 19, 20].

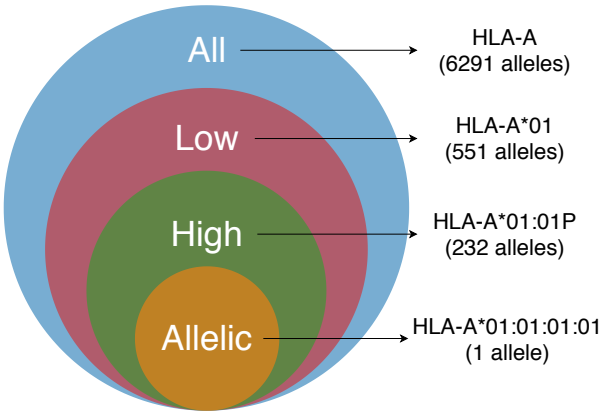

Figure 1.2.2: (Unofficial) HLA allele resolutions according to [19] with the allele HLA-A*01:01:01:01 used as example. The number of alleles on each resolution level is found by using the IMGT/HLA Allele Query Form and the provided list of P alleles[16]. Note: null alleles are excluded in P group resolution. The figure is adapted from [19, 21, 22] and made using `diagrams.net`[18].

Table 1.2.1: Explanation of trailing letters in HLA nomenclature. Previously, other letters than the ones shown in this table have been used, but these are the ones that are currently used and listed on the IPD-IMGT/HLA website[16, 22].

Suffixes denoting a change in expression	
HLA-A*24:09N	A null allele
HLA-A*30:14L	Significantly reduced or 'Low' cell surface expression
HLA-A*24:02:01:02L	Significantly reduced or 'Low' cell surface expression, where the mutation is found outside the coding region
HLA-B*44:02:01:02S	The protein is only expressed as a 'Secreted' molecule
HLA-A*32:11Q	An allele with 'Questionable' expression
Suffixes denoting allele groups	
A*02:01P	A group of alleles which share the same peptide sequence across the ARDs.
A*02:01:01G	A group of alleles which have identical nucleotides across the exons coding for the ARDs.

1.3 Disease associations of the HLA complex

The enormous diversity of HLA genes is thought to be a result of heterozygous individuals having an advantage over homozygous individuals. The HLA system in heterozygous individuals can bind and present a wider range of peptides to T-cells resulting in better detection of and protection against pathogens. It is further thought that heterozygous individuals with more dissimilar alleles have an additional advantage compared to individuals with heterozygous but similar alleles. The diversity of the HLA region is, therefore, an adaptation to combat diseases and the region is also associated with more diseases (mainly autoimmune and infectious) than any other gene region in the human genome[23, 24].
This section contains a brief overview of some of the diseases associated with the HLA complex. As opposed to a complete description of the HLA complex's role in these diseases, the following should serve as a justification of the relevance of researching the HLA complex and the importance of determining the specific HLA alleles (the HLA profile) of an individual. This section does not cover all medical situations associated with the HLA system, for example, pregnancy problems, drug sensitivity and HLA deficiency problems. It should be noted that not all the mechanisms behind all the disease associations are fully known. There are claims that some of the diseases correlated with specific HLA alleles are instead caused by other genes that are located close to the HLA locus. If these claims are correct, the correlation between disease and a specific HLA allele would be a result of linkage disequilibrium (LD). An example of this is the close association between the HLA-DQ allele (DQB1*06:02) and narcolepsy. The disease is caused by a mutation in the hypocretin receptor 2 gene, but the receptor gene is in close LD with HLA-DQ, and specific alleles, therefore, appear together. Another important point about these disease association studies of the HLA system is that subsequent studies often fail to replicate the results and some originally found associations might be due to errors in the original studies[12, 15, 25].

1.3.1 Autoimmune diseases

Autoimmune diseases are the result of self-peptides mistakenly being recognised as foreign by the immune system. This leads to the targeting and destruction of the organism's own cells. In healthy individuals, T cells are put through a strict selection process during

maturation to prevent this from happening. In this process, T cells interact with both class I and class II HLA molecules and develop into either CD4 helper or CD8 killer T-cells. The cells that develop TCRs with a strong affinity to self-peptides undergo apoptosis and are not released into the body[2, 12].

The association between autoimmune diseases and the HLA region has been studied for more than 50 years and several HLA alleles have been associated with a higher risk of these diseases. Without providing a complete list here, alleles from both the HLA class I and class II regions are associated with type 1 diabetes (T1D) and Grave's disease, class I is further associated with multiple sclerosis (MS) and class II with rheumatoid arthritis. Examples of disease associations for some alleles are listed in table 1.3.1.

Table 1.3.1: Examples of disease associations of HLA alleles[12, 26]

Disease	**Allele(s) associated with risk or protection**
T1D	Risk: B*39:06, DR3 haplotype (DQA1*05:01-DQB1*02:01) and DR4 haplotype (DQA1*03:01-DQB1*03:02)
MS	Risk: DR15 haplotype (DRB1*15:01-DQA1*01:02-DQB1*06:02-DRB5*01:01)
Malaria	Protection: B*53, DRB1*13:02/DQB1*05:01
HIV	Protection: B*27 and B*57

1.3.2 Infectious diseases

The immune system detects invading pathogens through presentation of degraded peptides from the pathogen on HLA molecules and recognition by TCRs. Since the HLA molecules only bind peptides which fit their peptide-binding cleft, a specific pathogen with its select proteins might therefore be more easily recognised by one HLA allele and more readily avoid detection from others. Examples of HLA alleles that seem good at detecting specific diseases are HLA-B*53 (predominantly found in African populations) and HLA-DRB1*13:02/DQB1*05:01 which seem to be correlated with a degree of protection from malaria[12].

HIV displays one of the most consistent associations with HLA. A normal progression of the disease is: initial symptoms upon infection of the virus, followed by an asymptomatic period of 8-10 years. Hereafter, $CD4^+$ counts drop and the immune defence is significantly weakened. Patients who are homozygous at the HLA locus however show a more rapid progression than heterozygous patients. This is because the immune system in heterozygous individuals can recognise a more diverse range of foreign antigens due to these individuals expressing two different versions of an HLA gene instead of only one version. This broader range of HIV peptides recognised by HLA heterozygous individuals brings a more efficient cytotoxic T-cell response and further makes it more difficult for the virus to acquire mutations that help the virus avoid or combat the immune system. Studies of HLA and HIV infection have found that class I alleles form a continuous spectrum of protective to hazardous alleles with B*27 and B*57 being especially protective. Interestingly, the protection of B*27 is also associated with protection against some other viral infections e.g. hepatitis C, hinting, that the mechanism with which the allele protects against HIV is not specific to this individual virus[12].

Recent COVID-19 studies have also investigated potential correlations between HLA alleles and disease spread/severity. An Italian study found that HLA-B*44 and HLA-C*01 were independently and positively associated with COVID-19[27], and an American study investigated whether the before mentioned B*27 allele gave protected against COVID-19 but found no significant correlation[28].

As HLA molecules are co-dominantly expressed and heterozygous individuals seem to have an immunological advantage, it makes sense for an individual to aim for a partner with a differing HLA profile. Such studies of how mating preferences are related to HLA profiles have been carried out for both rodents and humans, and some do indeed find a preference for HLA dissimilarity in the choice of partner[12].

1.3.3 Transplantaion

Malfunction of the HLA system can - as explained - result in diseases. In the case of transplantation, however, the generally well-functioning and highly specific system causes problems. When tissue/an organ is transplanted, recipient T cells might recognise the

donor's foreign HLA molecules. This can happen on donor APCs (direct presentation) but foreign HLA molecules can also be transported to recipient APCs and be recognised here (indirect presentation). This especially causes problems if the donor and the recipient have different HLA profiles. Therefore, HLA matching between donor and recipient is beneficial in both solid-organ transplantation and hematopoietic cell transplantation (HCT) [12].
The importance of the HLA complex in transplantation has been known for quite some time. The first descriptions of MHC proteins were in studies of rejection of tumours explanted from mice into other mice in the 1930s[29]. HLA matching has been used extensively in organ allocation for kidney transplantation where mismatches might result in immunological rejection or *de novo* donor-specific HLA antibody development[30]. More recently, advances in immunological suppression protocols that can be used to manage rejection episodes have lessened the importance of HLA matching, but multiple studies have shown that even with these new technologies, HLA matching still makes a significant positive difference[12].

1.3.4 Cancer

Cancer is a collection of related diseases in which self-cells in an organism undergo genetic changes that enable unlimited proliferation and tissue invasion. Cancerous cells have mutations that differ these cells to healthy cells, meaning that they also display different peptides on HLA molecules on the surface. These peptides, which are products of genetic alterations, are called neoepitopes and can usually be distinguished from self-epitopes by the T-cells (see also section 1.1.1). Abnormal mutations would normally make cells targets for destruction by the immune system so tumour cells need to actively avoid detection and destruction. Examples of avoidance mechanisms are loss of expression of tumour antigens/neoepitopes, production of immunosuppressive cytokines and altered HLA class I surface expression and/or function. Additional characteristics are needed for cells to develop into malignant tumour cells (cancer). The full set of attributes are referred to as the "Hallmarks of Cancer"[12, 31, 32].
When the immune system detects and destroys tumour cells this creates a selective process which favours less visible variants. This notion, that the immune system shapes tumour cells, is called cancer immunoediting and the process consists of three phases - (A) elimina-

tion, (B) equilibrium and (C) escape, which are shown in figure 1.3.1. In the elimination phase, the immune system can destroy tumour cells faster than they appear and thus ends in the host being completely tumour free. If single cells escape apoptosis, they enter the equilibrium phase, where the tumour cells are prevented from further outgrowth but not destroyed. Immunoselection and - as a consequence of this - immunoediting happens in this stage. The tumour might stay in the equilibrium phase but stronger, more immunoresistant cell variants might also appear, pushing the tumour into the final phase. This could be cells which avoid detection, are insensitive to immune effector mechanisms or induce an immunosuppressive state in the surrounding environment[31, 33].
Specific HLA alleles seem to have an impact on cancer prognosis e.g. HLA-A*02 is associated with a poor prognosis in ovarian, prostate, melanoma and lung cancer[12].

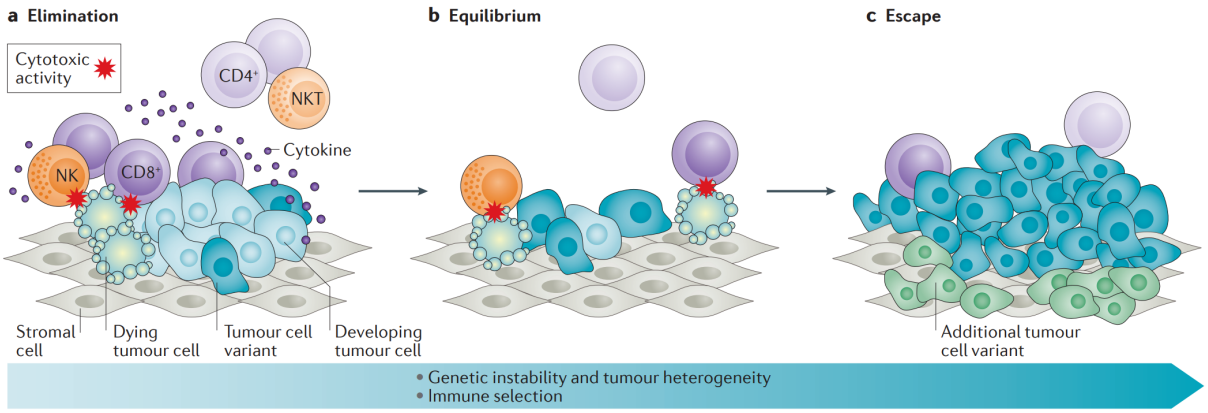

Figure 1.3.1: According to the theory of immunoediting, a tumour progresses through three phases before developing into cancer[33]. Figure adapted from [33]

1.3.5 Immunotherapy and neoepitope prediction

The HLA allele combination (HLA type) of an individual is valuable information which can be used to find specific alleles with a higher risk of disease and/or to match organ donors. A more recent use for is in cancer immunotherapy.
Immunotherapy is a cancer treatment method where the host's own immune system is used to combat and kill cancer cells and push tumours towards the elimination phase. Main strategies of immunotherapy include adoptive cell transfer (ACT), checkpoint inhibitor therapy (CPI), oncolytic virus (OV) therapy and cancer vaccines[34].

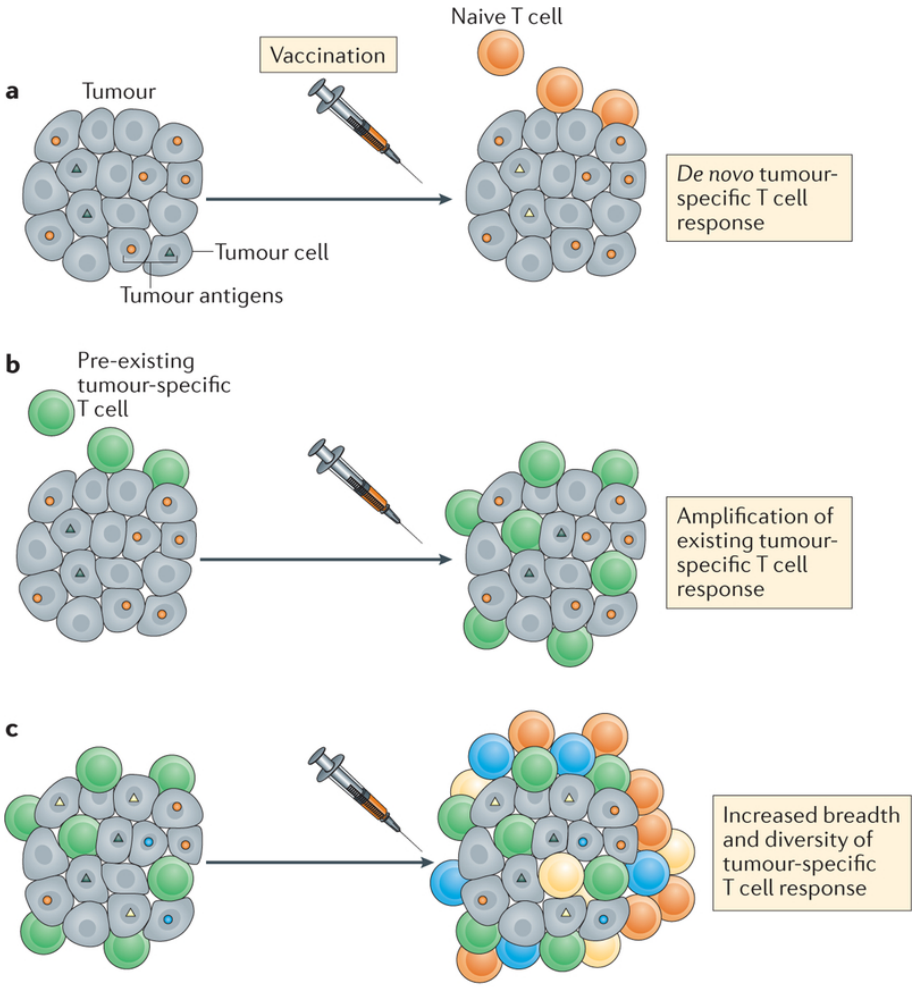

Figure 1.3.2: Cancer vaccines can give tumour regression in various ways. **a:** The specific antigens in a cancer vaccine can bring a new antigen-specific T cell response to previously untargeted tumour cells. **b:** If a T cell response is already present, a cancer vaccine further amplifies the response or **c:** gives a more diverse response. Figure taken from [35].

The first step of ACT is to isolate a culture of lymphocytes from a patient or a donor *in vitro*. The cells are then activated, selected for tumour specificity and expanded before being reinfused into the patient[34, 35].

CPI targets some of the immune inhibitory pathways that cancer cells exploit to suppress the immune system and avoid T cell activation. These T cell brakes are targeted and inhibited by monoclonal antibodies leading to T cell activation and increased immune response. One challenge of CPI is that the method lacks specificity which means that the tumour suppression from CPI also comes with toxic side effects in other parts of the body[34, 35].

Oncolytic viruses (OVs) selectively target, replicate in and kill tumour cells. OV's can be

designed to help immune cell recruitment and activation, but can also itself be targeted and cleared by the immune system, before having a real effect[34].

Some OVs contain antigens which are foreign to the human body but found in some tumour cells. Because of this tumour specificity, these antigens can be used in cancer vaccines. Several vaccines using such antigens have already been successful in preventing HPV-associated cancers. However, not all tumour-specific antigens (neoepitopes) are found in OVs. Therefore a more general approach is to analyse the tumour and healthy cells from the patient, find a selection of tumour-specific neoepitopes and design vaccines based on these. This vaccine would be specific to both the patient and the tumour[35]. Cancer vaccines have multiple effects which all seek to manipulate the immune system to target tumour cells as shown in figure 1.3.2).

Previously, discovering tumour antigens and developing cancer vaccines were very expensive, but the emergence of next-generation sequencing (further explained in section 1.4) has resulted in a significant price reduction[35]. An article from 2020[36] lists 20 ongoing and 7 completed clinical trials which target cancer neoepitopes. Neoepitope discovery is, however, not limited to cancer vaccines. Figure 1.3.3 shows a basic framework of neoantigen discovery for ACT.

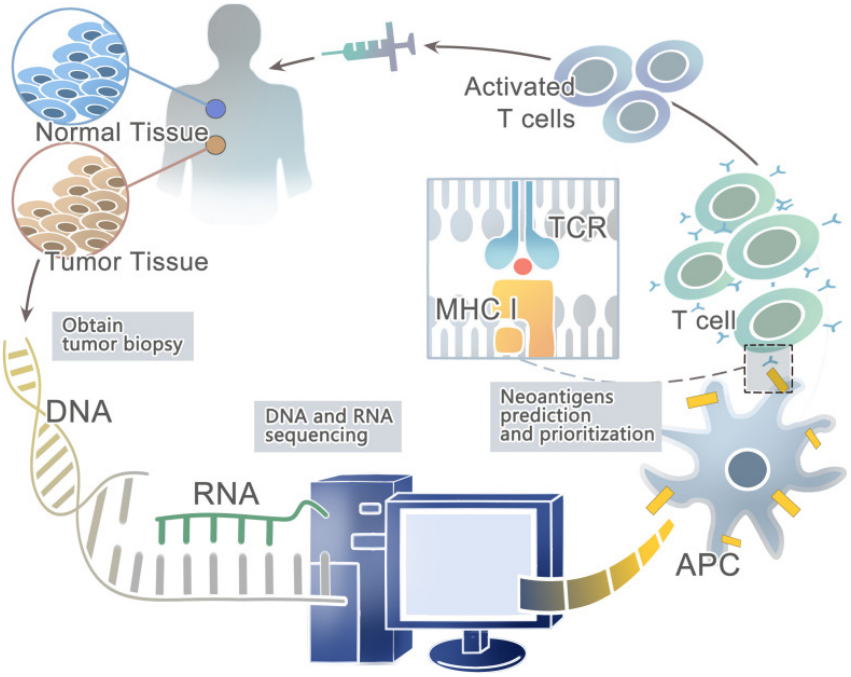

Figure 1.3.3: Identification and prioritisation of cancer neoepitopes for an individual cancer sample. First, next-generation sequencing is used to sequence both healthy and tumour tissue. Sequence data from healthy cells and HLA typing algorithms are then used to find the patient's HLA type (see also section 1.5). The HLA type is hereafter used together with sequence data from both healthy cells and tumour cells to find a selection of tumour-specific neoepitopes. The selected neoepitopes are then used to activate T cells which, when reinfused, can give a tumour-specific immune response. Figure taken from [36].

1.4 Next-Generation Sequencing

The Human Genome Project was a huge, worldwide genome sequencing project which was completed in 2003 and cost more than 5 billion US dollars (2020 currency rate - adjusted for inflation). One of the primary goals was to sequence a high-quality version of the human genome[37, 38]. Since then, the price of sequencing a human genome has diminished greatly (figure 1.4.1)[39].

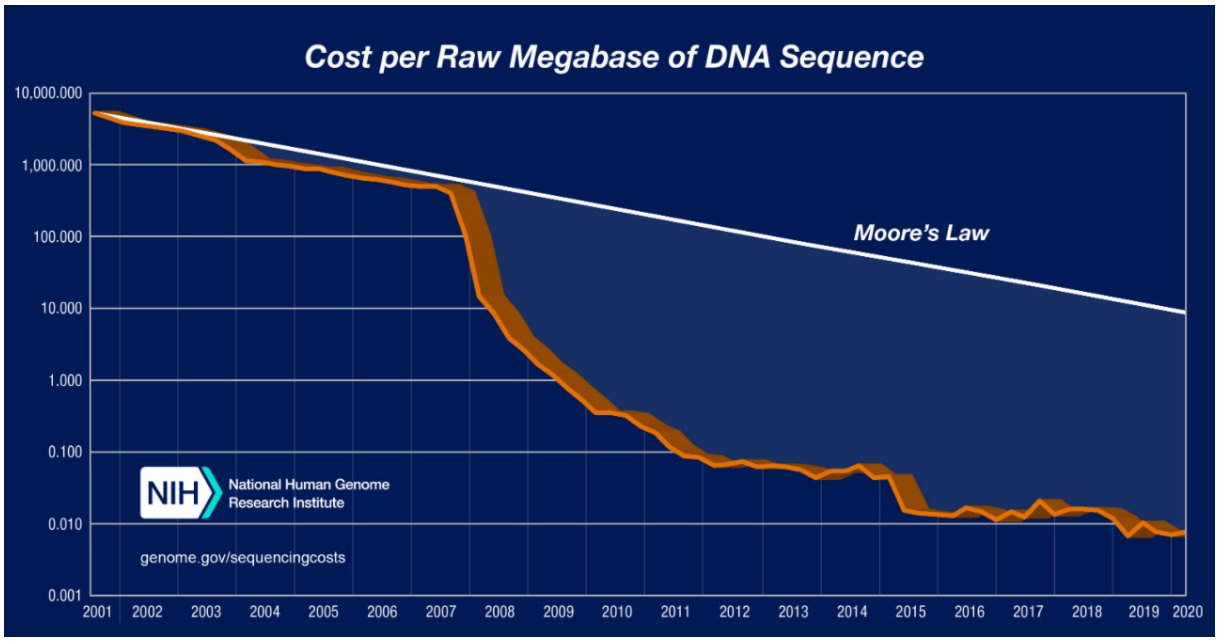

Figure 1.4.1: The cost of sequence data has been falling rapidly since 2000 and has since 2008 outpaced Moore's law. Figure taken from [39].

One of the earliest DNA sequencing methods is Sanger sequencing from 1977. The method consists of several steps. First, the original DNA is fragmented and amplified. Hereafter the generated DNA templates are mixed with specifically designed DNA primers, DNA polymerase and both labelled and unlabelled dideoxynucleotides. The primer then binds to single-stranded DNA templates and the DNA polymerase can add the dideoxynucleotides to match the DNA template. By labelling a specific dideoxynucleotide in each reaction mix, sequence detection is then possible using electrophoretic separation. The output data is high quality (up to 99.999% accuracy) sequence reads of around 1000 base pairs (bp). A modified version of this process was still the leading sequencing technology during the human genome project and until 2005, when massively parallel DNA sequencing/next-generation sequencing (NGS) was introduced[40, 41].

A large number of different NGS methods have been developed since 2005 and many of them - like Sanger - use sequence by synthesis (SBS) to generate data. However, instead of doing sequencing and detection in two separated steps, NGS technologies using SBS couple and parallelize these two processes. This enables sequencing of up to billions of templates simultaneously compared to a single template in Sanger. One drawback is, that the output data is of both lower accuracy and lower average read length compared to Sanger[40, 41].
Some of the latest NGS methods employ single-molecule sequencing (SMS) technology instead of SBS. These are sometimes referred to as third-generation sequencing implying SBS as the second generation and Sanger as the first. Unlike the prior generations, SMS technologies do not use DNA amplification in the process but can reduce potential bias by using the original genetic sample[42]. The single-molecule real-time platform from Pacific Biosciences and Oxford Nanopore Technologies are prominent examples of technologies employing SMS.
Compared to SBS methods, SMS have a lower accuracy but offer greatly longer reads which can provide an advantage, for example in *de novo* assembly where the goal is to find the sequence of a genome from many short reads without using a reference genome (described in section 1.4.4)[40, 41, 42].

1.4.1 Illumina sequencing

Illumina sequencing is the most widely used NGS technology today[43]. Several different Illumina technologies are commercially available, but they generally offer short read data of 150-300 bp[44]. Data generation projects, such as the 1000 Genomes Project (see also section 2.2), have used the SBS based technology to produce vast amounts of sequencing data[45]. The sequencing data in the gold standard dataset used to benchmark HLA typing algorithms in this project is made using Illumina.

Illumina sequencing consists of three steps, shown in figure 1.4.2. First, a library is created by taking fragmented DNA molecules and ligating two different adaptors to both ends of the molecule. These adaptors can have bar codes for later identification. The library is then amplified with longer primers further extending the adaptor sequences.

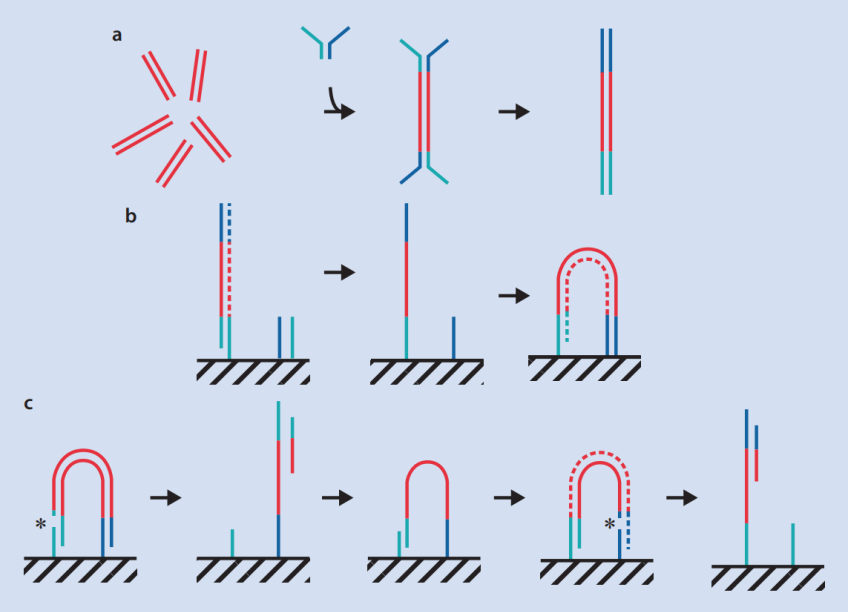

Figure 1.4.2: The three central steps of Illumina sequencing are **a:** Ligation and extension of adaptors, **b:** bridge amplification through multiple PCR cycles and **c:** sequencing by synthesis of single-stranded templates. Figure taken from [41].

The double-stranded molecules are then separated into single strands and added to a flow cell. The ligated adaptors are here able to bind to complementary strands/oligomers. The single-stranded molecules are copied, forming a double-stranded "bridge" attached on both ends to the flow cell. Multiple polymerase chain reactions (PCRs) are used for amplification, creating clusters of the flow cell which each are made from the same template. Finally, nucleotides carrying a laser detectable fluorophore and a chemical blocker are added in cycles. The blocker ensures that only one nucleotide is added per cycle, and the clusters emit light in a colour specific to the added base. The DNA sequence of the original single-stranded templates can then be read by looking at sequential light emissions from the flow cell[41].

1.4.2 The pitfalls of PCR

Bridge PCR amplification is central to the Illumina sequencing technology and it is therefore important to be aware of bias and potential problems related to this process. Three important errors: amplification bias, PCR crossover and PCR stutter.
Amplification bias means that one or both alleles are not amplified adequately. This can

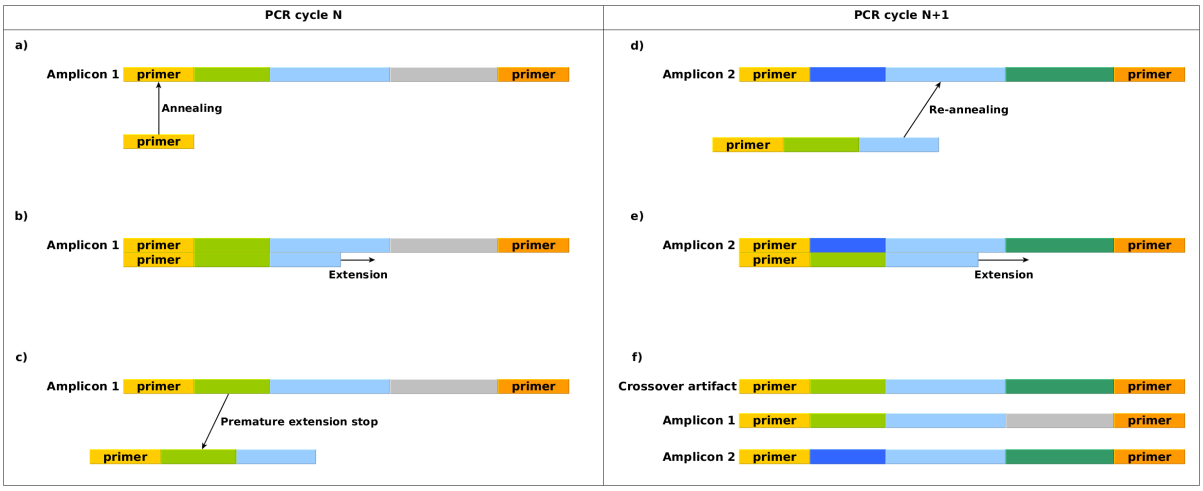

Figure 1.4.3: a: The beginning of PCR cycle N runs as expected, and **b:** amplicon 1 is partly replicated. **c:** After an extension step is interrupted, **d+e:** the half finished copy of amplicon 1 is only finished using amplicon 2 as template **f:** resulting in a crossover artifact[46, 47]. Figure taken from[47].

be due to differences in primer binding, GC content, length, low DNA quality or certain DNA structures such as G-quadruplexes. Small amplification differences between the alleles are expected. However, sometimes alleles are not amplified enough to be detected or - in extreme cases - not amplified at all. This is called allelic dropout (ADO)[46, 47].
PCR crossover (see figure 1.4.3) happens during a round of amplification when parts of two different alleles are merged into one single sequence instead of remaining separate. This new sequence is then replicated in subsequent amplification steps. Whether crossover happens depends on the number of cycles and the initial template concentration, amongst other things[46, 47].
Short tandem repeats (STRs) are genome sections, where a short nucleotide motif of a few bases is repeated several/many times[47]. PCR stutter happens when a repeat unit is deleted or added during a round of amplification. The difference between the alleles HLA-DRB1*03:01:01:01 and DRB1*03:01:01:02 is only a SNP in intron 1 and a single repeat unit in intron 2[46, 47]
The artefacts described here only rarely affects performance. However, in cases where high accuracy is required, e.g. in the determination of HLA alleles for clinical purposes, it is important to be aware of this pitfall.

1.4.3 WGS and WES

The goal of whole-genome sequencing (WGS) is to sequence the full genome of an organism without focusing on any specific region. In many cases, important information is however not distributed evenly across a genome[48].

Target-enrichment sequencing aims to capture and sequence a specific part of the genome. This could, for example, be used in a large population study of a specific gene or to determine whether a patient has specific alleles associated with a high risk of diseases, such as cancer. A variety of target-enrichment methods exist[49].

A generalised target-enrichment method is called whole-exome capture. This technique aims to capture all the protein-coding regions (the exome) of an organism. The exome only makes up about 1 % of the entire human genome but a majority of genetic disorders can be attributed to these regions. WES can therefore be a cheaper alternative to WGS and it can offer higher coverage (amount of reads per reference nucleotide) in the important coding regions[48].

An example of a whole-exome capture technique is array-based hybridisation. This is based on array bound probes that are complementary to the targeted exonic regions, and therefore able to capture them. When fragmented DNA is added to the array, the probes only hybridise to the exonic DNA and the remaining unbound DNA can be washed away. The captured exons can then be sequenced using, for example, Illumina technology. Exome capture and subsequent sequencing are referred to as whole-exome sequencing (WES)[48].

1.4.4 The human reference genome

Raw NGS data from Illumina usually consists of millions of short reads, all originating from one long genome. Finding the correct order of the sequences and assembling the original genome can be quite difficult. If no reference is used to assemble the genome from raw data, the process is referred to as *de novo* assembly. High coverage, read accuracy and paired-end read information (see figure 1.4.4) are especially useful in *de novo* assembly[50]. Humans only differ on about 20 million base pairs out of the 3.1 billion base pairs in a haploid genome (0.6%)[51, 52] and an easier way of getting useful information from raw NGS data is therefore to use a reference genome. By using a reference, which closely resembles the sequenced genome, ordering the raw reads becomes significantly easier. An

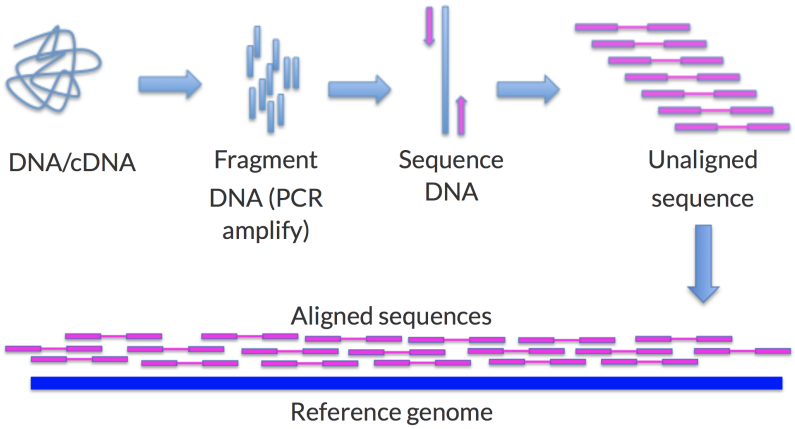

Figure 1.4.4: Aligning NGS reads to a reference genome using paired-end sequencing. Paired-end reads are two reads which originate from the same DNA fragment. They can be either overlapping or separated by a gap called an insert. Reads coupled in this way make subsequent assembly or alignment easier[50]. Figure taken from [53].

example of how reads are aligned to a reference genome is illustrated in figure 1.4.4.

One example of a reference genome is the first human genome published in the The Human Genome Project[37]. In the early days of genome sequencing, it was thought that the majority of the genetic variation between individuals was due to single nucleotide polymorphisms (SNPs), where one single nucleotide is substituted. These small differences between a sequenced genome and a reference genome are relatively easy to spot. The sequenced reads most likely still map to the correct position in the reference genome and all sequenced reads differ from the reference genome on this one specific position (see figure 1.4.4). The assumption of SNPs accounting for a majority of variation is however not true. Some regions of the genome have a high degree of structural variation that is much more complicated than substituting a single nucleotide. One type of variation is copy number variation (CNV) where a section of a genome is repeated a specific number of times that varies between individuals. Another example is the previously mentioned and highly polymorphic HLA region. The found diversity is accounted for in newer releases of the human reference genome by including multiple possible paths through the genome. The first patch of the current build (GRCh38.p1) has eight different paths through the HLA region. In practice, a primary assembly is released with multiple alternative DNA segments as illustrated in figure1.4.5[54].

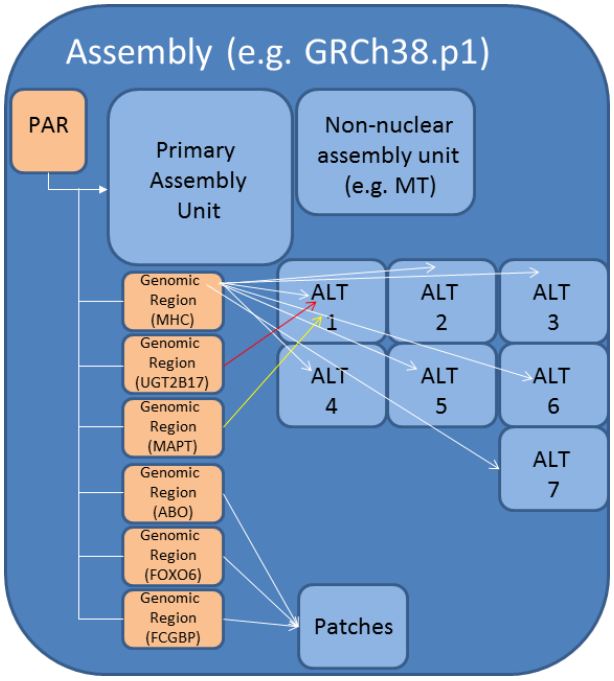

Figure 1.4.5: An overview of the first patch of the newest build of the human reference genome (GRCh38.p1) with a primary assembly and 7 alternative reference loci. PAR is the pseudo-autosomal region. GRCh38.p13, which is the newest patch of the GRCh38 build has 35 alternative reference loci[52]. Figure taken from[54].

1.4.5 Read alignment

A wide selection of read alignment algorithms exist. These can be divided into groups depending on how a specific read is allocated the correct position in the reference genome. Two of the most widely used are: hash-based algorithms and algorithms based on the Burrows-Wheeler transformation (BWT)[55, 56, 57].

The hashing method works by dividing the reference genome into small chunks of a fixed length (k-mers) and then using these k-mers as keys in a large dictionary that links a k-mer to its position in the genome. The read sequences can now be compared to the large list of k-mers in the dictionary. If a read contains a specific k-mer, the data structure can then be used to find a probable position in the genome for this read. The k-mers in the dictionary are hashed and as hash lookup is extremely fast this method often outperforms simpler algorithms, such as sorting based ones. After the k-mer lookup returns probable positions where the input read map to the reference genome, slower but more precise alignment algorithms such as Needleman-Wunsch or Smith-Waterman can be used to find the best alignment and map the read to a specific position in the reference genome. One drawback

is that it requires a lot of memory to create this large k-mer dictionary[56, 58, 57]. Two examples of hash-based algorithms are SMALT[59] and Novoalign[60]. The latter can report multiple alignments per read, which (as is described in section 1.5.3) can be an advantage when aligning reads to the HLA region[58].
Another prominent approach to alignment is to use the BWT of the reference DNA. BWT is a reversible transformation that compresses a datafile. A transformed file can be indexed using Ferragina Manzini (FM) indexing which can be used to find matches to a specific k-mer requiring relatively little memory and time[55, 61]. Prominent aligners using this approach include the Burrows-Wheeler aligner (BWA)[62], Bowtie[63], Bowtie2[64] and SOAP2[65].

1.4.6 Graph-based alignment

Section 1.4.4 described that a newer version of the human reference genome includes several alternative DNA segments representing the genetic variation in the human population. An alternative to this approach is to abandon the linear reference and instead, represent the human genome using a graph-based structure, as shown in figures 1.4.6 and 1.4.7. In a graph genome, each variant is represented with an edge in the structure creating an enormous amount of valid "possible paths" through the genome. An area where this is especially advantageous is in highly polymorphic regions. Linear references focus on the one most prevalent version of a gene along with some frequently found variations. Graph-based can more efficiently store a larger number of variations and thereby increase the number of reads that align to the reference. Another advantage of using a graph genome, as opposed to a linear, is improved representation of ethnic diversity. African genomes are especially diverse creates difficulties for methods using linear reference[66]. Several types of graphs exist - examples are De-Bruijn graphs and Enredo graphs. These will not be described in detail here, but further information can be found in[67].
One drawback of graph genomes is that the alignment to the reference becomes more complex. Many available alignment tools only work with a linear reference but some graph aligners do exist. One of the newest graph-based alignment algorithms is called HISAT2. This tool uses an extended BWT for graphs[68] and an extended FM indexed called hierarchical Graph FM index (HGFM).

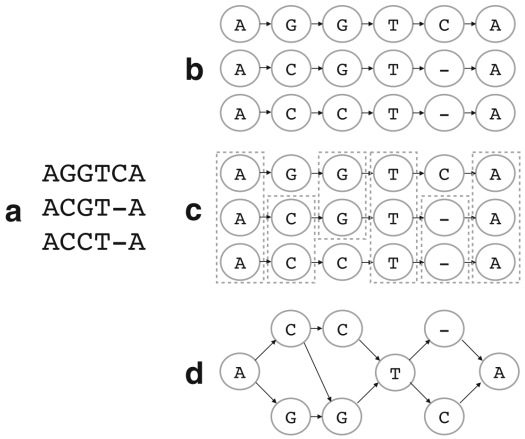

Figure 1.4.6: Construction of a graph-based reference. **a:** A simple multiple sequence alignment (MSA) with three sequences. **b:** The sequences are shown as a series of nodes with directed edges. **c:** Duplicates are marked for merging. **d:** The three differing sequences are now represented in one graph structure. Figure taken from [69].

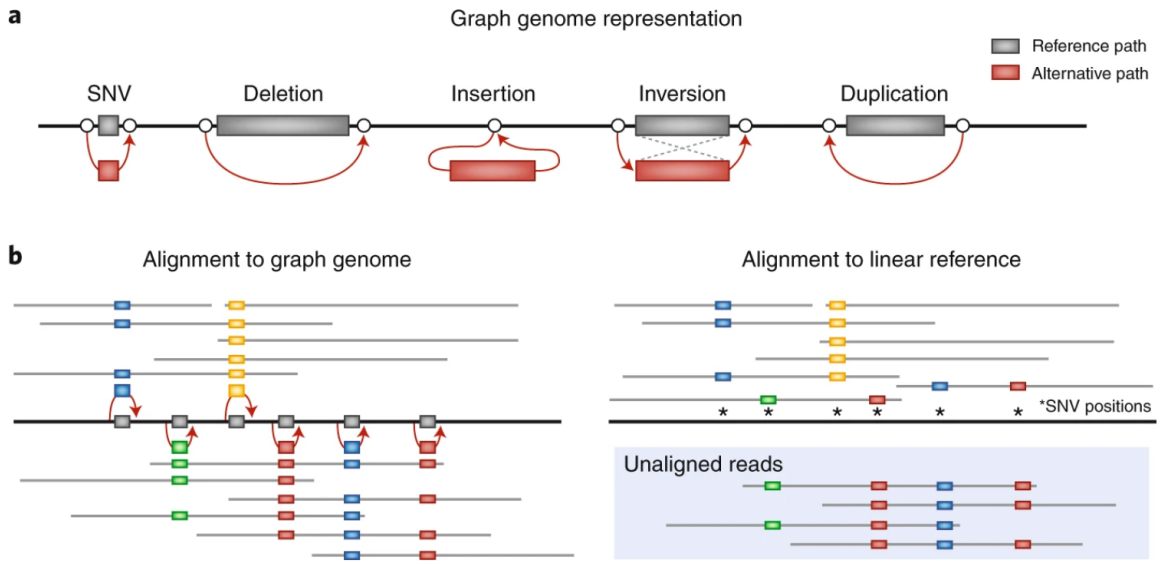

Figure 1.4.7: **a**: Genetic variation is more easily represented in a graph genome using small alternative paths - here shown using red arrows. **b:** A simplicstic illustration of alignment to a graph genome and a linear reference. The sample is heterozygous and has several SNPs at the locus. The complexity is here shown to be best represented using the graph-based approach. Figure taken from [66]

1.4.7 NGS data formats

The raw data in an NGS workflow often consist of FASTQ files. Each entry in a FASTQ file consists of four lines, as shown in figure 1.4.8. The first line is the identifier. This line

always starts with an "@" and contains information, such as the sequencing technology used, run id, whether the read is a member of a pair etc. The second line is the nucleotide sequence, the third is simply a "+" and the fourth line is the quality scores (Phred quality scores) of the individual nucleotides in the second line[70]. Phred is a system showing the probability of an error with a single symbol. For instance, in the ASCII base 33 system, which is commonly used, an error probability of 10% is denoted "+" and an error probability of 0.1% is denoted "?"[71]. The FASTQ entries of an analysis usually go through pre-processing. This could be removing of sequenced adaptors and low quality reads/parts of reads, but also more advanced techniques, such as identifying whether an individual single read/part of read is likely to be erroneous given the information of the rest of the reads (K-mer correction)[72].

```
@HWI-ST193:542:C2H0GACXX:8:1101:4404:2179 1:Y:0:ACACGA
ATGCNTTTTATAATCAAAAGCGAAGACCTAGCAGGAGGTTAAAAACCTTT
+
<<<<#2<@@5:9@44:@@?4(-8@(<9@<<658.==5=0<>??????9??
```

Figure 1.4.8: An example of a fastq entry from an Illumina sequence system. Figure taken from [70]

Given that a FASTQ file mostly contains a list of individual reads, the file is not directly useful for answering biological questions, such as variant calling and feature counts. Instead, the reads are ordered and mapped to a reference genome in a SAM or BAM file. SAM means Sequence Alignment/Map format while BAM is the binary version of a SAM file and contains the same information. A SAM/BAM file contains all the information from the original FASTQ files and conversion is reversible. In addition to the information found in the FASTQ file, SAM/BAM files contain the reference genome and information, such as a note of the alignment tool which was used. A full description can be found in [70]. BAM files often require too much storage space and alignment files are therefore alternatively stored as CRAM files. These are reduced versions of the alignment where only the differences between the aligned sequence and a reference genome are stored. Since the typical difference between two human genomes is very small, this gives a significant reduction in file size. One drawback is that to be converted back to e.g. BAM format, the original reference sequence used in the alignment is needed as a separate file[73, 74].

If only parts of the alignment are needed - for example, the section of chromosome 6 containing the HLA region - the reads mapping to this region can be obtained using an extraction tool (such as bedtools[75]) and a BED file. A BED file could contain the entry "chr6 28510120 33480577" and, if used for extraction, this means that the DNA sequence between coordinates 28510120 and 33480577 in chromosome 6 should be extracted[70].

1.5 HLA Typing

Variant calling is the process of determining sequence variants in an analysed sample[76]. This is a well-described process and a simplified example is shown in figure 1.5.1. Due to the unusually high degree of polymorphism in the HLA region, traditional variant calling methods for NGS data perform poorly in this region. Accurate variant calling in the HLA region (HLA typing) therefore currently requires variant calling algorithms specifically designed to the region[77]. This section introduces some of these algorithms, as well as some of the challenges in the field. The specific HLA typing algorithms, which are benchmarked in this project, are described in section 2.4.

HLA typing, for the most part, focuses on the six classic HLA genes (HLA-A, -B, -C, -DRB1, -DQB1, -DPB1). These genes both have the highest degree of polymorphism (most registered alleles) and are the most clinically relevant. Other HLA genes are, however still studied and some HLA typing methods do perform typing for non-classical alleles (such as HLA-E, -F and -G) in addition to the classical HLA genes[6, 78]. An overview of the number of alleles for the HLA genes is shown in appendix A.

Figure 1.5.1: A standard approach for variant calling. After reads (light green) are aligned to a reference genome (blue) a sequence alignment can be generated (dark green). Differences between the base calls in the reads (red) are sometimes due to errors, but if many reads differ from the reference genome on a position, it is likely to be a position, where the sample differs from the reference. Figure taken from [79]

1.5.1 Traditional HLA typing

One of the earliest HLA typing methods is serology-based HLA typing. One form of this method relies on mixing specific antibodies and cells with unknown HLA phenotypes. Cells that bind to the antibody are eventually killed and detected by a dye, while cells which are not bound to the antibody survive. By using antibodies to the HLA-A*02 antigen cells with the HLA-A*02 allele can be distinguished from those without. The method as such only gives HLA typing in 1-field resolution. Furthermore, the individual experiment only answers the question, ”Do these cells have this allele or not?” instead of answering ”Which specific HLA profile do this/these cells have?”[80, 81].

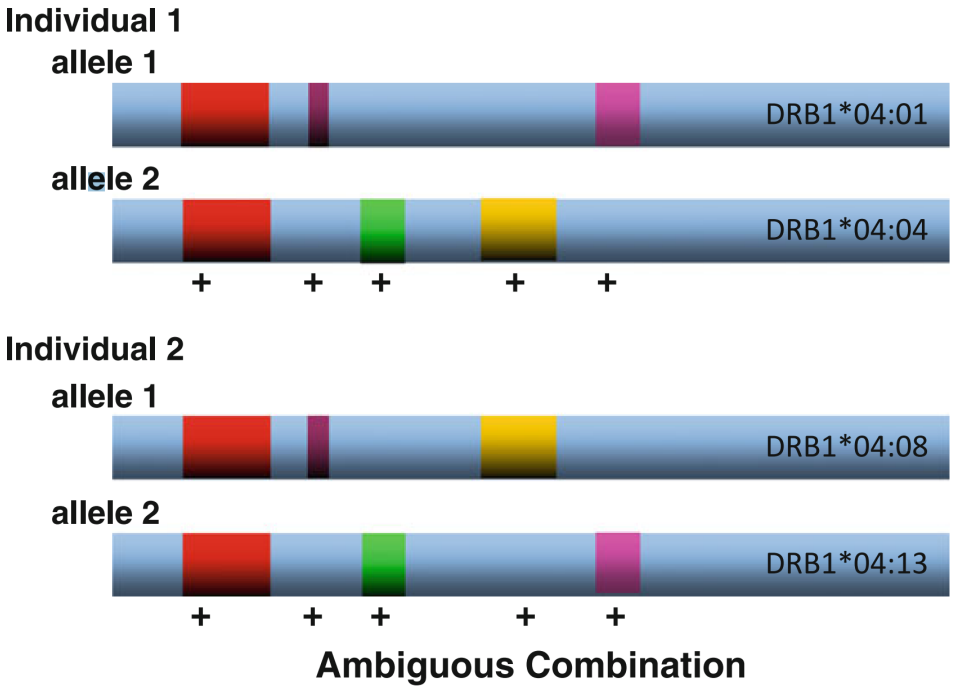

Figure 1.5.2: The two heterozygous individuals have four different HLA-DRB1 alleles between them. The polymorphic positions are distinguished by the coloured blocks. SSOP, SSP and SBT are unable to distinguish the two individuals from each other. These methods only provide the information that all the polymorphic sites (assigned ”+”) are present in the individual. Additional methods are therefore needed to find out which polymorphic sites belong to each haplotype. The figure is taken from [82]

More recent examples of conventional HLA typing methods use PCR to amplify the exons coding for the ARDs in HLA genes. After amplification, specific HLA alleles can then be identified through hybridisation with either sequence-specific oligonucleotide probes

(SSOP) or sequence-specific primers (SSP). PCR can also be followed by Sanger sequencing of the amplified regions and the sequences can be compared against HLA reference sequences. This last method, called sequence-based typing (SBT), gives very accurate results and is currently used as the gold standard for HLA typing[80, 82, 83].
The three methods provide 2-field resolution but need to be preceded by serology-based HLA typing. They are, therefore, very labour intensive, time-consuming and expensive[84]. Another problem, shared by all three methods, is a difficulty with resolving phasing ambiguities for HLA alleles[82], as illustrated in figure 1.5.2.

1.5.2 HLA typing using NGS data

The rapid development of NGS has made it affordable to sequence the full genome of individuals. In the later years, this has shifted the focus of HLA typing towards the use of data from the whole exome or genome (WES or WGS), instead of data from traditional HLA typing methods that solely focus on the HLA region and have a laborious HLA enrichment step.
WGS and WES produce sequencing data which are not restricted to the one or two exons coding for the ARD. This means, that the sequences of all exons (WES and WGS) as well as the introns the untranslated regions (only WGS) can be found. In the situation shown in figure 1.5.2, the additional information also means that the ”barcode” (the ”+” symbols) for an allele combination becomes much more complex and, in most cases also unique. These newer NGS based workflows therefore often solve the problem of phasing ambiguity[46, 47].
Another - perhaps more important - advantage to using WES or WGS, as opposed to data solely generated for HLA typing, is that HLA typing can be integrated into a general genetic analysis. WGS/WES data are generated for multiple purposes and in many cases, the data is already available. Most other genes can be typed from WGS or WES data by simply using a variant calling tool. HLA typing from NGS data harbours separate challenge as universal variant calling tools fail to accurately predict the correct allele combination in the polymorphic HLA region. Instead, the typing in this region is done using specialised HLA variant calling tools/HLA typing algorithms. One of the first of such algorithms was HLAminer from 2012[85] and new algorithms, which outperform

existing ones, are still being published[46, 80, 86].

NGS based HLA typing can be performed using WGS and WES data but also whole-transcriptome sequencing (RNA-seq) data by referring HLA genotypes from the sequenced transcriptome data. A generalised workflow for HLA typing using WGS or WES data is shown in figure 1.5.3. Since the data is not made from an HLA targeting approach, a majority of the reads are not from HLA genes. Removing reads, that do not map to the HLA region before typing makes files smaller the typing problem simpler. HLA typing algorithms therefore often begin with a filtering/read extraction step as also shown in figure 1.5.3[46, 80].

1.5.3 General trends in NGS based HLA typing algorithms

Based on the typing step (**C**) in figure 1.5.3, HLA typing algorithms can roughly be divided into two groups: methods using *de novo* assembly-based methods and methods which directly align to a reference. The alignment-based methods can further be divided into methods which use a linear reference and methods which use a graph-based one[46]. Methods using a *de novo* assembly approach first finds a consensus sequence for the HLA genes from the input reads without using a reference sequence. Only after reads have been assembled to consensus sequences are they matched with reference HLA sequences for allele assignment. Three algorithms which use this approach are HLAminer[85], ATHLATES[87] and HLAreporter[88]. HLAminer uses a k-mer table of the IPD-IMGT/HLA database to generate seeds used as starting points for subsequent assembly. ATHLATES works similarly, but only extracts exonic sequences, while HLAreporter uses the assembly tool TASR.

Algorithms using a *de novo* assembly approach still need an alignment step with a comparison to HLA reference sequences. This reference can be the IPD-IMGT/HLA database. NGS based HLA typing algorithms have used a number of the different alignment tools described in section 1.4.5 and 1.4.6 to perform this step[46].

Many of the HLA typing algorithms use BWT-Based mapping algorithms. HLAreporter, ALPHLARD-NT[89] and OncoHLA[86] use BWA-MEM and STC-seq[84] (described further in section 2.4.5) uses Bowtie. The two graph-based algorithms Kourami [69] and HLA-LA[78] (described in section 2.4.2 and 2.4.3) both use BWA in an early, linear

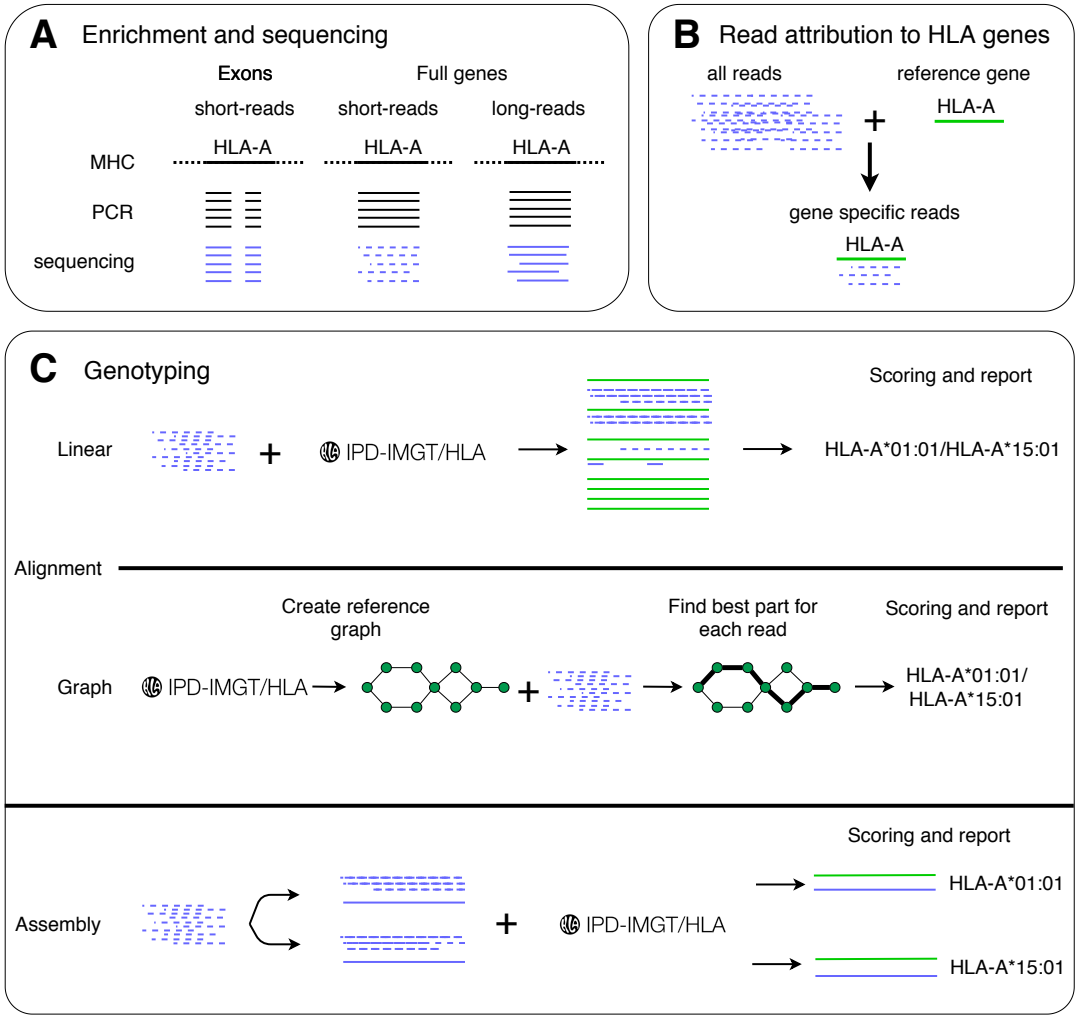

Figure 1.5.3: A generalised overview of HLA typing with WGS or WES data and a 2nd generation sequencing method, which has an amplification step. **A:** Target enrichment is used to capture the exons (only WES) and PCR is used for amplification. Long PCR products are fragmented for reading by short-read technologies. The original sequence and the output of the PCR are black and sequenced reads are blue **B:** Reads not specific to the HLA genes are removed. The reference sequences are shown in green. **C:** Generalised workflow for linear alignment-based algorithms (**C** top) graph-based algorithms (**C** middle) and assembly-based algorithms (**C** bottom). HLA typing workflows often share the first steps (**A** and **B**), but differ on the typing step (**C**). Figure is made with inspiration from [46] using diagrams.net[18]

alignment step before moving on to using a graph reference. One last example of an HLA typing algorithm using BWT is HISAT-genotype[66] (described in section 2.4.4). This tool is an extension of the graph-based aligner HISAT2 (see section 1.4.6).
Other mapping tools used by HLA typing algorithms include Novoalign (used by Polysolver[90] and PHLAT[91]) and RazerS3 (which is used by Optitype[92]). Optitype is described in

section 2.4.1.

The last step in the HLA typing process is to determine which HLA alleles best explain the reads which are now organised using assembly and/or alignment. This inference is shown in figure 1.5.3.**C**).

HLA typing tools employ various methods to infer HLA alleles, but most use some version of the IPD-IMGT/HLA database and try to identify the alleles from this database that best explain the found NGS reads. The tools PHLAT[91] and Polysolver[90] use a Bayesian approach, ATHLATES[87] identifies alleles based on their Hamming distance, while Optitype[92] uses an allele scoring matrix and integer linear programming to find the optimal allele combination explaining the maximum number of reads[46, 80]. The graph-based alignment used in HLA*LA, Kourami and HISAT-genotype means that HLA inference takes another form in these tools. Instead of finding the best alignment to linear references, inference of the set of HLA alleles is done by finding the most probable set of paths through a graph structure. This is explained further in sections 2.4.2, 2.4.3 and 2.4.4) [46, 69, 78, 93].

Individual alleles do not appear at a similar frequency but vary greatly both within a population and between ethnic groups. The website Allelefrequencies.net[94] provides information of allele frequencies from different worldwide population studies. This information is being used by typing algorithms in various ways to give more qualified predictions. Optitype uses a simple approach which ignores all alleles in the IPD-IMGT/HLA database that do not have a reported frequency on Allelefrequencies.net[46].

1.5.4 Challenges of NGS based HLA typing

The previous sections in this chapter have described some general trends and some individual differences of HLA typing and NGS based HLA typing algorithms. This section will outline some of the most important challenges and pitfalls of NGS based HLA typing and how algorithms are constructed to address these.

The HLA region is - as earlier mentioned - notoriously difficult to type. This is because of the high degree of variation in the HLA region and the high degree of homology between both different alleles and alleles of different genes. Functional alleles and null alleles can differ on as little as a 6C stretch (B*51:01:01:01) versus a 7C stretch (B*51:11N) and

a single base can be the only difference between two exons belonging to different HLA genes. Furthermore, several pseudogenes exist in this region which typing algorithms might mistake for real genes, leading to mistyping.
Section 1.5.2 mentioned how NGS based typing helped solve the heterozygosity problem. However, when polymorphisms are separated by long (>1000bp) homozygous regions, resolving ambiguity is still a problem for short-read (2nd generation) NGS based methods. These stretches are especially common in HLA class II genes and can explain part of why class II genes generally are more difficult to type than class I genes. Optitype and Polysolver only offer typing for HLA class I genes[46, 90, 92].

The sequencing of a sample using NGS is not a perfect process. As is also explained in section 1.4.2, NGS data does not always accurately reflect the original sample. One reason for this is PCR amplification, which is a biased step. When an allele is not amplified (ADO), or when an extra repeat is added to an STR region, the HLA typing algorithms have difficulties in accurately typing based solely on the low quality of the data. Some measures can be used to detect faulty reads, but since there are a huge amount of HLA alleles, which sometimes differ very little from each other, these measures are not always enough[46].

When comparing HLA typing algorithms it is important to note that typing can be performed on different resolutions, with 2-field and G group being some of the most common. The performance of an HLA typing tool is therefore not only determined by how often the predictions are correct, but also the resolution of the predictions. It is, of course, easier to achieve higher performance if results are only reported in 1-field resolution. Therefore, comparing the accuracy of different HLA typing tools is not always straightforward. Many tools only provide 2-field resolution as this is sufficient for immunotherapy pipelines, such as the one seen in figure 1.3.3[46].
HLA typing using SBT (section 1.5.1) has been used to discover many new HLA alleles and to construct the IPD-IMGT/HLA database, which many tools use as a reference. The method however only provides the sequences for the ARD coding exons and a significant number of alleles in the database are therefore missing introns and non-ARD coding exons. One consequence of this is that more reads map to the alleles with a full sequence

as the reference sequence is longer. An algorithm that naively assumes the HLA allele explaining the most reads is the correct one is therefore biased towards the alleles with a fully registered sequence.
Some tools assess this problem by replacing missing sequences with the most similar genomic sequences, effectively giving a qualified guess of the missing information. Others penalise mismatches in the ARD coding region higher than in other exons or simply choose to ignore all partially known alleles. A fourth option is to only focus on the ARD coding exons and disregard other parts of the sequence in the inference step, as is done by Optitype, HLA*LA and Kourami. One drawback of this choice is that results at best can be reported in G-group resolution. Typing in 3-field resolution requires knowledge about the ignored additional exons and 4-field resolution additionally needs information about the ignored introns[46].

The IPD-IMGT/HLA database is not only incomplete in regards to partly missing sequences, the rapid growth in newly found alleles (see figure A.0.2 in appendix A) also indicates that many HLA alleles are still undiscovered. These alleles might be rare as they have avoided discovery so far, but HLA typing tools that naively align to the IPD-IMGT/HLA database - assuming it complete - will always mistype these undiscovered alleles. Kourami and HISAT-genotype are two algorithms which can discover new alleles based on input data. In Kourami's case, this is done by allowing penalised extensions of the graph structure[46, 69].

1.6 Objectives of this project

Section 1.5 mentions some of the many NGS based HLA typing tools available, but as noted in a 2019 review of NGS based HLA typing: "... many of these tools are proofs-of-concept only, with few systematic studies on accuracy and use-cases being carried out apart from benchmarking the new tools."[46]. When the benchmarking of new tools is done by the developers of the tools, the estimated performance could be biased.

This project was designed by Evaxion Biotech, a company focusing on AI-driven immunotherapy. An overview of their neoepitope prediction pipeline is shown in figure 1.6.1. Evaxion uses WES data with a coverage of at least 250X from both healthy and tumour cells to identify somatic mutations. HLA typing is performed in a separate step. This pipeline can be optimised as the WES data itself is enough for HLA typing (as explained in section 1.5).

The primary aim of this project is to find the HLA typing tool that is best suited for Evaxion's purposes. Specifically, the goal is to find the tool with the highest typing accuracy for WES data (typing resolution is discussed in section 2.3.1). Another aim is to curate a gold standard dataset with WES data and corresponding HLA alleles. This dataset can be used to benchmark the HLA typing tools included in this project, but also newly developed tools in the future. The next chapter (2) describes the details of how these steps of this project were carried out.

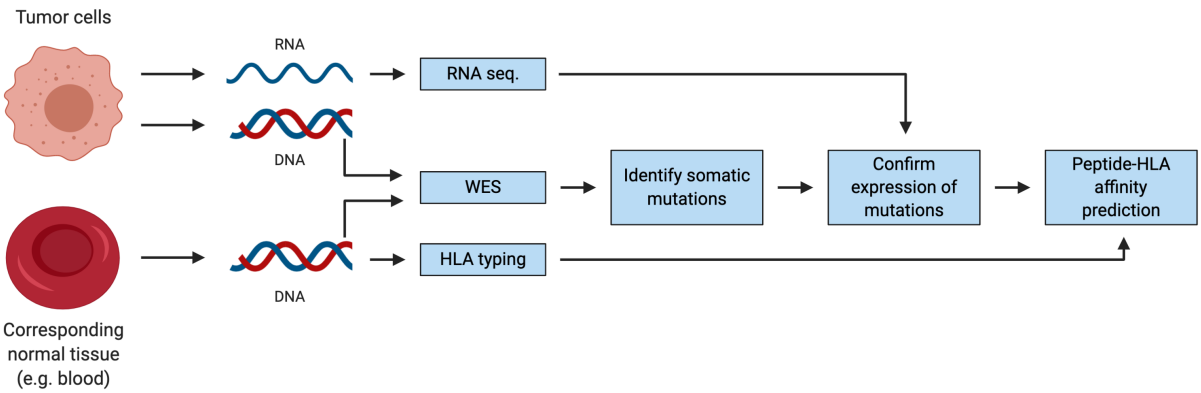

Figure 1.6.1: Evaxion provides personalised neoepitope prediction from WES data (tumour and healthy tissue) and from RNA seq data (tumour tissue). HLA typing is carried out separately, even though it could be done using the available WES data. Figure taken from [31]

CHAPTER 2

Methodology

This chapter contains a description of how the benchmarking of five HLA typing tools was conducted. The first section describes the selection criteria for including HLA typing algorithms in the benchmarking study. This is followed by a description of the gold standard dataset used in the analysis and description of how specifically the performances of the tools were evaluated. Finally, this chapter contains an outline of the central algorithms of the five selected HLA typing tools and a reasoning for their inclusion in this study.

All relevant code from this project can be found GitHub: https://github.com/nikolasthuesen/hlatyping. This includes the bash scripts that were used to run the HLA typing programs. These have full lists of the software dependencies of each HLA typing tool. Further examples of code found on the GitHub are workflow management scripts (Snakemake) and Jupyter Notebooks used in result processing and plotting.

2.1 Selection of HLA typing tools

There are numerous NGS based HLA typing tools available. A number of these were chosen for the benchmarking study according to a list of strict selection criteria that the tools need to fulfil and a list of beneficial traits favoured some tools over others. Both lists are shown below:

Selection Criteria:

- The tool must be able to work with WES data
- The tool must function on computerome 2.0 (discussed in section 2.1.1)

Beneficial traits:

- Good performance (typing accuracy) in previous benchmarking studies
- Non-restricted use (discussed in section 2.1.2)
- Not too computationally demanding (Memory use, CPU time)
- Interesting or unique features setting the tool apart from others.

Evaxion uses WES data and utilises HLA allele information for the HLA class I genes: HLA-A, -B and -C, and the HLA class II gene: HLA-DRB1. Tools that only accept transcriptome data (RNA-seq) as input data e.g. HLAProfiler[95], arcasHLA[96] and seq2hla[9] were therefore disregarded and the four genes relevant to Evaxion's pipeline were especially relevant when assessing a tool's performance.
Articles introducing NGS based HLA typing tools often include a proof-of-concept study, whereby the tool's performance is proven and compared to the performance of other available tools. As only a subset of the tools have been included in independent benchmarking analyses, these proof-of-concept studies are often the best source available for determining whether or not a tool has a "Good performance in previous benchmarking studies".
Table 2.1.1 shows the five HLA typing tools selected for this benchmarking study. Of these five, only Optitype has previously been included in peer-reviewed independent benchmarking studies[80, 97]. The conclusions of these studies are included in section 2.4.1. Outlines

of algorithms and a brief justification for their inclusion are found in section 2.4.

Table 2.1.1: The highest offered typing resolution and main approach for HLA typing for the five NGS based HLA typign algorithms studied in this project.

Tool Name	**Resolution**	**Original article**	**Approach**
Optitype	2-field	Szolek et al. 2014 [92]	Integer linear programming to find the allele combination that explains the highest number of reads.
HLA*LA	G group	Dilthey et al. 2019 [78]	Linear alignments projected on a population reference graph. Likelihood functions to infer the HLA type.
Kourami	G group	Lee et al. 2018 [69]	Weighted graph structure from alignment of input reads aligned to reference sequences. Most probable graph path is the inferred type.
HISAT-genotype	4-field	Kim et al. 2019 [93]	Graph-based alignment (HISAT2) and an expectation maximisation algorithm.
STC-seq	3-field	Jiao et al. 2018 [84]	Dense chip-based probes that capture the coding regions of HLA. Linear alignment algorithm.

2.1.1 Computerome 2.0

The benchmarking analysis was carried out using the Danish National Supercomputer for Life Science (Computerome 2.0, https://www.computerome.dk/) which is located at DTU Risø. Some of the HLA typing tools on the market do not function on a standard laptop but Computerome 2.0 offers the computational resources, which are needed to run the heaviest tools. An additional benefit is that the supercomputer is used by bioinformaticians at both DTU and Evaxion. One drawback is that since many users use the system, it

is not always possible to run programs without waiting time - especially programs that require a lot of memory or CPU time. Hence why it is an advantage for Evaxion if the HLA typing tool is not too computationally heavy. Another drawback of Computerome is that it hinders the use of Docker (https://www.docker.com/products/docker-desktop) which excludes the tool xHLA from this benchmarking study, as xHLA is installed using Docker[98, 99]

2.1.2 Availability of HLA typing tools

HLA typing tools are developed by both research groups and companies and can be split into three categories based on availability: commercial, public and academic non-commercial research purposes only[47].
The first category - commercial HLA typing tools - are developed by software companies that often sell software licenses for these programs. Two examples are NGSengine developed by GenDx (https://www.gendx.com/product_line/ngsengine/) and HLA Twin developed by Omixon (https://www.omixon.com/products/hla-twin/).
This project focuses on publicly available tools, which Evaxion and the public can use freely without the need to purchase a license. The reason that ”non-restricted use” is deemed a beneficial trait, instead of a hard selection criterion, is that some of the tools within the third category (academic, non-commercial) have interesting properties that justify inclusion in the benchmarking study - although they are not potential candidates for use by Evaxion.
Examples of tools, that only are available for non-commercial research purposes are ATHLATES[87], PHLAT[91] and HLAVBSeq v2[100].

2.2 Gold standard dataset

The gold standard dataset used to benchmark the five HLA typing tools consists of WES sequences from 829 individuals and corresponding HLA alleles for the genes: HLA-A, -B, -C, -DRB1 and -DQB1. The sequencing coverage ranges from 37X to 456X with the median being 86X. The sequencing was done as part of the 1000 genomes project and the validated HLA types for the samples are primarily taken from a 2014 study.

The 1000 Genomes Project was a major genome sequencing project running from 2008 to 2015 and had the goal to find and map genetic variation within the global human population. The final data set consisted of data from 2504 individuals distributed over 26 populations all sequenced using Illumina technology[45, 51, 101]. After the conclusion of the project, the International Genome Sample Resource (IGSR) was established to maintain and extend the 1000 genomes project data. The IGSR was as of 2017 the world's largest open collection of human variation data[102]. One of the aims of the ISGR is to expand the data further and since its launch, this been actively happening through both the incorporation of data from other databases and the addition of data from novel studies[103, 104]. The database currently contains many different types of sequence data, e.g. low/high coverage WGS, PCR-free, long read and WES.
The exome capturing technology used in the 1000 genomes project has varied from centre to centre - some examples include: NimbleGen's SeqCap EZ Human Exome Library v2/v3 and Agilent SureSelect All Exon V2[105]. A full description can be found at https://www.internationalgenome.org/.
The exome sequence files used in this project were downloaded in CRAM format from the 1000 genomes site. One CRAM file corresponds to one individual but sometimes to several FASTQ files, as an individual can be sequenced multiple times[106]. The CRAM files are aligned to the GRCh38 build of the human genome, more specifically to the genome found at: ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/GRCh38_reference_genome/GRCh38_full_analysis_set_plus_decoy_hla.fa. A full description of how the quality of the data was assured and how alignments were generated is found on the ISGR website[106]. A full list of URLs used to download CRAM files is found at this project's GitHub (https://github.com/nikolasthuesen/hlatyping).

A 2014 study published a dataset containing the HLA alleles (HLA-A, -B, -C, -DRB1 and DQB1) for 1267 of the individuals sequenced in the 1000 genomes project. The typing was performed using SBT (see section 1.5.1) and only focused on the ARD coding exons. Thus, polymorphisms outside exon 2 and 3 for HLA class I genes or outside exon 2 for HLA class II genes were not investigated. Due to this and SBT's typing ambiguity problem (see figure 1.5.2 in section 1.5.1), the dataset consists of ambiguous typing for a number of the alleles. The typing resolution is primarily 2-field, but in some cases 3-field[107].
A 2018 study reevaluated the results from 2014 and provided HLA types for 1,701 additional samples. The 2018 study did not use SBT, but instead the licensed HLA typing tool, PolyPheMe. 105 predictions were found to be discordant with the 2014 dataset and in these cases, the correct allele was found using sequence inspection and targeted resequencing[108].

The dataset used in this benchmarking study was constructed by taking the 2014 dataset and updating the HLA alleles found to be false in the 2018 study. The 2018 dataset is not used further as it was constructed using an in silico HLA typing algorithm. Not all the 1267 samples had corresponding WES files, and the final dataset therefore only consisted of CRAM files from 829 individuals.
During the initial assessment of the dataset, it was found to contain three deleted alleles that have since changed names. These are A*01:34N (shown to be expressed at low levels and renamed A*01:01:38L), A*03:260 (shown to be identical to A*03:284N) and C*03:99 (renamed C*01:169)[109]. For both the gold standard dataset and for the predictions made by the HLA typing tools, these three alleles were converted to their newest name.

2.3 Performance Evaluation

The five HLA typing tools were evaluated on several different parameters. The most important metric is perhaps the number of correct predictions a tool makes out of the total amount of HLA alleles - the typing accuracy. As explained in section 1.5.4, the typing accuracy can be assessed at different typing resolutions (discussed in section 2.3.1).
The benchmarking also included a downsampling analysis where a selection of WES samples was downsampled to lower coverages - ranging from 1X to 100X - and each tool's performance was evaluated at the varying sequencing depths (section 2.3.3).
The time and memory use of each HLA typing tool was noted in both the analysis using the full data files and the downsampling analysis (section 2.3.4).

2.3.1 Typing accuracy and conversion of typing resolutions

The typing accuracy of a tool is defined as the number of correctly predicted HLA alleles divided by the total number of HLA alleles. Some previous studies have disregarded alleles for which a tool did not return a prediction, e.g. if a tool returned a prediction for one out of ten samples but the one prediction was correct, the typing accuracy would be 100%[69]. This gives a higher accuracy on paper, but it does not truly reflect how well an HLA typing tool performs. In addition to the typing accuracy, the tool's call rate was also registered. This is the number of predictions of HLA alleles a tool makes divided by the total number of HLA alleles in the dataset. If a tool outputs a prediction for every single HLA allele, it, therefore, has a call rate of 100%.
It is easier to achieve a higher typing accuracy if you return predictions at 1-field resolution than at 4-field resolution (figure 2.3.2) and tools should therefore be compared at the same resolution. The gold standard dataset in this study provides the HLA alleles in (mostly) 2-field resolution which can unambiguously be converted to both P-group resolution and 1-field resolution. 2-field resolution cannot be unambiguously converted to G group resolution, as 2-field resolution separates different proteins while G groups separate alleles based on differences at a genomic level. Likewise, G group resolution cannot be unambiguously converted to 2-field resolution, as G group resolution solely refers to differences in the ARD coding exons while 2-field resolution addresses the full protein and therefore the full protein-coding sequence.

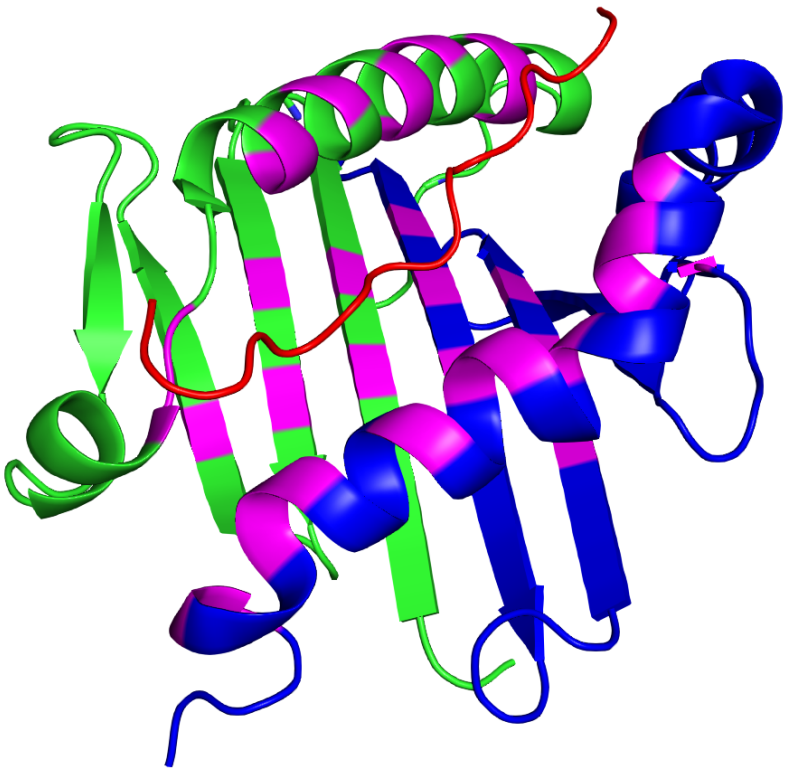

Figure 2.3.1: A binding pocket of a HLA-II molecule (see also figure 1.1.1). DRA is shown in green and DRB1 in blue. A melanoma antigen (red) is shown in the binding pocket and the residues that are important for predictions made by NetMHCIIpan are shown in pink. Protein data was found in the Protein Data Bank (PDB)[110, 111] and the figure was made using PyMOL[112]

G group and P group resolution were constructed to group alleles with identical ARDs, but the binding of a peptide does not necessarily require all the amino acids of the ARD. The tools NetMHCpan (MHC class I) and NetMHCIIpan (MHC class II), that both predict how well peptides bind to specific MHC alleles, only consider the amino acids that are directly involved in the binding of a peptide when making predictions[13, 113]. On paper, Evaxion uses 2-field resolution in their pipeline (figure 1.6.1, section 1.6), but in reality, the only amino acids that make a difference to the final prediction of the pipeline are those considered by NetMHCpan (HLA-A, -B and -C) and NetMHCIIpan (HLA-DRB1).
To best fit Evaxion's pipeline, this project introduces a new (unofficial) typing resolution - NetMHCpan resolution. This resolution groups proteins that share the amino acids directly involved in the antigen-binding but does not distinguish between those that differ at any other site.

In summary, the typing accuracy of each of the HLA typing tools is measured in 1-field resolution, NetMHCpan resolution, P group resolution and 2-field resolution.

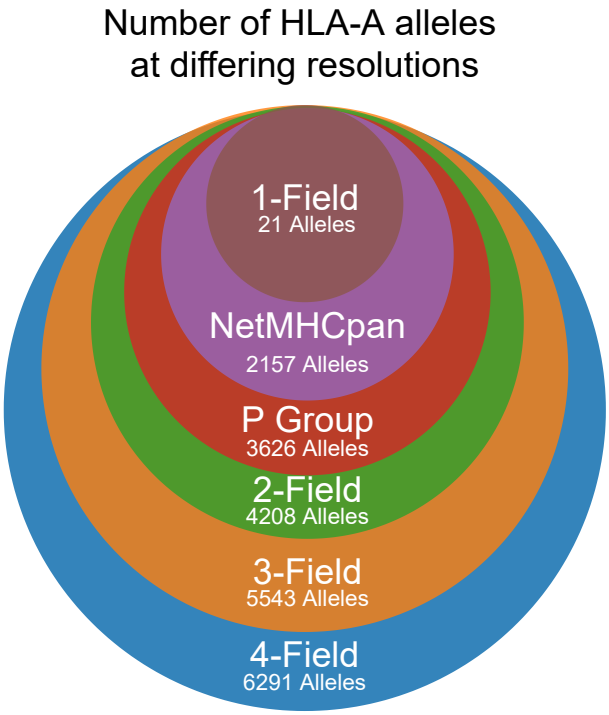

Figure 2.3.2: How many HLA-A alleles exist? The answer varies greatly depending on the typing resolution. The NetMHCpan resolution is not an official resolution, but is defined in this study. The statistics are taken from IPD-IMGT/HLAFigure made using diagrams.net[18]

Section 1.2.1 mentions that null alleles (non-expressed) are not included in P groups, even though they share an identical ARD with the alleles in a P group. This is also reflected in this study and predictions are counted as wrong if, e.g. a tool predicts A*01:01P and the correct allele is A*01:04N and vice versa.

Some HLA typing tools return more than one prediction of an HLA allele when, for example typing confidence is low[80]. Kourami's standard output is in G group resolution, but it sometimes outputs ambiguous allele predictions. An example prediction could be "B*08:01:01G" for one allele and "B*57:11;B*57:83" for the other. B*57:11 and B*57:83 only differ on one amino acid but are nevertheless different alleles in 2-field resolution. To maintain the one prediction to one allele set up, in the final analysis, the first Kourami predication was accepted and the additional prediction ignored. It was investigated whether including both predictions would make a large difference, but it was

found not to be the case. At 2-field resolution, allowing ambiguous typing only resulted in 7 additional correct allele predictions, improving Kourami's typing accuracy for genes HLA-A, -B, -C, -DRB1 and -DQB1 from 83.16% to 83.24% (see also figures 3.1.1 and 3.1.2)

The gold standard dataset also included some ambiguous allele calls, as explained in section 2.2. In these cases, a prediction was noted as "correct" if it matched any of the correct alleles listed in the gold standard dataset.

2.3.2 Ensemble approach

A 2017 study analysed platform-specific genotyping errors for NGS based HLA typing tools[114]. One finding of the study was that Optitype commonly miscalled HLA-A*26:01 to HLA-A*25:01, HLA-B*45:01 to HLA-B*44:15 and HLA-C*08:02 to HLA-C*05:01. One way of avoiding bias from platform-specific errors could be to use an ensemble approach i.e. taking allele predictions from multiple tools into account when making a final, overall prediction.
In this project, it was investigated whether an ensemble approach would outperform any of the tools. This was done in two groups of tools. The first group consisted of Kourami, HLA*LA, HISAT-genotype and Optitype and the second group consisted only of the graph-based tools: Kourami, HLA*LA and HISAT-genotype. The ensemble prediction was found by taking a simple majority vote for each sample. In the case of a tie, the lowest numerical allele was chosen, e.g. A*01:01 was chosen in a tie between A*01:01 and A*01:02. The lowest numerical name often means an earlier discovery which could be correlated with a higher allele frequency in the population. This is, however, only a hypothesis and tied predictions could perhaps be decided in a better way. These results are discussed in section 3.1.3.

2.3.3 Downsampling

230 out of the 829 samples in the gold standard dataset had a coverage of at least 100X. These samples were used in a downsampling study to investigate how the performance of each tool depended on the coverage of the input sample. The performance of the tools was examined at 1X, 2X, 5X, 10X, 20X, 50X, 75X and 100X. Downsampling was performed by first finding the full sequencing depth of the CRAM files used in the

gold standard dataset. This was done using *mosdepth* (version 0.2.6)[115]. Hereafter, the alignment files were downsampled using *samtools view -s* to achieve the range of needed coverages[116]. The downsampling scripts are on this project's GitHub (https://github.com/nikolasthuesen/hlatyping).

The downsampled files were used for HLA typing in a procedure parallel to the typing process in the main study (overview shown in figure 2.3.3 in section 2.3.5). The results of the downsampling analysis are shown in section 3.2.

2.3.4 Time- and memory usage

For each tool, the peak memory usage, the CPU time usage and the wall time were tracked for all samples in both the main benchmarking analysis and downsampling analysis. This was done to compare their efficiency and to give en estimate of the requirements for running each tool.

Specifically, this was carried out by running the *qstat* command at the end of individual job scripts used for each tool to type each sample.

All tools in the analysis were run using 10 threads and were given as much memory as they needed.

2.3.5 Overview of the typing workflow

The workflow of the preprocessing of data and which files were given to the HLA typing algorithms (not including the downsampling analysis) is shown in figure 2.3.3. In practice, the workflow was constructed using Snakemake[117] - the code is found on this project's GitHub (https://github.com/nikolasthuesen/hlatyping). The five HLA typing algorithms are described in section 2.4.

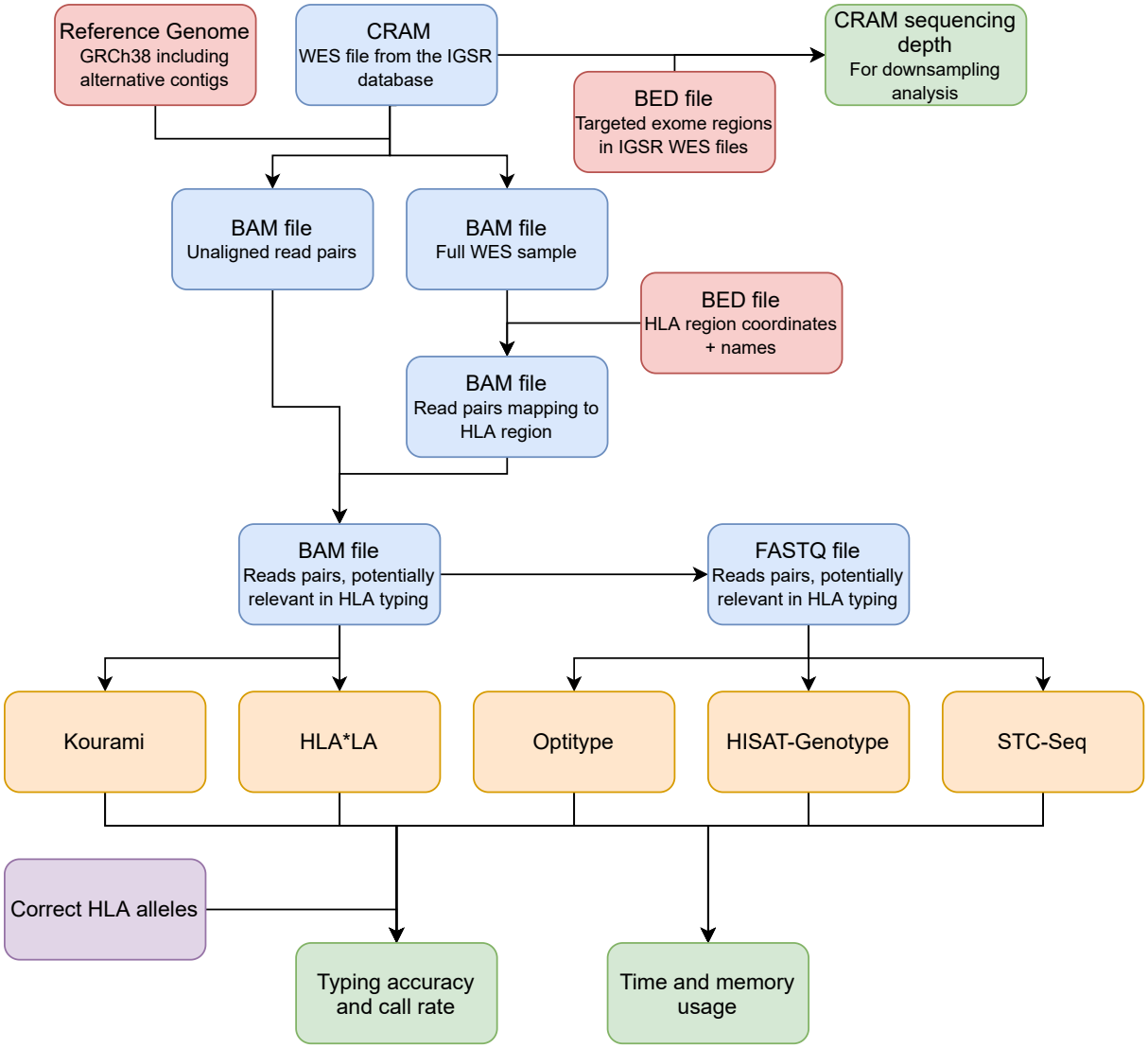

Figure 2.3.3: An overview of the benchmarking study. The input CRAM file from the IGSR database (gold standard WES samples) and files deriving from it are shown in blue. These are specific to the individual samples. Reference files which are identical for each sample are shown in red, the HLA typing tools in yellow, the correct HLA alleles for each sample is purple and the output files are all shown in green. For each sample (CRAM file), the sequencing depth is found (used in the downsampling analysis) and the HLA typing tools predict the HLA alleles. Predictions are compared to the correct alleles (see section 2.2) and the computational resources used are noted for each tool. In the downsampling analysis, the found sequencing depth for each CRAM file is used, but it is not the raw CRAM file that is downsampled. Instead, the reduced BAM file (with read pairs potentially relevant in HLA typing) is downsampled, thereby avoiding the unnecessary extraction of HLA reads in the downsampling analysis. Figure created using diagrams.net[18]

2.4 Outline of HLA typing algorithms

This section contains descriptions of each of the five NGS based HLA typing tools chosen for the benchmarking study, as well as some comments on why they were chosen for this project and how they were installed. These descriptions are made to give the reader an understanding of how the algorithms work and how they differ from each other but are not fully comprehensive. For a complete description of each tool, one should refer to the original articles.
All tools were run with default options.

2.4.1 Optitype

Optitype is a publicly available HLA typing algorithm from 2014. It is one of the earliest HLA typing tools which does not require HLA enriched NGS data solely generated for HLA typing and is still regularly updated. Optitype accepts both DNA (WGS and WES) and RNAseq data as input data in FASTQ format. Two disadvantages of Optitype are that the typing results are only reported in 2-field resolution and that it is only able to type HLA class I genes[47, 92].
There have been relatively few independent benchmarking studies of NGS based HLA typing algorithms, but Optitype is included in a significant amount of them. It has furthermore been one of the best-performing tools[80, 86, 118]. Previous benchmarking studies of Optitype have reported its accuracy for class I genes to be 97.2 %[114] and 98%[80] (both for WES samples at 2-field resolution).
Optitype was included in this project both because it has been shown to perform well and as it allows easier comparison between the results of this study and others that also assess Optitype's performance.

Optitype belongs to the group of HLA typing tools based on linear alignment (see figure 1.5.3). Figure 2.4.1 shows Optitype's typing pipeline. It is recommended to begin the process by filtering out the sequenced reads that do not map to the HLA region. This step is not considered part of Optitype, but the tool's GitHub both has an example of how to perform this extraction using the read mapper RazerS3[119] and has the relevant files which are needed[92, 120].

The first step in Optitype is an alignment of input reads to exons 2 and 3 (the ARD coding regions) of HLA class I genes and the intron sequences flanking these exons. For alleles where these intron sequences are not found in the IPD-IMGT/HLA database, the intron sequences of the most similar, fully sequenced HLA allele are used instead.
Aligned reads are then used to construct a binary hit matrix with aligned input reads as rows and candidate HLA alleles as columns. A hit is noted in all cases where a read maps to an allele. Here, the candidate HLA alleles are limited to non-rare alleles with a reported frequency (in 2-field resolution) on Allelefrequencies.net.
Optitype finally finds the HLA allele combination (3 classical and three non-classical class I genes) that maximises the number of reads potentially originating from the selected alleles. Specifically, this is done by combining the hit matrix as well as information about the goals and constraints in an integer linear program (ILP) and using an ILP solver. CPLEX 12.5 was used in the original article, but other solvers can be used as well[92].

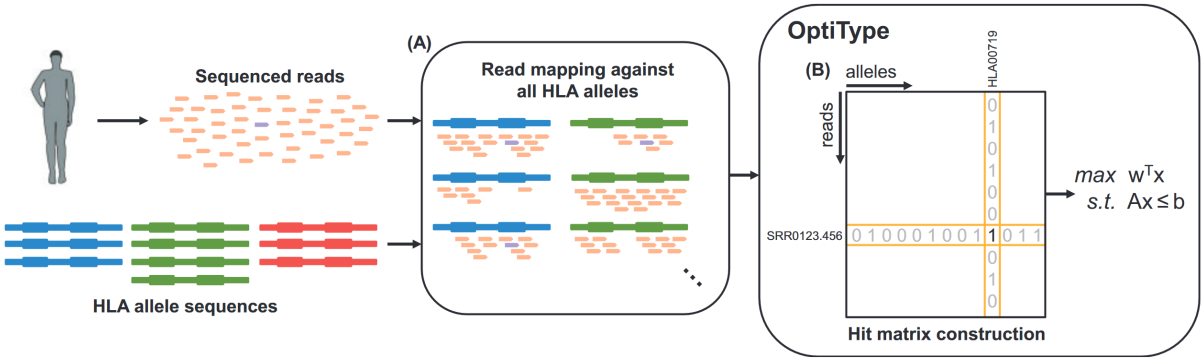

Figure 2.4.1: Optitype's HLA typing pipeline consists of three key steps. **A:** First, the sequenced reads are aligned to a database of HLA allele sequences (only exon 2 and 3). **B:** The alignments are then converted to a binary hit matrix (alleles as columns, reads as rows) where a hit is noted when an allele can explain a specific read. In the last step, the hit matrix is used to formulate an ILP in which the predicted HLA alleles are those that maximise the number of explained reads.

In this project, Optitype version 1.3.3 was installed using the guide on the GitHub[120] and an article from 2018[77]. CPLEX 12.7 was used in initial tests of Optitype but this ILP solver often did not converge to a solution. It was therefore replaced with cbc version 2.9.5[121] which was found to always converge at a solution.

2.4.2 Kourami

Kourami is a publicly available, graph-based HLA typing algorithm from 2018. The tool accepts WES and WGS data in BAM format and provides typing in G group resolution. A unique feature of Kourami is that it is not limited to infer HLA alleles based on a database. It can also be used to discover novel alleles. There are no independent benchmarking studies of Kourami's performance, but the tool is - in addition to its own article - included in the proof-of-concept studies contained in the two articles introducing HLA*LA and OncoHLA[78, 86]. Kourami performs as well as HLA*LA and OncoHLA for high coverage WGS data but is outperformed when the coverage drops and for WES data sets. This is not surprising, as the tool is designed for data that, when aligned, has few gaps resulting in a large, connected graph structure.

The article introducing Kourami reports that the tool achieved 100% typing accuracy (132/132 correctly predicted HLA alleles) for WGS data and a typing accuracy of 82.3% (284/345) for a WES data set[69].

Kourami (v0.9.6) was installed using the guide on GitHub[122] and a 2018 paper[123].

Kourami is graph-guided but is not an entirely graph-based algorithm. Instead, it is a hybrid method that uses both a linear alignment tool and a graph structure. An overview of the steps in the algorithm is shown in figure 2.4.2.

Kourami uses both the full-length and exon-only alleles from the IPD-IMGT/HLA database to construct the MSA in step **a** figure 2.4.2, to which the input reads are aligned in step **c**. These alignments are represented as paths through the graph in step **d** and read depths (the number of reads aligning to a sequence) are converted to weights on the individual edges in the graph. The weight on a particular edge is a measure of how many times this variation was found in the input data. This means that the edges with higher weights together make a path through the graph that matches the correct allele, as a high number of input reads will align to these edges. In areas where the input reads do not align perfectly to the MSA, Kourami allows for graph modification to reflect insertions, deletions or substitutions[69].

Kourami does HLA inference in step **e** by finding the two paths through the graph that best explain the weight *and* the phasing information in the input reading data. Solely relying on the number of reads (weight) would not account for the fact that the reads originate from two distinct and likely different alleles. Read pair information is therefore used to

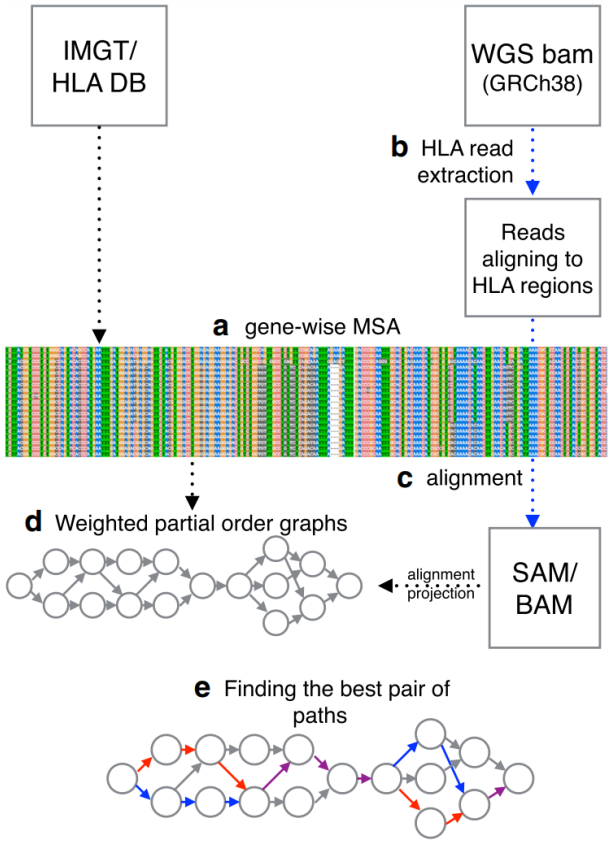

Figure 2.4.2: A simplified overview of the Kourami algorithm. **a:** First, an MSA is constructed based on the sequences from the IPD-IMGT/HLA database. **b:** Reads that could potentially be attributed to the HLA region are extracted from the input file, which here is a WGS, BAM file. **c:** These extracted reads are then realigned to the MSA constructed in part **a**. **d:** The graph structure (a modified partial-order graph) is constructed based on the MSA and the extracted and aligned input data. **e:** Finally, the HLA inference is done by finding the best pair of paths through the graph structure. These two paths are shown in red and blue and in purple when they overlap. Step **e** is further elaborated in figure 2.4.3. Figure taken from [69]

find out which variations (paths through the graph) belong together. In a heterozygous region, there are often homologous stretches where two paths in the graph structure merge and pass through the same nodes and edges. In regions where alleles differ, the paths split. Kourami calls these split regions *bubbles*. Specifically, a bubble consists of all edges in a stretch where the paths are separate and the two shared edges on each side of this stretch. Such a bubble is shown in figure 2.4.3. There are of course ideally only two plausible paths in a bubble, but sequencing errors and amplification of these can result in a bubble having three or more. The bubbles are found from the graph structure by selecting the pair (potentially identical in case of homozygosity) that maximise probability given the

input data (the weight on the edges). Subsequently, phasing information is used to connect bubble pairs and to merge bubbles from left to right until two, full, individual paths are found. The best pair of candidate HLA alleles are those that are best explained by the two bubble paths[69].

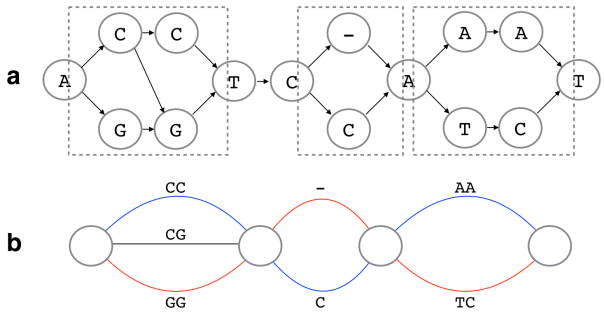

Figure 2.4.3: Kourami's bubbles consist of at least three columns of nodes, where the leftmost and rightmost node are shared between the two (or more) paths and the rest are not. This figure shows three bubbles, two of length 4 and one of length 3. In the first bubble, there are three alternative paths which can be due to a sequencing error. The paths through the bubbles here match variations between one allele (red) and another (blue) for a homozygous region. Figure is taken from[69].

2.4.3 HLA*LA

Like Kourami, HLA*LA is a hybrid tool. It seeks to combine the advantages of representing variation in a graph genome with the efficiency of linear alignment (the ”LA” in HLA*LA). HLA*LA accepts both short-read and long-read NGS data as BAM or as CRAM files and reports typing results in G group resolution. The tool was released in 2019 and has not yet been included in other peer-reviewed studies, other than its own proof-of-concept study. Here, it is reported to achieve a 99.4% accuracy on a WGS data set and a 93% accuracy on a WES dataset. The tool (v 1.0.1) was installed using the guide on the HLA*LA GitHub[124].

Briefly, HLA*LA constructs a linear alignment of input reads, projects the linear alignments onto a graph-genome reference, realigns after projection and finally, infers HLA alleles based on the graph alignment. Figure 2.4.4 shows an illustration of the workflow with a full explanation of the algorithm.

HLA*LA's graph-genome (a population reference graph - PRG) is constructed in similar fashion to what is shown in figure 1.4.6, section 1.4.6. This is completed before the typing process begins. The reference sequences used in the initial linear alignment are alleles with fully known sequences in the IPD-IMGT/HLA database and HLA sequences from the the the reference genome build GRCh38, including alternative loci. The graph is constructed based on the same reference sequences but also includes the HLA alleles without completely known sequences[78, 125].

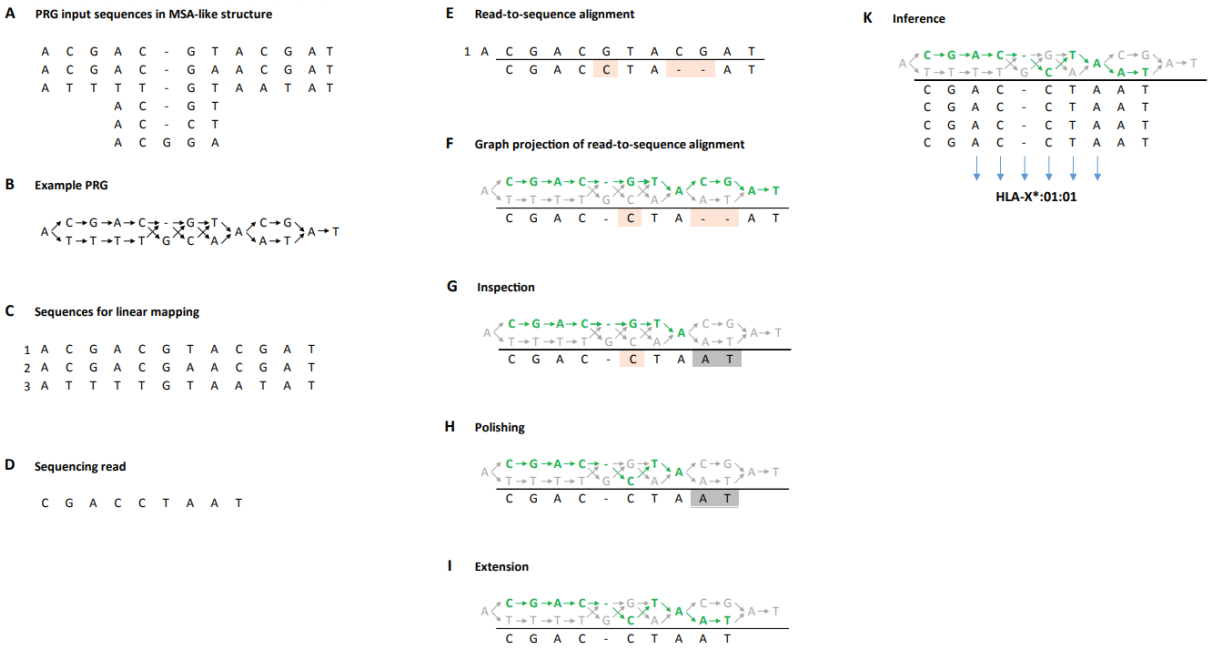

Figure 2.4.4: HLA*LA's workflow with labelled graphs. **A:** An MSA of the sequences used to construct PRG. This includes both fully and partially known alleles. **B:** The PRG is constructed from the MSA and is universal. This differs from the graph structure in Kourami, which is changed based on the input data. **C:** An MSA of the reference sequences used in the initial linear alignment. Only HLA alleles, for which the full sequence is known, are included here. **D:** Input sequences from the sample that HLA*LA is typing. **E:** Linear alignment of input read to the MSA shown in **C** using BWA-MEM. Mismatches are shown in red. **F:** The alignment is projected onto the PRG. The new graph alignment is shown in green. This is followed by a three-step optimisation of the graph alignment (**G-I**). **G:** The alignment is inspected and larger alignment regions, that potentially map to the wrong path in the graph structure, are removed. **H:** The remaining graph alignment is optimised by smaller local changes using dynamic programming. **I:** The part(s) of the alignment removed in step **G** are reintegrated into the alignment using a computationally expensive graph alignment method. **K:** Finally, a likelihood model is employed to find the allele combination (in G group resolution) that best fits the graph alignment. Figure taken from [78].

2.4.4 HISAT-genotype

HISAT-genotype is a graph-based HLA typing algorithm based on the graph alignment program, HISAT2 (hierarchical indexing for spliced alignment of transcripts). By using a graph-based aligner, HISAT-genotype differs from the hybrid methods Kourami and HLA*LA that use a graph reference and linear alignment. HISAT-genotype can - like Kourami - be used to discover new alleles and can provide typing results in up to 4-field resolution.

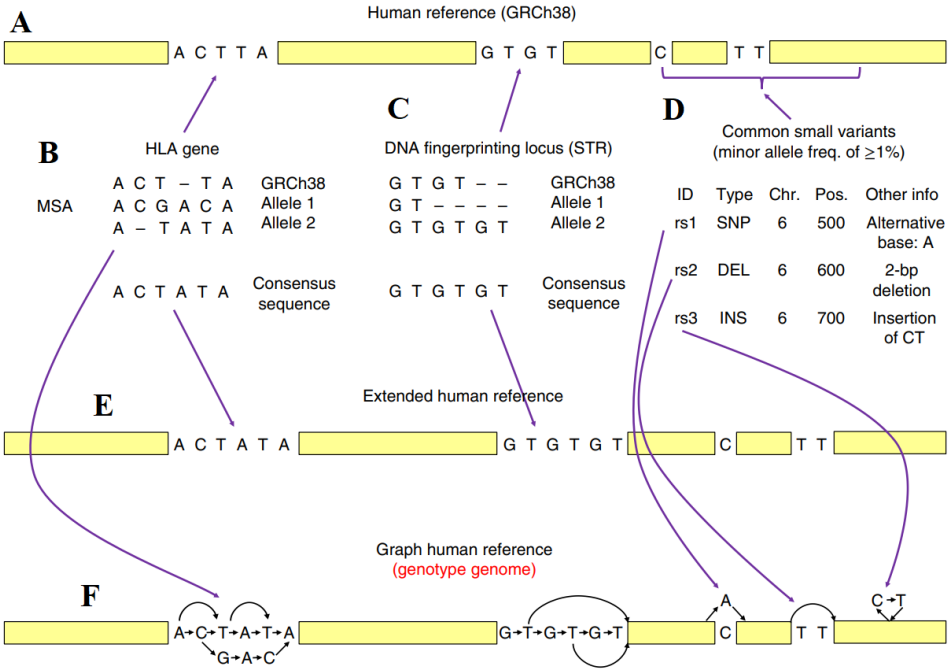

Figure 2.4.5: Construction of HISAT-genotype's graph reference. **A:** The GRCh38 human reference genome. **B:** An MSA of HLA allele sequences from the IPD-IMGT/HLA database and a consensus sequence made from the most frequent nucleotides in each position of the MSA. **C:** DNA fingerprinting loci (STRs) for HLA alleles taken from NIST STRBase. The longest allele (most STRs) is used as consensus sequence. **D:** Common small variant data (frequency over 1%) for HLA alleles taken from dbSNP. **E:** An extended version of the human reference genome is made by combining the consensus sequences shown in **B** and **C** with the GRCh38 human reference genome (**A**). The extended reference genome is used as a starting point to construct HISAT-genotype's graph reference (called genotype genome). Here, all variants from **B**, **C** and **D** are integrated. The constructed graph reference is not modified during typing, but the indexes in the graph are specific to the version of the human genome (here GRCh38) from which it was made. Figure modified from [93].

The HISAT2/HISAT-genotype article was published in 2019, but HISAT-genotype has received multiple updates since and is continuously being improved. Version 1.3.2 was used in this project. The code is available on GitHub[126] and was installed using the corresponding web-guide[127].

HISAT-genotype's typing process begins by using HISAT2 to produce an alignment of the input sequences to the reference graph genome (**F** in figure 2.4.5) and thereafter, extracts the reads mapping to the HLA region. Briefly, HISAT2 uses an extended BWT for graphs and a modified version of FM indexing (hierarchical graph FM indexing), developed with HISAT2. The developers of HISAT2 report that HISAT2 is approximately half as fast (half the number of read pairs processed per second) and uses about a third more memory than the linear aligner, BWA-mem[93].

The next step is a graph alignment of the extracted reads to the IPD-IMGT/HLA database. To address the many alleles that have missing sequences for introns and non-ARD coding exons, HISAT-genotype groups alleles that share identical ARD coding exons, finds the groups that best match the input reads and then uses information from both introns and exons to give the final prediction of HLA alleles. An expectation-maximisation function is used to both prioritise groups and for the final selection[93].

2.4.5 STC-seq

STC-Seq is an HLA typing method that combines an SSO like technique with an NGS and an in silico HLA typing pipeline. Briefly, it consists of initial HLA allele capture, PCR amplification and Illumina sequencing of captured HLA reads and finally, an HLA allele-calling algorithm. The HLA calling algorithm can however be used independently of the rest of the workflow, as it simply predicts HLA alleles from input NGS data. In this study, only the HLA calling algorithm is tested and it is provided the same (non-HLA enriched) input data as the rest of the benchmarked algorithms.

An overview of STC-Seq's HLA calling pipeline is shown in figure 2.4.6. The code and an installation manual are available on BioCode[128]. Version 1.0 was used in this project as it has not been updated since its release in 2017.

STC-Seq is based on data specifically made for HLA typing and is not designed to use WES or WGS data. STC-Seq's HLA calling algorithm is nevertheless included in the

benchmarking analysis to assess the effect of using general WES data, as opposed to HLA targeted data. The article describing STC-Seq reports that the tool was able to correctly predict 98% of alleles using a dataset with 382 high coverage samples.

The STC-Seq article states that the tool is exclusively for non-commercial use, but as STC-Seq is not a potential candidate for Evaxion, this is not a problem. The code is public and can be downloaded from the BioCode website[128] (https://bigd.big.ac.cn/biocode/tools/BT007068).

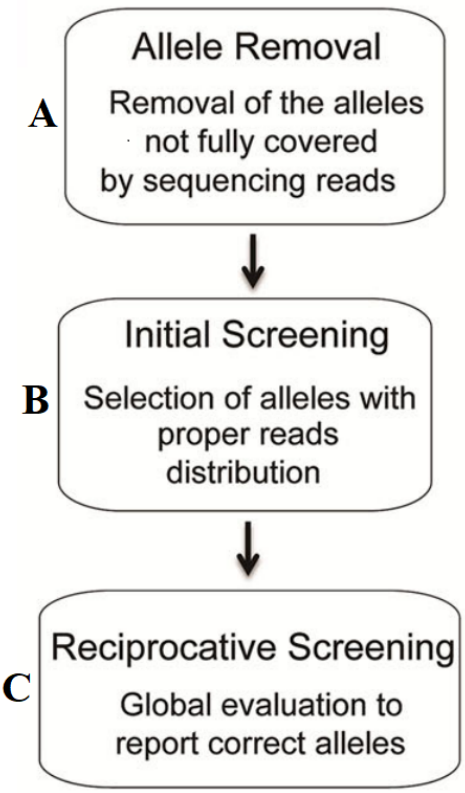

Figure 2.4.6: The three steps in STC-Seq's HLA allele-calling pipeline. **A:** The large exons (>70 bp) of alleles from the IPD-IMGT/HLA database are used as a reference. Alleles in this reference that have any exons which are not sufficiently covered by the input reads are removed. Here, "sufficiently" means at least a 70 bp continuously aligned region. **B:** The input reads are aligned to the full coding regions of the remaining candidate HLA alleles. Alleles are removed if they have uncovered regions and the corresponding region is covered for other alleles. **C:** After the two filtration steps, a scoring algorithm is used to find the allele combination that explains the maximum number of reads. Figure adapted from [84].

CHAPTER 3

Results

This chapter outlines the most important results from the benchmarking study of five NGS based HLA typing tools which were introduced in the previous chapter. The first section in this chapter investigates how well the tools perform using the full gold standard dataset, while the second section focuses on how the tools' performance depends on the amount of available data, specifically the coverage/sequencing depth of input files.

3.1 Benchmarking (full gold standard dataset)

This section focuses on HLA typing using the full gold standard dataset (all reads from all samples). The five NGS based HLA typing tools Kourami, HLA*LA, Optitype, HISAT-genotype and STC-Seq were evaluated based on their call rate and typing accuracy (section 2.3 and on their time and memory usage (section 2.3.3).

3.1.1 Typing accuracy

Figure 3.1.1 shows the five tools' call rate and typing accuracy of the three classical HLA class I genes. For HLA class I genes HLA*LA, Optitype and HISAT-genotype all have a call rate of 100%, meaning that they return allele predictions for all alleles in all samples. Kourami fails to return a prediction for almost 9% of alleles and STC-Seq for more than 30%. In Kourami's case, a failure to return a prediction often happens when there is not enough data to sufficiently cover important regions leading to Kourami's graph structure being disconnected

Optitype is the best performing of the tools with an almost perfect typing accuracy at 1-field resolution: 99.54%. The tool only mistypes 23 out of 4974 alleles. Optitype's typing accuracy expectedly drops when converting to higher resolutions, but is still above 98% at 2-field resolution. This confirms what previous studies have found regarding Optitype's performance[80, 97]. HLA*LA performs almost as well as Optitype at 1-field resolution but struggles to keep up Optitype at the higher resolutions. At 2-field resolution, HLA*LA on average only mistypes one allele out of four samples (6 alleles per sample) but still mistypes almost thrice as often as Optitype.

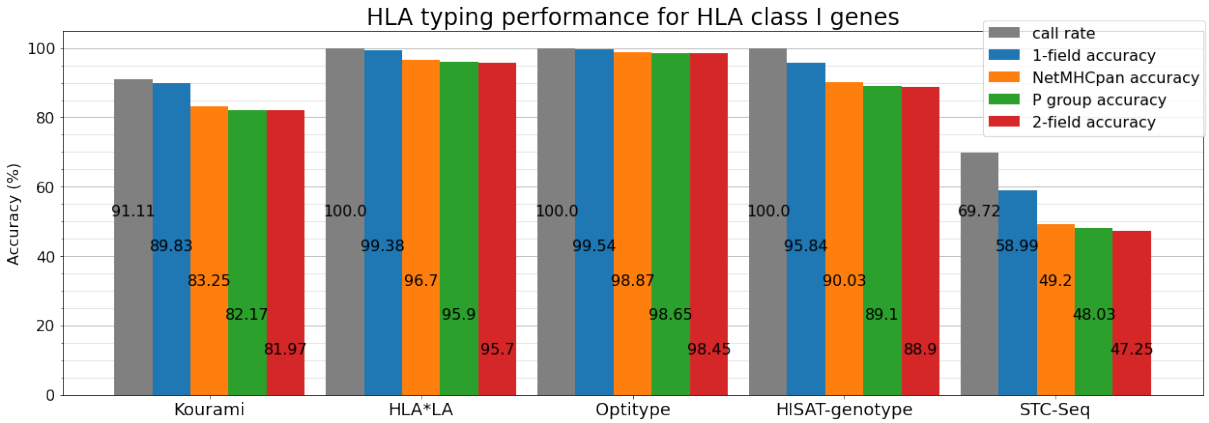

Figure 3.1.1: Typing accuracies for HLA-A, -B and -C (HLA class I genes). Optitype has the highest typing accuracy but HLA*LA performs almost as well. HLA*LA, Optitype and HISAT-genotype all have a call rate of 100 %.

HISAT-genotype has the third-best performing tool but like HLA*LA, HISAT-genotype's typing accuracy drops significantly when converting from 1-field resolution to a higher resolution.

Kourami does not return a prediction for all alleles, but the predictions that it *does* return are almost all (98.60%) correct at 1-field resolution. At 2-field resolution, 89.97 % of Kourami's predictions are correct resulting in a typing accuracy of 81.97%.

STC-Seq is designed to analyse high coverage data where reads are extracted and amplified, not WES data. Figure 3.1.1 shows that the tool performs significantly worse than the other four tools - as was expected with the WES dataset. The tool fails to return a prediction for more than 30% of the alleles and at 2-field resolution, roughly 1/3 of the predictions are wrong.

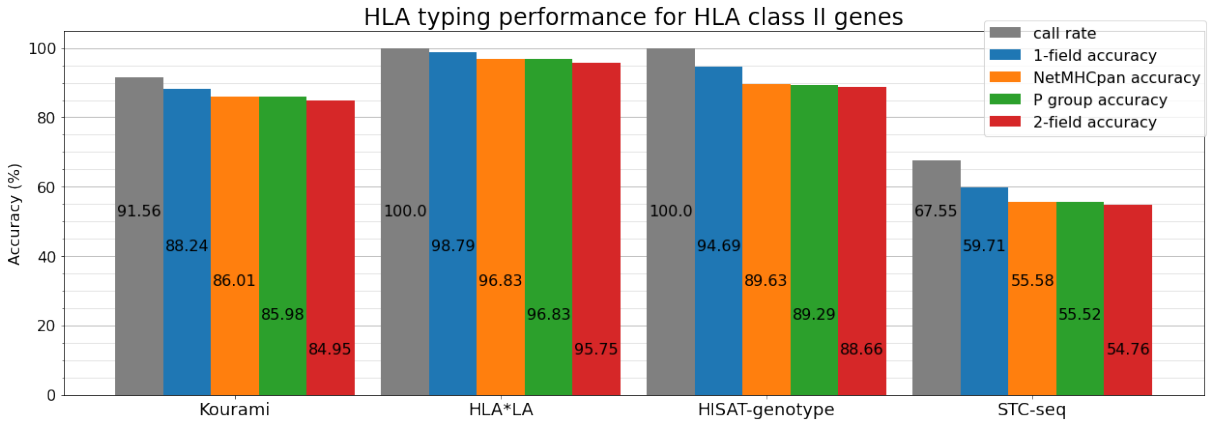

Figure 3.1.2: Typing accuracy for HLA-DRB1 and -DQB1 (HLA class II genes). Optitype is not included here, as it does not provide allele predictions for HLA class II genes. HLA*LA and HISAT-genotype have a call rate of 100% for both HLA class I and class II genes.

Figure 3.1.2 is parallel to figure 3.1.1 and shows the call rate and typing accuracy of the class II genes HLA-DRB1 and DQB1. Although there are more registered alleles for the three classical class I genes (HLA-A, -B and C) than the three classical class II genes (HLA-DR, -DQ and DP) (see figure A.0.2 in appendix A), class II genes are generally more difficult to type than class I genes (also mentioned in section 1.5.4). This trend is not reflected in this study as Kourami, HLA*LA, HISAT-Genotype and STC-Seq all seem to perform roughly as well for HLA class II genes as for HLA class I genes. Optitype - as mentioned previously - does not provide typing for HLA class II genes.

For both HLA class I and class II genes, a conversion between typing resolutions does not change the rank order of the tools' performance. In all typing resolution, Optitype has the highest typing accuracy for class I genes, HLA*LA has the second-highest and HISAT-genotype, Kourami and STC-Seq follows in order. The rank order is also the same for class II genes, save Optitype. The typing accuracy of a tool at a specific resolution can be relevant in some cases, but figures 3.1.1 and 3.1.2 indicate that the best performing tool at one resolution is also the best performing tool at other resolutions.

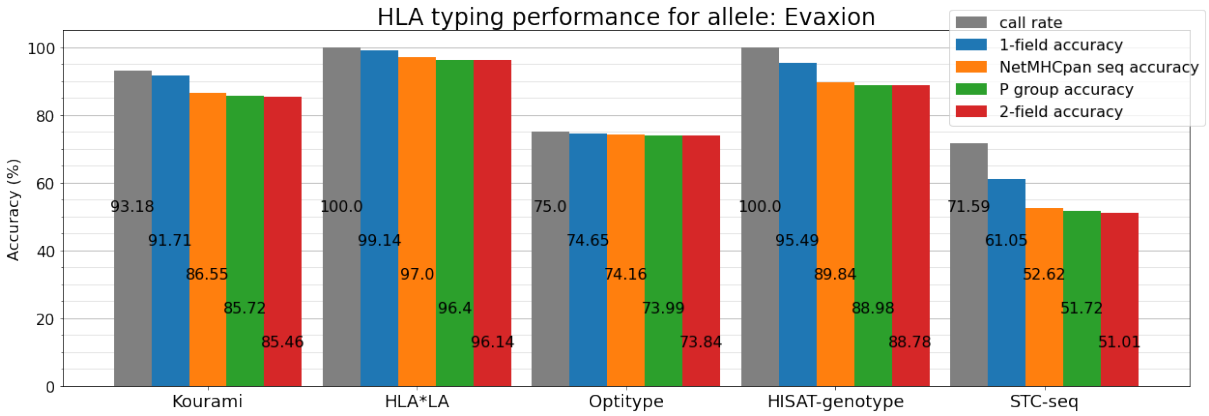

Figure 3.1.3: Typing accuracy for HLA-A, -B, -C and -DRB1 - the genes included in Evaxion's neoepitope prediction pipeline shown in figure 1.6.1.

Figure 3.1.3 shows the call rate and typing accuracy for the four HLA genes that are included in Evaxion's pipeline: - HLA-A, -B, -C and -DRB1. Optitype here has a call rate of 75% as it outputs predictions for the 3 represented class I genes. HLA*LA has the highest typing accuracy for this selection of HLA genes - 97% in the resolution relevant to Evaxion.

Figure 3.1.4 shows the overall typing accuracy of the five HLA typing algorithms for the five genes relevant to this study. Plots showing the performance of the HLA typing for individual genes in this study can be found in appendix B.

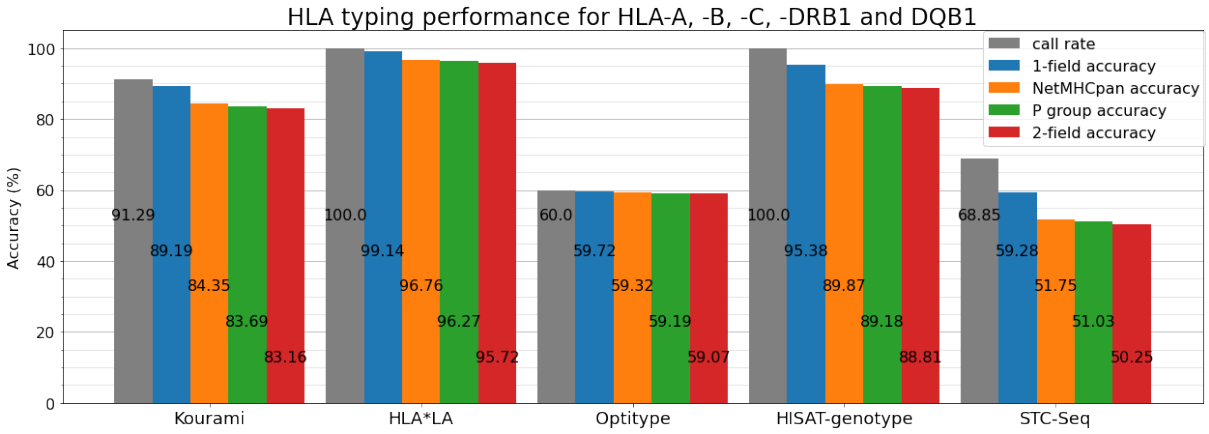

Figure 3.1.4: The typing accuracy of the five HLA typing tools for all samples and all genes included in the gold standard dataset. Optitype only outputs predictions for class I genes that make up three out of the five HLA genes which this plot is based on.

3.1.2 Time and memory usage

HLA*LA had the best typing accuracy for genes included in Evaxion's pipeline. The typing accuracy is the most important parameter in the comparison of the tools, but there are other features to take into account. It is also relevant to compare the computational resources that each tool needs to perform HLA typing. For each of the five tools, the peak memory usage and total CPU time and wall time usage was therefore registered for each of the 829 typed samples.

It is important to note that the registered computational resources to not comprise the total HLA typing process. Instead, this analysis only focuses on the step that is specific to each tool - the yellow boxes in figure 2.3.3, section 2.3.5. Steps not included in this analysis are: the initial preprocessing measures, the extraction of HLA reads and the conversion of CRAM files to BAM files and further to FASTQ files. The full process of HLA typing could therefore have a higher peak memory usage and would require more CPU time than denoted in this section.

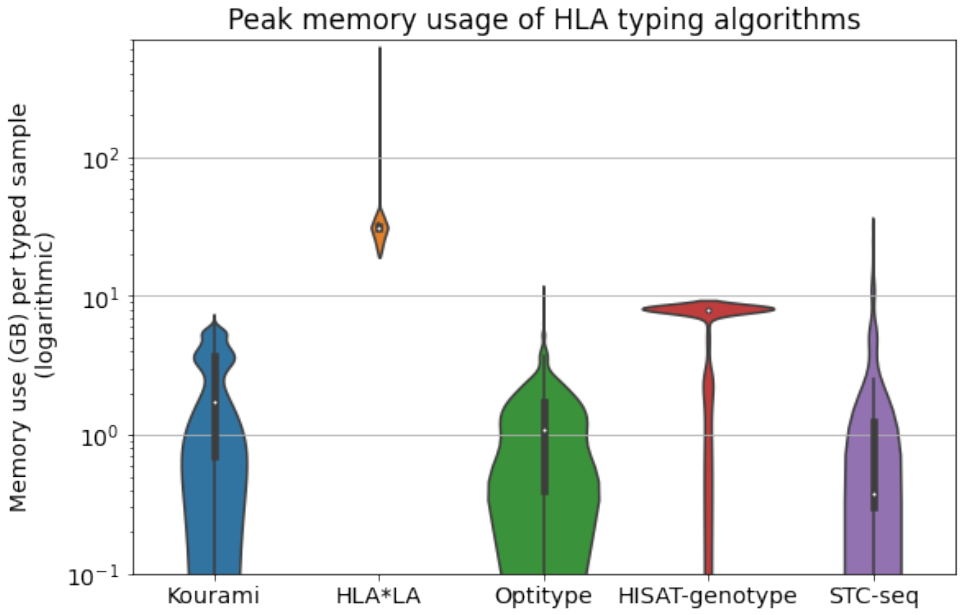

Figure 3.1.5: Peak memory usage distribution in the typing of the 829 samples in gold standard data set. Note that the scale is logarithmic. HLA*LA uses around 30 GB per sample while the other four tools almost consistently use less than 8 GB per sample. Of the four with a high typing accuracy, Optitype uses the least amount of memory - around 1 GB per typed sample.

Another important feature to note is that although the tools are compared directly in this

section, Optitype only provides typing for 3 out of the 5 HLA genes in the gold standard dataset.

Figure 3.1.5 shows how the peak memory usage was distributed over the 829 samples in the gold standard dataset for each of the five tools. HLA*LA uses a median of 31 GB memory per sample. However, some samples required significantly more - NA18504 required 610 GB of memory. This high memory usage is due to HLA*LA's expensive alignment step that uses dynamic-programming and is also noted in the HLA*LA blogpost[69, 125]. As a result, it is impossible to run HLA*LA on a normal laptop with 8 or 16 GB RAM thus the remaining four tools are preferable when a supercomputer is not available. Optitype has a median memory usage of 1.1GB and STC-Seq of 0.38 GB. However, these tools can also pose a problem for a standard laptop, as they for some samples (2 for Optitype and 82 for STC-Seq) use more than 8 GB of memory. Without considering STC-Seq due to its low typing accuracy, Optitype has the lowest memory usage. Kourami's median usage is 1.7 GB and the most demanding sample takes 6.4 GB. HISAT-genotype is limited to using a maximum of 8 GB memory (median 8.03 and maximum is 8.04). This results in a top-heavy, mushroom-looking violin plot. It does not seem like the limit to memory usage can be increased, but if it could (and if a system with more than 8 GB of memory is available) this could perhaps increase the speed of HLA typing for HISAT-genotype.

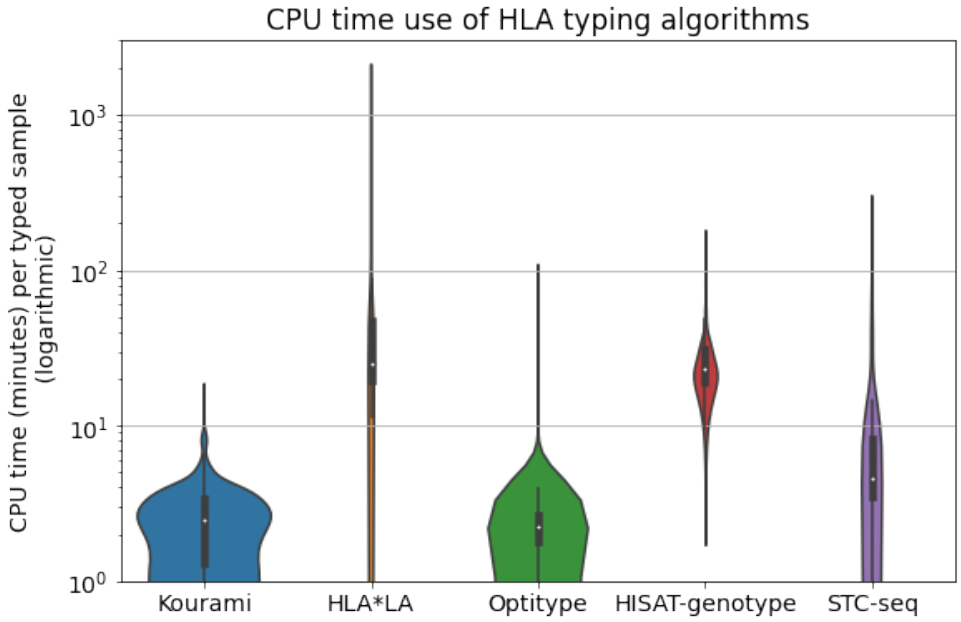

Figure 3.1.6: Distribution of CPU time usage. HLA*LA and HISAT-genotype have the highest median CPU time usage (between 20 and 30 CPU minutes), while the rest of the tools have a median less than 10 CPU minutes. For HLA*LA, the CPU time varies a lot between samples. Some samples are typed using a similar amount of resources as is used by Kourami and Optitype, while others require over 100 times more CPU time.

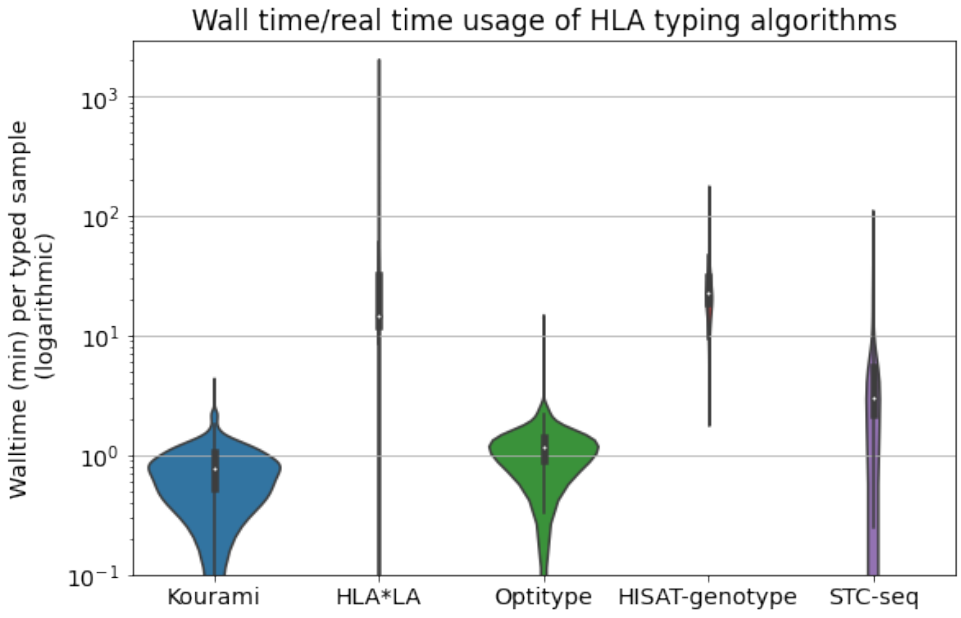

Figure 3.1.7: The real time usage of the five HLA typing tools. This figure tells a similar story to figure 3.1.6. HLA*LA and HISAT-genotype use the most time per typed sample, but some samples require several hours or more than a day for HLA*LA.

Figure 3.1.6 shows the CPU time usage and figure 3.1.7 shows the real time usage for the five tools. HLA*LA and HISAT-genotype require the most resources and type per sample (medians of 24 and 23 CPU minutes respectively), but for a few samples, HLA*LA requires an extraordinarily high amount of CPU time and real time. The sample requiring the longest real time to type for HLA*LA took more than 33 hours, while Kourami's maximum wall time usage was 4.2 minutes and Optitype's 14 minutes.

3.1.3 Ensemble method

Some HLA typing tools have typing problems for specific alleles. Therefore, it was investigated whether multiple tools using a majority vote (an ensemble method) could outperform any single tool (see section 2.3.2). This was not the case. For HLA class I genes, the ensemble was constructed by taking a majority vote from predictions by Kourami, HLA*LA, Optitype and HISAT-genotype. This approach was outperformed by Optitype in all resolutions and by HLA*LA in 1-field resolution (figure 3.1.8). For HLA class II genes, the ensemble was constructed using predictions by Kourami, HLA*LA and HISAT-genotype. This approach was outperformed by HLA*LA in all resolutions and HISAT-genotype in 1-field resolution (see figure 3.1.9).

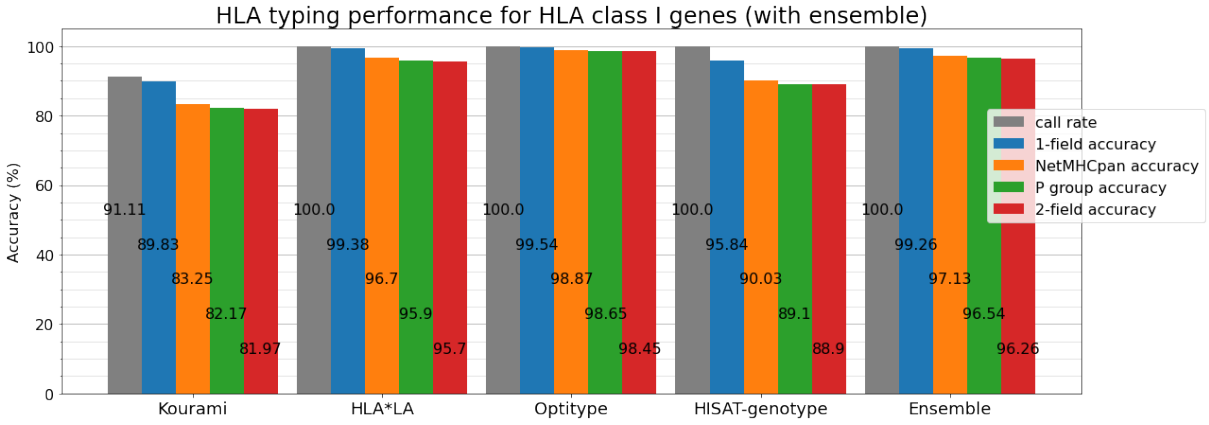

Figure 3.1.8: For HLA class I genes, an ensemble approach for HLA typing tools gives no advantage in comparison to using the best performing individual tool.

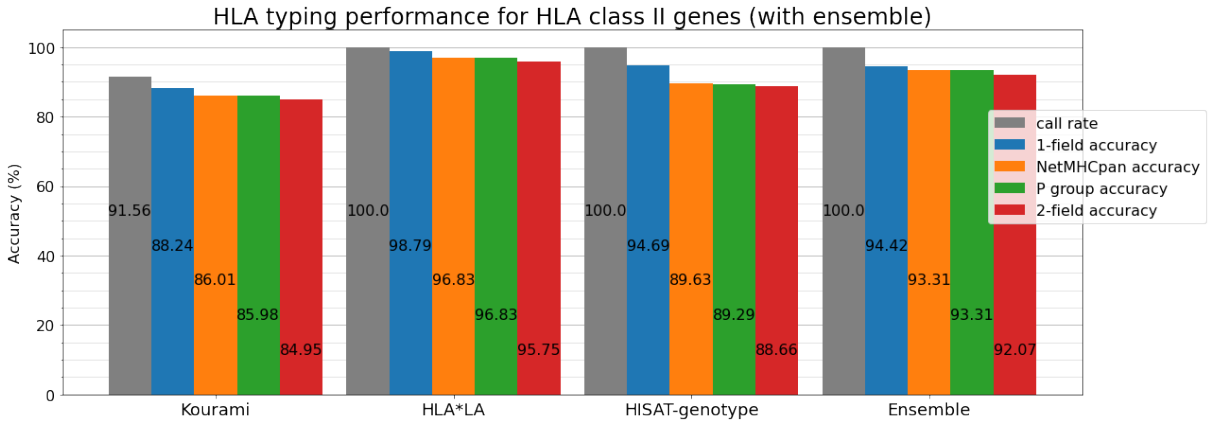

Figure 3.1.9: For HLA class II genes, an ensemble approach for HLA typing tools gives no advantage in comparison to using the best performing individual tool.

3.2 Downsampling analysis

NGS based HLA typing tools require sequencing data to perform HLA typing and more data is often an advantage. HLA typing tools will have a higher typing accuracy for a sample with coverage of 100X coverage than a sample with coverage of 1X. More data is, however, only an advantage up until a certain point - the typing accuracy might not increase as a consequence of increasing the coverage from 1000X to 2000X.
The amount of coverage that an HLA typing tool needs for optimal typing accuracy depends on the tool, e.g. STC-Seq is designed for HLA enriched (high coverage) data. This section aims to investigate how much coverage is required for optimal typing from WES data for the five HLA typing tools.

3.2.1 Typing accuracy

In some instances, samples with high coverage may be difficult to type, and samples with low coverage may be easily typed. This can be due to, e.g. the sample having rare or common HLA alleles. In the downsampling analysis, the same 230 samples were used and the typing accuracy of the five HLA typing tools was evaluated for these 230 samples at varying coverages. The samples were downsampled to 100X, 75X, 50X, 20X, 10X, 5X, 2X and 1X as shown in figures 3.2.1 (HLA class I genes) and 3.2.2 (HLA class II genes).
The figures show that for all tools and both HLA class I and II genes general trend is - as expected - that higher coverage correlates with higher typing accuracy. The correlation is however not linear and differs between the HLA typing tools.
The general trend is that the gain in typing accuracy is greatest at lower coverages and that there are diminishing returns in typing accuracy from increasing the coverage when the coverage is above a threshold.
At 1X, HLA*LA, Optitype and HISAT-genotype are still able to correctly type 15-25% of alleles whereas Kourami and STC-Seq are unable to type a single allele correctly (2300 alleles from 230 samples of 5 genes). For HLA class I genes, HISAT-genotype and Kourami have almost identical typing accuracy at 100X, but Kourami's typing accuracy diminishes much faster as coverage decreases - at 10X, Kourami's typing accuracy is 4.6% whereas HISAT-genotype's is 39.30%. Optitype is the most robust of the tools and maintains a

typing accuracy above 90% until 20X.

Perhaps, more interestingly, a significant increase in typing accuracy between 75X and 100X is observed for Kourami, HISAT-genotype and STC-Seq for HLA class I genes and for all tools in HLA class II genes. This indicates that increasing the coverage beyond 100X might result in an increased typing accuracy for these tools. Both the median coverage and the mean coverage of the gold standard dataset were lower than 100X (86X and 89X respectively). This indicates that the typing accuracies found in section 3.1 might not be truly reflective of how well the tools would perform if they had input data with sufficiently high coverage.

When looking at Optitype and HLA*LA (class I genes) it seems that these tools have hit a saturation point at 100X and might not benefit from increasing the coverage beyond 100X. This could well be the case, but Optitype's typing accuracy does increases from 97.68% at 75X to 98.12% at 100X and HLA*LA's (class I) increases from 96.09% to 96.16%.

In summary, there seems to be an advantage in increasing the coverage from 75X to 100X - especially for Kourami, HISAT-genotype and STC-Seq. HLA*LA (class I genes) and Optitype achieves higher typing accuracies at 100X than at 75X, but further studies would have to investigate whether increasing the coverage beyond 100X would be an advantage for these tools.

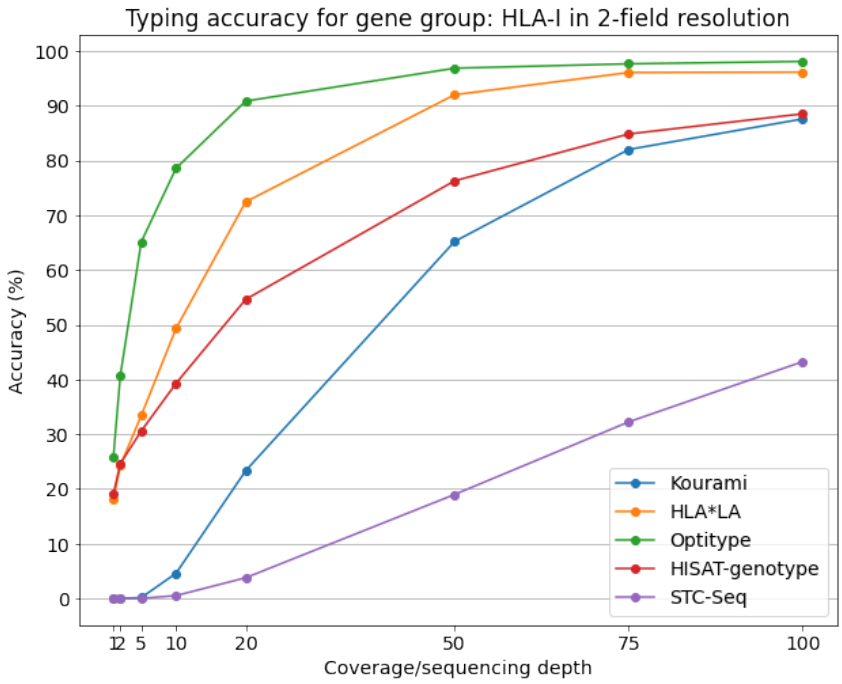

Figure 3.2.1: The typing accuracy of Kourami, HLA*LA, Optitype, HISAT-genotype and STC-Seq depends on the coverage of the sample. This figure shows the typing accuracy for the HLA class I genes HLA-A, HLA-B and HLA-C.

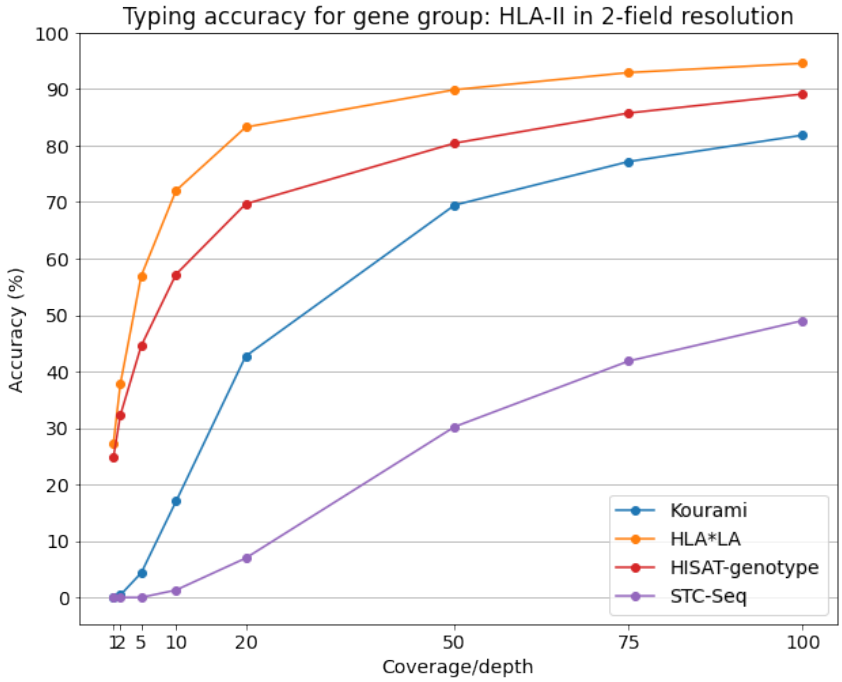

Figure 3.2.2: The typing accuracy of Kourami, HLA*LA, HISAT-genotype and STC-Seq depends on the coverage of the sample. This figure is parallel to figure 3.2.1, but shows the typing accuracy for the HLA class II genes, HLA-DRB1 and HLA-DQB1.

3.2.2 Time and memory usage

Increasing coverage means increasing the amount of data. Section 3.2.1 showed that this increased the typing accuracy, but it may also increase the computational resources required for HLA typing. This is investigated in the following.

Figure 3.2.3 shows how the median peak memory usage for the 230 samples in the downsampling analysis changes at increasing coverages. As also noted earlier, STC-Seq and Optitype use very little memory compared to the rest of the tools and HLA*LA uses significantly more. Neither Optitype's, STC-Seq's or HLA*LA's median peak memory usage differs much between low and high coverages. HISAT-genotype uses its peak 8GB of memory until between 50X and 20X when memory usage drops. Kourami strangely uses most memory at 20X and uses less at both 100X and 1X. One reason could be that there are more viable paths through the graph structure which Kourami keeps in memory at low coverages, but that it simply gives up predicting when the coverage becomes too low. Importantly, no tool seems to require significantly more memory if the sequencing coverage is increased beyond 100X.

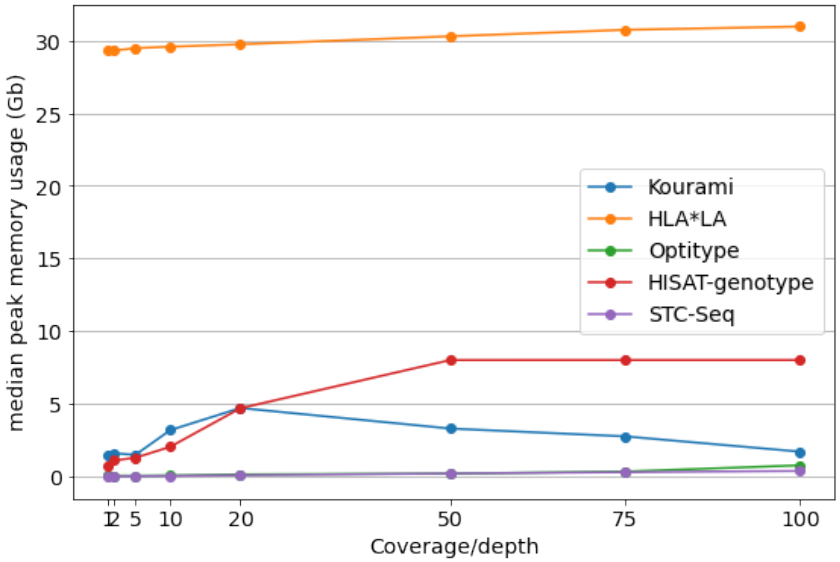

Figure 3.2.3: An increased coverage generally increases the peak memory usage of the HLA typing tools, but not greatly. For HLA*LA there is almost no difference in peak memory usage for 1X and 100X. HISAT-genotype uses more memory when the coverage is increased, but only up until 8 GB.

Figures 3.2.4 and 3.2.5 shows how the CPU time usage and real time usage of the five

HLA typing tools depend on the coverage of the input data. All tools use more resources at higher coverages. The increase in demand is, however, greater for HLA*LA and for HISAT-genotype, which are the tools using both the most CPU time and real time to perform the typing. Both CPU time usage and real time usage of HISAT-genotype seem to be linearly correlated with the coverage of the input data and use around 30 mins (both CPU and real time) for a sample with 100X.

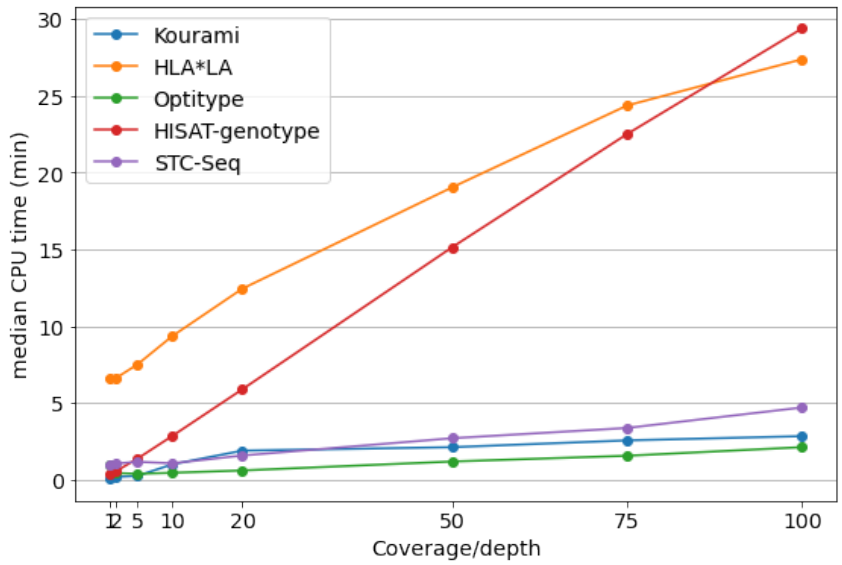

Figure 3.2.4: An increase in coverage also gives an increase in the CPU time needed to perform HLA typing. HLA*LA and HISAT-genotype need more resources than the rest of the tools. These two tools also need relatively more additional CPU time, when the coverage is increased, which could be a problem for high coverage samples.

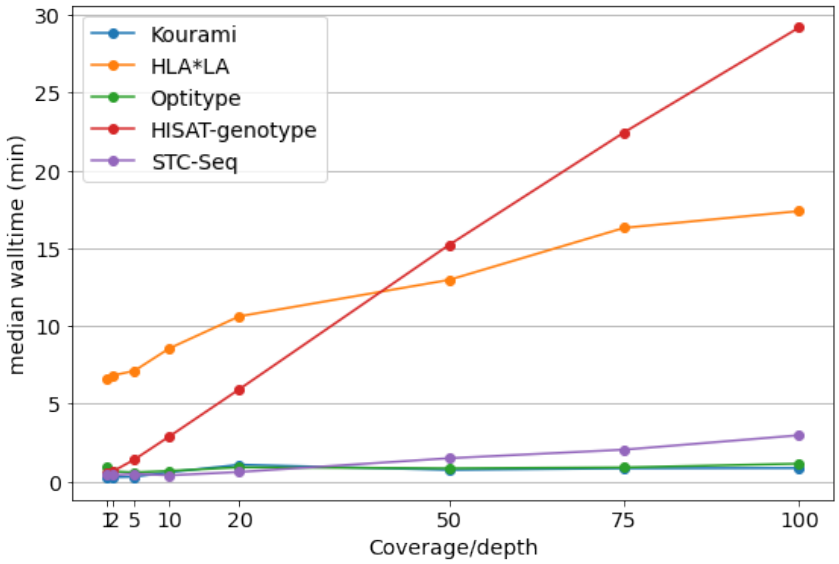

Figure 3.2.5: An increase in coverage also means that it takes more time for the tools to perform HLA typing. The results for walltime are parallel to the ones for CPU time in figure 3.2.4. By extrapolating the trend for HISAT-genotype, the tool would need more than an hour to type samples with a coverage of above 200X (using a setup identical to the one in this benchmarking study).

CHAPTER 4

Discussion

The last chapter discusses the overall findings of the benchmarking study. It also contains a discussion of the study's weaknesses in regards to the validity of the results and how studies could improve upon the findings in this project.

4.1 Inherent bias in the gold standard dataset

The WES samples from the IGSR database have been used both in previous benchmarking studies and in the development of HLA typing tools. This is important to be aware of when using the dataset to avoid drawing conclusions that the data does not support.
Figure 4.1.1 shows the number of individuals whose sequencing data are used both in the development of the HLA typing tools benchmarked in this project and in the gold standard dataset. Optitype was developed using data from 95 out of the 827 individuals that the gold standard data set is based on, and both Kourami and HLA*LA also have an overlap of samples. This means that performance measured using this dataset might not be truly reflective of how well these tools would perform with new data.
Another potential issue with using a dataset consisting only of registered, known HLA alleles is that the benefit of Kourami's and HISAT-genotype's ability to recognise novel alleles is not shown.

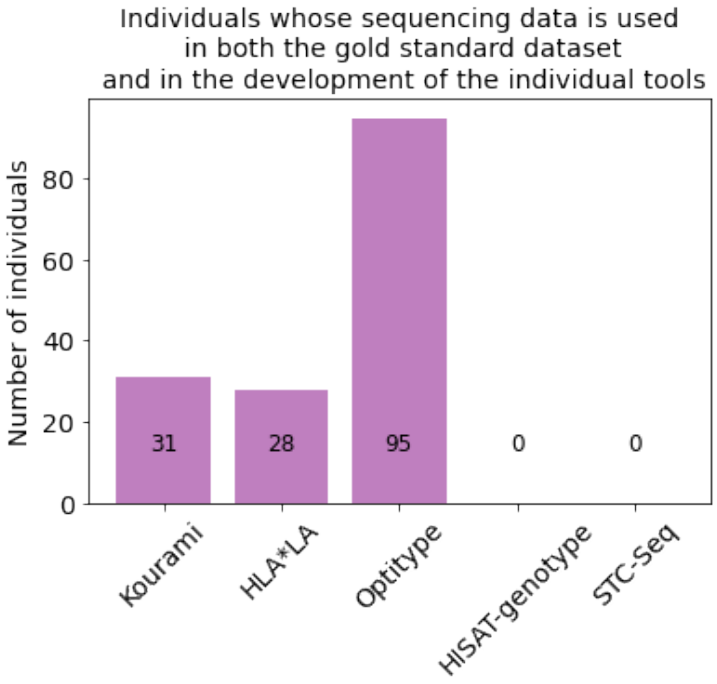

Figure 4.1.1: Kourami, HLA*LA and Optitype were developed using some of the samples included in the gold standard dataset used in this project and the accuracy of these tools might therefore be overestimated in this study.

In the downsampling analysis (section 3.2), it was found that all tools had a higher typing

accuracy at 100X than at 75X. This was especially the case for Kourami, HISAT-genotype and STC-Seq whereas the difference in typing accuracy was smaller for HLA*LA and Optitype. A majority of the samples in the full gold standard dataset (used in the main benchmarking analysis - section 3.1) had a coverage below 90X and the gold standard dataset, therefore, favours tools that perform well for samples with lower coverage and disfavours tools that only perform well for high coverage samples.
Sequencing at a higher coverage is more expensive and requires more storage; thus, it is a clear advantage if a tool can achieve high typing accuracy with low coverage samples. However, in Evaxion's case, all WES samples are high coverage (>250X), so Kourami, HISAT-genotype and STC-Seq may perform better using Evaxion's data rather than the gold standard dataset.

A potential pitfall, when trying to estimate the performance of an HLA typing tool, is to assume that two benchmarking studies using the same dataset are independent. WES samples from the IGSR database have been included in a number of the proof-of-concept studies in articles introducing NGS based HLA typing tools, but also in benchmarking studies from 2017[97] and 2018[80]. Optitype performs well in both studies, but as the studies use some of the same data and since some of that data was also used in the development of Optitype, these findings are not as conclusive as they would be if the dataset were mutually exclusive.

In summary, this gold standard dataset likely overestimates Optitype's performance compared to how well it would perform with new data and underestimates Kourami's, HISAT-genotype's and STC-Seq's performance compared to how well they would perform with sufficient coverage. Kourami and HISAT-genotype could perhaps be viable choices for high coverage WES data.

This project uses WES from the IGSR database and the typing data from the 2014 and 2018 study (see section 2.2) since this seems to be by far the largest and most complete dataset available. There are, however, some problems with this dataset and it could therefore be interesting in a future study to investigate the performance of the five HLA typing tools using new data. In Evaxion's perspective, it would especially be interesting,

if a new dataset contained samples with higher coverage comparable to the WES data used in their neoepitope prediction pipeline.

All tools in this study are designed to perform HLA typing based on DNA sequencing data. However, the benchmarking analysis shows that STC-Seq and Kourami are not well suited to perform HLA typing from lower coverage WES data. Both tools are reported to have a typing accuracy very close to 100% with their preferred input data (HLA enriched data for STC-Seq and high coverage WGS data for Kourami), but, in this benchmarking study, STC-Seq and Kourami only have call rates of 68.85% and 91.29% and typing accuracies of 50.25% and 83.16% (2-field resolution) respectively[69, 84]. The fact that the performance of these HLA typing tools is greatly affected when typing is performed from input data that does not exactly match what the tools are designed for, underlines the specificity of certain HLA typing tools and the difficulty of HLA typing compared to variant calling in general. When choosing between HLA typing algorithms and when evaluating their performance, it is important to not only note the type of input data (e.g. DNA sequencing), but also the specific technique (WGS/WES/HLA enriched data) and the specific coverage that the tool is designed for.

4.2 The performance of the tools

This section expands on section 3.1 and 3.2 and discusses the performance of Kourami, HISAT-genotype, HLA*LA and Optitype. STC-Seq is not elaborated on further due to its low typing accuracy for this gold standard dataset.

Kourami has a relatively low typing accuracy (83.16 % for all genes in 2-field resolution). It also only returned a prediction for 91% of the examined alleles, whereas HLA*LA, Optitype and HISAT-genotype all have a call rate of 100%. Some redeeming features of the tool are that it is fast, requires very little memory despite being a graph-based tool and it can discover novel alleles.
HISAT-genotype is also able to discover novel alleles and has a higher typing accuracy than Kourami. It, however, also uses more computational resources and might take more than an hour to type one of Evaxion's WES samples that have coverages of at least 250X. It is the only tool that provides 4-field resolution, but, in most practical cases, such high resolution is not necessary. Both HISAT2 and HISAT-genotype are continuously receiving new updates and their performance might improve in the future[127, 129].
The difference between the typing accuracy of Kourami and HISAT-genotype and the typing accuracy of the better-performing tools HLA*LA and Optitype would likely be smaller than shown in this study for Evaxion's high coverage WES samples. This is because Kourami and HISAT-genotype both seem to benefit from a coverage higher than 100X, while HLA*LA and Optitype are closer to a saturation point at 75X, where additional data does not benefit much. This is however only hypothesis and would have to be proven in a future study using high coverage WES data.
HLA*LA has the highest typing accuracy (95.72% at 2-field resolution) for the five HLA genes included in this study. However, this tool also requires the largest amount of memory (around 30 GB per sample), CPU time and real time to type a single sample. Using HLA*LA to type a few patients would generally not be an issue for Evaxion, but if HLA typing needed to be done quickly, many samples needed to be typed or a supercomputer is not available, HLA*LA is not a viable option.
Optitype has the highest typing accuracy for HLA class I genes in this study (98.45% at 2-field resolution). It further requires little memory, CPU time and real time. However, in

cases where a patient has rare HLA alleles not included in Optitype's reference database, this tool will return wrong predictions. This is a clear disadvantage of the tool that (as explained in section 4.1) is not necessarily reflected in the results of this benchmarking study. Based on the data from this benchmarking analysis, however, Optitype is the best tool for typing HLA class I genes.

None of the tools offers a fully satisfactory solution alone and one option for Evaxion is, therefore, to use Optitype to type HLA-A, -B and C and another to type HLA-DRB1. Figure 4.2.1 shows how well the other four tools typed HLA-DRB1 using the full gold standard dataset and figure 4.2.2 shows the results from the downsampling analysis specifically for HLA-DRB1.
Kourami and HLA*LA have the highest typing accuracy for both the full gold standard dataset and for the samples included in the downsampling analysis at 100X (the highest coverage). Kourami has a much higher call rate and typing accuracy for HLA-DRB1 than the other four HLA genes in the study. Since Evaxion has high coverage samples and Kourami performs worse with low samples, it is perhaps most relevant to compare the performances of the tools in the downsampling analysis at 100X. For the 230 samples in the downsampling analysis, both Kourami and HLA*LA have a call rate of 100%. HLA*LA has a slightly higher typing accuracy (97.83% at 2-field resolution and 98.04% at NetMHCpan resolution) than Kourami (97.61% at 2-field resolution and 97.83% at NetMHCpan resolution), but since HLA*LA requires significantly more computational resources, Kourami is likely the most fitting choice.

The recommendation for Evaxion based on the results of this project is therefore to perform HLA typing of class I genes with Optitype and of HLA-DRB1 with Kourami. If a combination of HLA*LA and Optitype is used instead, a coverage of above 250X is most likely not needed for accurate HLA typing, as both these tools already perform optimally at lower coverages.

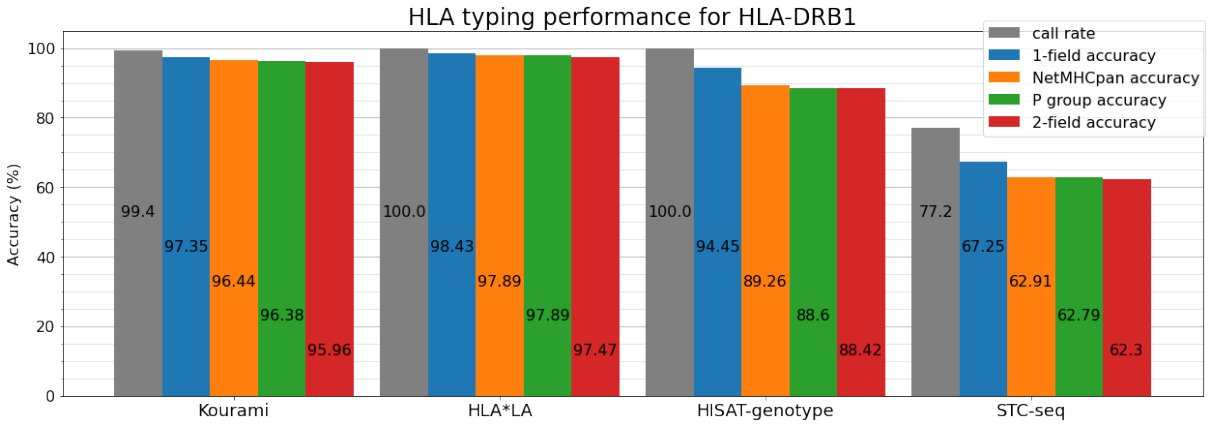

Figure 4.2.1: The typing accuracy and call rate of HLA-DRB1 alleles for the four NGS based HLA typing tools that predict HLA class II genes. The full dataset was used in producing these results. HLA*LA and Kourami are the best tools for typing HLA-DRB1 with a typing accuracy at 2-field resolution of 97.47% (HLA*LA) and 95.96% (Kourami). Kourami has a much higher call rate for HLA-DRB1 (99.4%) than the average call rate for all five genes in this study (91.29 %).

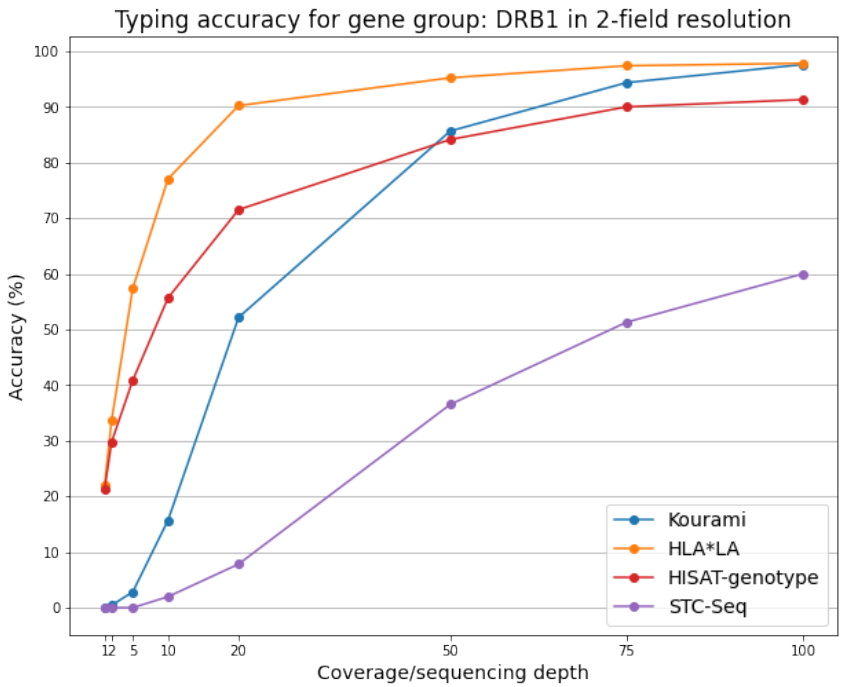

Figure 4.2.2: The typing accuracy of HLA-DRB1 alleles at differing coverages for the four NGS based HLA typing tools that predict HLA class II genes. Kourami and HLA*LA have the highest typing accuracy at 100X.

4.3 Future work and the future of HLA typing

New HLA typing tools are continuously being developed and some of the existing ones are continuously being improved. This means that the results in this thesis may be obsolete in a few years. This section discusses how the work in this project could be expanded upon and how NGS based HLA typing might develop in the future.

Section 4.1 discussed flaws of the gold standard dataset used both in this study and in previous benchmarking studies. This included an overlap between samples used in the development of HLA typing tools and samples in the gold standard dataset. Sections 3.2 and 4.2 further discussed how the coverage of many of the WES samples in the gold standard dataset is much lower than that of WES samples created for clinical purposes, and how that might lead to a benchmarking analysis underestimating how well some tools (e.g. Kourami and HISAT-genotype) would perform in industry with high coverage samples. A future benchmarking study could evaluate the performances of HLA typing tools by only using the subset of samples that are both high coverage and not included in the development of the tested tools. One drawback of this approach is that there only are a limited number of available samples that fulfil these criteria and a curated dataset constructed from in this way would be a small dataset. A better, but also more expensive, approach would be to create a new dataset of samples with NGS data and known HLA alleles. This could, for example, be done by coordinating studies that investigate HLA allele frequencies in populations (such as the ones listed on Allelefrequencies.net) and studies that seek to sequence a larger number of individuals.

Optitype was found to have the highest typing accuracy for the HLA class I genes included in the gold standard dataset, but the tool lacks HLA class II typing. The developers behind Optitype claim that it easily can be adapted to predict HLA class II genes and this could therefore be the objective of a future HLA typing project.
OncoHLA is a commercial HLA typing tool that provides typing for both class I and class II genes by using a similar approach to Optitype. It is reported to perform just as well as Optitype on HLA class I genes and to outperform Kourami and HLA:PRG (a precursor to HLA*LA) on HLA class II genes. This tool has a two-step HLA inference process

and HLA allele frequencies are considered in the last of these two steps. OncoHLA's code is similar to Optitype's publicly available code and OncoHLA uses public data from Allelefrequencies.net. The approach does not seem revolutionary and a public tool with a similar algorithm and performance will probably be developed in the future.

Another direction that HLA typing might take in the future is to expand and improve upon graph-based approaches. The graph-based extension of BWT was only developed in 2014 and the hierarchical graph FM index, that enables HLA typing using a fully graph-based approach, was partly developed with HISAT2 (first released in 2015). The developers behind HISAT2 and HISAT-genotype are still optimising the tools and future tools might also expand on these methods, further advancing and improving graph-based alignment. With an increasing number of registered alleles (also for other regions than the HLA region), the graph-based approach might overtake the linear based one in the future.

4.4 Conclusion

This study investigated the performance of five NGS based HLA typing tools: Kourami, HLA*LA, HISAT-genotype, Optitype and STC-Seq for the five HLA genes: HLA-A, -B, -C, -DRB1 and DQB1. The predictions were compared to 829 whole-exome sequencing samples with confirmed HLA alleles. HLA*LA was found to have the highest overall typing accuracy (95.72% at 2-field resolution) and the highest typing resolution for the two HLA class II genes (95.75% at 2-field resolution) while Optitype was found to have the highest typing accuracy for the three HLA class I genes (98.45% at 2-field resolution). The tools varied greatly in computational resource consumption with HLA*LA requiring 30 GB of memory per typed sample whereas Optitype only required 1 GB.

A downsampling analysis using 230 of the samples in the gold standard dataset showed that Kourami, HISAT-genotype and STC-Seq require coverage of above 100X for WES samples to type optimally. For HLA*LA and Optitype, increasing coverage beyond 75X only gives a limited increase in typing accuracy.

Considering both the typing accuracy and the required computational resources, it was found that the best method for typing the four HLA genes HLA-A, -B, -C and -DRB1 was to use Optitype for the HLA class I genes and Kourami for the class II gene HLA-DRB1.

References

[1] Jose L. Sanchez-Trincado, Marta Gomez-Perosanz, and Pedro A. Reche. Fundamentals and Methods for T- and B-Cell Epitope Prediction. *Journal of Immunology Research*, 2017, 2017. ISSN 23147156.

[2] Juan-Manuel Anaya, Yehuda Shoenfeld, Adriana Rojas-Villarraga, Roger A. Levy, and Ricard Cervera. *AUTOIMMUNITY From Bench to Bedside*. El Rosario University Press, 2013. ISBN 978-958-738-376-8.

[3] Jacques Neefjes, Marlieke L.M. Jongsma, Petra Paul, and Oddmund Bakke. Towards a systems understanding of MHC class i and MHC class II antigen presentation. *Nature Reviews Immunology*, 11(12):823–836, 2011. ISSN 14741733.

[4] Marine Leclerc, Laura Mezquita, Guillaume Guillebot De Nerville, Isabelle Tihy, Ines Malenica, Salem Chouaib, and Fathia Mami-Chouaib. Recent advances in lung cancer immunotherapy: Input of T-cell epitopes associated with impaired peptide processing. *Frontiers in Immunology*, 10(JUL):1–8, 2019. ISSN 16643224.

[5] James Robinson, Dominic J. Barker, Xenia Georgiou, Michael A. Cooper, Paul Flicek, and Steven G.E. Marsh. IPD-IMGT/HLA Database. *Nucleic Acids Research*, 48:D948–D955, 1 2020. ISSN 13624962.

[6] Sebastian Boegel. The Past, Present, and Future of HLA Typing in Transplantation. In Sebastian Boegel, editor, *HLA Typing (Methods in Molecular Biology)*, pages 1–10. Springer Science+Business Media, LLC, part of Springer Nature 2018, 2018. ISBN 978-1-4939-8546-3. URL `http://www.springer.com/series/7651`.

[7] Thomas Trolle, Curtis P McMurtrey, John Sidney, Wilfried Bardet, Sean C Osborn, Thomas Kaever, Alessandro Sette, William H. Hildebrand, Morten Nielsen, and

Bjoern Peters. The length distribution of class I restricted T cell epitopes is determined by both peptide supply and MHC allele specific binding preference. *Physiology & behavior*, 196(4):1480–1487, 2016.

[8] Kamilla Kjærgaard Jensen, Massimo Andreatta, Paolo Marcatili, Søren Buus, Jason A. Greenbaum, Zhen Yan, Alessandro Sette, Bjoern Peters, and Morten Nielsen. Improved methods for predicting peptide binding affinity to MHC class II molecules. *Immunology*, 154(3):394–406, 2018. ISSN 13652567.

[9] Sebastian Boegel, Thomas Bukur, John C. Castle, and Ugur Sahin. In Silico Typing of Classical and Non-classical HLA Alleles from Standard RNA-Seq Reads. In Sebastian Boegel, editor, *HLA Typing (Methods in Molecular Biology)*, pages 177–192. Springer Science+Business Media, LLC, part of Springer Nature 2018, 2018. ISBN 978-1-4939-8546-3. URL `http://www.springer.com/series/7651`.

[10] Calliope A. Dendrou, Jan Petersen, Jamie Rossjohn, and Lars Fugger. HLA variation and disease. *Nature Reviews Immunology*, 18(5):325–339, 5 2018. ISSN 14741741.

[11] Christian Kurts, Bruce W.S. Robinson, and Percy A. Knolle. Cross-priming in health and disease. *Nature Reviews Immunology*, 10(6):403–414, 2010. ISSN 14741733.

[12] Y. M. Mosaad. Clinical Role of Human Leukocyte Antigen in Health and Disease. *Scandinavian Journal of Immunology*, 82(4):283–306, 2015. ISSN 13653083.

[13] Massimo Andreatta and Morten Nielsen. Gapped sequence alignment using artificial neural networks: Application to the MHC class i system. *Bioinformatics*, 32(4): 511–517, 2016. ISSN 14602059.

[14] S. G.E. Marsh, E. D. Albert, W. F. Bodmer, and et al. An update to HLA Nomenclature, 2010. *Bone Marrow Transplantation*, 45(5):846–848, 2010. ISSN 02683369.

[15] Sabry Shoeib, Emad El-Shebiny, Alaa Efat, and KhairyA El-Hady. Human leukocyte antigen in medicine. *Menoufia Medical Journal*, 32(4):1197, 2019. ISSN 1110-2098.

[16] James Robinson, Jason A. Halliwell, James D. Hayhurst, Paul Flicek, Peter Parham, and Steven G.E. Marsh. IPD-IMGT/HLA webpage, 2015. URL `https://www.ebi.ac.uk/ipd/imgt/hla/`.

[17] James Robinson, Jason A. Halliwell, James D. Hayhurst, Paul Flicek, Peter Parham, and Steven G.E. Marsh. The IPD and IMGT/HLA database: Allele variant databases. *Nucleic Acids Research*, 43(D1):D423–D431, 2015. ISSN 13624962.

[18] JGraph. diagrams.net, v13.10.0, 2020. URL `https://www.diagrams.net/index.html`.

[19] Eduardo Nunes, Helen Heslop, Marcelo Fernandez-Vina, Cynthia Taves, Dawn R. Wagenknecht, A. Bradley Eisenbrey, Gottfried Fischer, Kay Poulton, Kara Wacker, Carolyn Katovich Hurley, Harriet Noreen, and Nicoletta Sacchi. Definitions of histocompatibility typing terms. *Blood*, 118(23):180–183, 2011. ISSN 00064971.

[20] Eduardo Nunes, Helen Heslop, Marcelo Fernandez-Vina, Cynthia Taves, Dawn R. Wagenknecht, A. Bradley Eisenbrey, Gottfried Fischer, Kay Poulton, Kara Wacker, Carolyn Katovich Hurley, Harriet Noreen, and Nicoletta Sacchi. Definitions of histocompatibility typing terms: Harmonization of Histocompatibility Typing Terms Working Group. *Human Immunology*, 72(12):1214–1216, 2011. ISSN 01988859. URL `http://dx.doi.org/10.1016/j.humimm.2011.06.002`.

[21] IGSR. IPD-IMGT/HLA: Nomenclature for Factors of the HLA System - G/P groups (Accessed: 01-12-2020). URL `http://hla.alleles.org/alleles/g_groups.html`.

[22] IGSR. IPD-IMGT/HLA: Suffixes used in the HLA Nomenclature (Accessed: 01-12-2020). URL `https://www.ebi.ac.uk/ipd/imgt/hla/nomenclature/suffixes.html`.

[23] Josué da Costa Lima-Junior and Lilian Rose Pratt-Riccio. Major histocompatibility complex and malaria: Focus on Plasmodium vivax Infection. *Frontiers in Immunology*, 7(JAN):1–13, 2016. ISSN 16643224.

[24] Alicia Sanchez-Mazas. A review of HLA allele and SNP associations with highly prevalent infectious diseases in human populations. *Swiss medical weekly*, 150(April): w20214, 2020. ISSN 14243997.

[25] Joseph Holoshitz. The quest for better understanding of HLA-disease association: Scenes from a road less travelled by. *Discovery Medicine*, 16(87):93–101, 2013. ISSN 19447930.

[26] Janelle A. Noble and Ana M. Valdes. Genetics of the HLA region in the prediction of type 1 diabetes. *Current Diabetes Reports*, 11(6):533–542, 2011. ISSN 15344827.

[27] Pierpaolo Correale, Luciano Mutti, Francesca Pentimalli, Giovanni Baglio, Rita Emilena Saladino, Pierpaolo Sileri, and Antonio Giordano. Hla-b*44 and c*01 prevalence correlates with covid19 spreading across italy. *International Journal of Molecular Sciences*, 21(15):1–12, 2020. ISSN 14220067.

[28] James T. Rosenbaum, Hedley Hamilton, Michael H. Weisman, John D. Reveille, Kevin L. Winthrop, and Dongseok Choi. The Effect of HLA-B27 on Susceptibility and Severity of COVID-19. *The Journal of Rheumatology*, page jrheum.200939, 2020. ISSN 0315-162X.

[29] Samuel E. Weinberg and Lawrence J. Jennings. HLA and autoimmune diseases. *ADVANCES IN MOLECULAR PATHOLOGY*, 3:207–219, 2020. ISSN 00926019. URL `https://doi.org/10.1016/j.yamp.2020.07.016`.

[30] Gizem Kumru Sahin, Christian Unterrainer, and Caner Süsal. Critical evaluation of a possible role of HLA epitope matching in kidney transplantation. *Transplantation Reviews*, 34(2), 2020. ISSN 15579816.

[31] Nadia Viborg. *Interrogating T cells specific to mutated and shared tumor epitopes in mouse and man PhD thesis*. PhD thesis, Technical University of Denmark, 2019.

[32] Kenneth Crump and Douglas H. Thamm. Cancer chemotherapy for the veterinary health team. In *John Wiley & Sons, Inc.*, pages 15–22. 2011. ISBN 9781118785621.

[33] Sjoerd H. Van Der Burg, Ramon Arens, Ferry Ossendorp, Thorbald Van Hall, and Cornelis J.M. Melief. Vaccines for established cancer: Overcoming the challenges posed by immune evasion. *Nature Reviews Cancer*, 16(4):219–233, 2016. ISSN 14741768. URL `http://dx.doi.org/10.1038/nrc.2016.16`.

[34] Sofia Farkona, Eleftherios P. Diamandis, and Ivan M. Blasutig. Cancer immunotherapy: The beginning of the end of cancer? *BMC Medicine*, 14(1):1–18, 2016. ISSN 17417015. URL `http://dx.doi.org/10.1186/s12916-016-0623-5`.

[35] Zhuting Hu, Patrick A. Ott, and Catherine J. Wu. Towards personalized, tumour-specific, therapeutic vaccines for cancer. *Nature reviews. Immunology*, 18(3):168–182, 2018. ISBN 2163684814.

[36] Xue Jiao Han, Xue Lei Ma, Li Yang, Yu Quan Wei, Yong Peng, and Xia Wei Wei. Progress in Neoantigen Targeted Cancer Immunotherapies. *Frontiers in Cell and Developmental Biology*, 8(July):1–19, 2020. ISSN 2296634X.

[37] F.S. Collins, E.S. Lander, J. Rogers, and Et Al. International Human Genome Sequencing Consortium,Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004. ISSN 0028-0836.

[38] National Human Genome Research Institute. Human Genome Project FAQ (Accessed: 04-12-2020). URL `https://www.genome.gov/human-genome-project/Completion-FAQ`.

[39] National Human Genome Research Institute. DNA Sequencing Costs: Data (Accessed: 04-12-2020), 2020. URL `https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data`.

[40] W. Richard McCombie, John D. McPherson, and Elaine R. Mardis. Next-generation sequencing technologies. *Cold Spring Harbor Perspectives in Medicine*, 9(11), 2019. ISSN 21571422.

[41] Christoph Bleidorn. Sequencing Techniques. In *Phylogenomics: An Introduction*, pages 43–60. Springer International Publishing, 2017. ISBN 9783319540641.

[42] James M. Heather and Benjamin Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1–8, 2016. ISSN 10898646. URL `http://dx.doi.org/10.1016/j.ygeno.2015.11.003`.

[43] Kevin Truong. Sequencing giant Illumina scraps $1.2 billion PacBio acquisition (Accessed: 21-01-2020), 2020. URL `https://www.bizjournals.com/sanfrancisco/news/2020/01/03/sequencing-giant-illumina-scraps-1-2-billion.html`.

[44] Illumina Inc. Illumina - sequencing platforms (Accessed: 04-12-2020). URL `https://www.illumina.com/systems/sequencing-platforms.html`.

[45] IGSR. Sequencing platform in the 1000 genomes project (Accessed: 10-01-2021). URL https://www.internationalgenome.org/faq/what-sequencing-platforms-were-used-1000-genomes-project/.

[46] Steffen Klasberg, Vineeth Surendranath, Vinzenz Lange, and Gerhard Schöfl. Bioinformatics Strategies, Challenges, and Opportunities for Next Generation Sequencing-Based HLA Genotyping, 10 2019. ISSN 16603818.

[47] Szilveszter Juhos, Krisztina Rigó, and György Horváth. On Genotyping Polymorphic HLA Genes — Ambiguities and Quality Measures Using NGS. In *Next Generation Sequencing - Advances, Applications and Challenges*, pages 369–386. 2015. URL https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics.

[48] Sarah B Ng, Emily H Turner, Peggy D Robertson, Steven D Flygare, W Abigail, Choli Lee, Tristan Shaffer, Michelle Wong, Arindam Bhattacharjee, E Evan, Michael Bamshad, Deborah a Nickerson, and Jay Shendure. Targeted Capture and Massicely Parallel Sequencing of twelve human exomes. *Nature*, 461(7261):272–276, 2010. ISBN 1476-4687 (Electronic)\r0028-0836 (Linking). ISSN 1476-4687.

[49] Lira Mamanova, Alison J. Coffey, Carol E. Scott, Iwanka Kozarewa, Emily H. Turner, Akash Kumar, Eleanor Howard, Jay Shendure, and Daniel J. Turner. Target-enrichment strategies for next-generation sequencing. *Nature Methods*, 7(2):111–118, 2010. ISSN 15487091.

[50] Jang Il Sohn and Jin Wu Nam. The present and future of de novo whole-genome assembly. *Briefings in Bioinformatics*, 19(1):23–40, 2018. ISSN 14774054.

[51] Adam Auton, Gonçalo R. Abecasis, David M. Altshuler, and Et Al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015. ISSN 14764687.

[52] Genome Reference Consortium. Genome Reference Consortium Human Build 38 patch release 13 (GRCh38.p13), 2019. URL https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39.

[53] Rockefeller University Bioinformatics Resource Centre. Genomic Data, Lecture Slides (accessed 07-12-2020). URL https://rockefelleruniversity.github.io/Genomic_Data/.

[54] NCBI. NCBI Genome Assembly Model (Accessed: 08-12-2020). URL https://www.ncbi.nlm.nih.gov/assembly/model/.

[55] M. Burrows and D.J. Wheeler. A Block-sorting Lossless Data Compression Algorithm. *Systems Research*, 1994.

[56] John C. Mu, Hui Jiang, Amirhossein Kiani, Marghoob Mohiyuddin, Narges Bani Asadi, and Wing H. Wong. Fast and accurate read alignment for resequencing. *Bioinformatics*, 28(18):2366–2373, 2012. ISSN 13674803.

[57] Nauman Ahmed, Koen Bertels, and Zaid Al-Ars. A comparison of seed-and-extend techniques in modern DNA read alignment algorithms. *Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016*, pages 1421–1428, 2016. ISBN 9781509016105.

[58] Subazini Thankaswamy-Kosalai, Partho Sen, and Intawat Nookaew. Evaluation and assessment of read-mapping by multiple next-generation sequencing aligners based on genome-wide characteristics. *Genomics*, 109(3-4):186–191, 2017. ISSN 10898646. URL http://dx.doi.org/10.1016/j.ygeno.2017.03.001.

[59] Sanger. SMALT (Accessed: 13-12-2020). URL https://www.sanger.ac.uk/tool/smalt-0/.

[60] Novocraft. Novoalign (Accessed: 12-12-2020). URL http://www.novocraft.com/products/novoalign/.

[61] Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 390–398. IEEE, 2000.

[62] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009. ISSN 13674803.

[63] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3), 2009. ISSN 14747596.

[64] Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, 2012. ISSN 15487091.

[65] Ruiqiang Li, Chang Yu, Yingrui Li, Tak Wah Lam, Siu Ming Yiu, Karsten Kristiansen, and Jun Wang. SOAP2: An improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966–1967, 2009. ISSN 13674803.

[66] Adam Ameur. Goodbye reference, hello genome graphs. *Nature Biotechnology*, 37 (8):866–868, 2019. ISBN 4158701902041. ISSN 15461696.

[67] Birte Kehr, Kathrin Trappe, Manuel Holtgrewe, and Knut Reinert. Genome alignment with graph data structures: A comparison. *BMC Bioinformatics*, 15(1), 2014. ISSN 14712105.

[68] Jouni Sirén, Niko Välimäki, and Veli Mäkinen. Indexing graphs for path queries with applications in genome research. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(2):375–388, 2014. ISSN 15455963.

[69] Heewook Lee and Carl Kingsford. Kourami: Graph-guided assembly for novel human leukocyte antigen allele discovery. *Genome Biology*, 19(1):1–16, 2018. ISSN 1474760X.

[70] Hongen Zhang. *Statistical Genomics - Chapter 1: Overview of Sequence Data Formats*, volume 1418. Springer Science+Business Media, LLC, 2016. ISBN 9781493935765. ISSN 10643745. 3–18 pp.

[71] Robert Edgar. Quality (Phred) scores (Accessed: 08-12-2020). URL `https://www.drive5.com/usearch/manual/quality_score.html`.

[72] Shifu Chen, Yanqing Zhou, Yaru Chen, and Jia Gu. Fastp: An ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics*, 34(17):i884–i890, 2018. ISSN 14602059.

[73] Samtools view manual (Accessed: 08-12-2020). URL `http://www.htslib.org/doc/samtools-view.html`.

[74] Markus Hsi Yang Fritz, Rasko Leinonen, Guy Cochrane, and Ewan Birney. Efficient storage of high throughput DNA sequencing data using reference-based compression. *Genome Research*, 21(5):734–740, 2011. ISSN 10889051.

[75] Aaron R. Quinlan and Ira M. Hall. BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010. ISSN 13674803.

[76] Sohyun Hwang, Eiru Kim, Insuk Lee, and Edward M. Marcotte. Systematic comparison of variant calling pipelines using gold standard personal exome variants. *Scientific Reports*, 5(December):1–8, 2015. ISSN 20452322.

[77] Sebastian Boegel. HLA Typing from Short-Read Sequencing Data with OptiType. In Sebastian Boegel, editor, *HLA Typing (Methods in Molecular Biology)*, pages 215–224. Springer Science+Business Media, LLC, part of Springer Nature 2018, 2018. ISBN 978-1-4939-8546-3. URL `http://www.springer.com/series/7651`.

[78] Alexander T. Dilthey, Alexander J. Mentzer, Raphael Carapito, Clare Cutland, Nezih Cereb, Shabir A. Madhi, Arang Rhie, Sergey Koren, Seiamak Bahram, Gil McVean, and Adam M. Phillippy. HLA-LA - HLA typing from linearly projected graph alignments. *Bioinformatics*, 35(21):4394–4396, 2019. ISSN 14602059.

[79] BaseClear. Baseclear, Variant detection pipeline (accessed 10-12-2020). URL `https://www.baseclear.com/blog/bioinformatics/baseclears-variant-detection-pipeline-renewed/`.

[80] Denis C. Bauer, Armella Zadoorian, Laurence O.W. Wilson, and Natalie P. Thorne. Evaluation of computational programs to predict HLA genotypes from genomic sequencing data. *Briefings in Bioinformatics*, 19(2):179–187, 3 2018. ISSN 14774054.

[81] J.L. Schmitz. HLA typing using molecular methods. *Molecular Diagnostics*, pages 485–493, 2006.

[82] Emma C. Morris and Kirsty J. Thomson. An Overview of HLA Typing for Hematopoietic Stem Cell Transplantation. In John M. Walker, editor, *Bone Marrow and Stem Cell Transplantation (Methods in Molecular Biology)*, volume 37, pages 73–86. Springer Protocols, 2nd edition, 2014. ISBN 9781461494362. ISSN 13573039.

[83] Minh Duc Do, Linh Gia Hoang Le, Vinh The Nguyen, Tran Ngoc Dang, Nghia Hoai Nguyen, Hoang Anh Vu, and Thao Phuong Mai. High-Resolution HLA Typing of HLA-A, -B, -C, -DRB1, and -DQB1 in Kinh Vietnamese by Using Next-Generation Sequencing. *Frontiers in Genetics*, 11(April):1–10, 2020. ISSN 16648021.

[84] Yang Jiao, Ran Li, Chao Wu, Yibin Ding, Yanning Liu, Danmei Jia, Lifeng Wang, Xiang Xu, Jing Zhu, Min Zheng, and Junling Jia. High-sensitivity HLA typing by Saturated Tiling Capture Sequencing (STC-Seq). *BMC Genomics*, 19(1):1–10, 2018. ISBN 1286401844. ISSN 14712164.

[85] René L. Warren, Gina Choe, Douglas J. Freeman, Mauro Castellarin, Sarah Munro, Richard Moore, and Robert A. Holt. Derivation of HLA types from shotgun sequence datasets. *Genome Medicine*, 4(12), 2012. ISSN 1756994X.

[86] Angelina Sverchkova, Irantzu Anzar, Richard Stratford, and Trevor Clancy. Improved HLA typing of Class I and Class II alleles from next-generation sequencing data. *HLA*, 94(6):504–513, 12 2019. ISSN 20592310.

[87] Chang Liu, Xiao Yang, Brian Duffy, Thalachallour Mohanakumar, Robi D. Mitra, Michael C. Zody, and John D. Pfeifer. ATHLATES: Accurate typing of human leukocyte antigen through exome sequencing. *Nucleic Acids Research*, 41(14), 8 2013. ISSN 03051048.

[88] Yazhi Huang, Jing Yang, Dingge Ying, Yan Zhang, Vorasuk Shotelersuk, Nattiya Hirankarn, Pak Chung Sham, Yu Lung Lau, and Wanling Yang. HLAreporter: A tool for HLA typing from next generation sequencing data. *Genome Medicine*, 7(1), 3 2015. ISSN 1756994X.

[89] Shuto Hayashi, Takuya Moriyama, Rui Yamaguchi, Shinichi Mizuno, Mitsuhiro Komura, Satoru Miyano, Hidewaki Nakagawa, and Seiya Imoto. Alphlard-nt: Bayesian method for human leukocyte antigen genotyping and mutation calling through simultaneous analysis of normal and tumor whole-genome sequence data. *Journal of Computational Biology*, 26(9):923–937, 2019. ISSN 10665277.

[90] Sachet A. Shukla, Michael S. Rooney, Mohini Rajasagi, Grace Tiao, Philip M. Dixon, Michael S. Lawrence, Jonathan Stevens, William J. Lane, Jamie L. Dellagatta, Scott

Steelman, Carrie Sougnez, Kristian Cibulskis, Adam Kiezun, Nir Hacohen, Vladimir Brusic, Catherine J. Wu, and Gad Getz. Comprehensive analysis of cancer-associated somatic mutations in class i HLA genes. *Nature Biotechnology*, 33(11):1152–1158, 10 2015. ISSN 15461696.

[91] Yu Bai, Min Ni, Blerta Cooper, Yi Wei, and Wen Fury. Inference of high resolution HLA types using genome-wide RNA or DNA sequencing reads. *BMC Genomics*, 15 (1), 5 2014. ISSN 14712164.

[92] András Szolek, Benjamin Schubert, Christopher Mohr, Marc Sturm, Magdalena Feldhahn, and Oliver Kohlbacher. OptiType: Precision HLA typing from next-generation sequencing data. *Bioinformatics*, 30(23):3310–3316, 2014. ISSN 14602059.

[93] Daehwan Kim, Joseph M. Paggi, Chanhee Park, Christopher Bennett, and Steven L. Salzberg. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature Biotechnology*, 37(8):907–915, 8 2019. ISSN 15461696.

[94] Faviel F. Gonzalez-Galarza, Antony McCabe, Eduardo J.Melo Dos Santos, James Jones, Louise Takeshita, Nestor D. Ortega-Rivera, Glenda M.Del Cid-Pavon, Kerry Ramsbottom, Gurpreet Ghattaoraya, Ana Alfirevic, Derek Middleton, and Andrew R. Jones. Allele frequency net database (AFND) 2020 update: Gold-standard data classification, open access genotype data and new query tools. *Nucleic Acids Research*, 48(D1):D783–D788, 2020. ISSN 13624962.

[95] Martin L. Buchkovich, Chad C. Brown, Kimberly Robasky, Shengjie Chai, Sharon Westfall, Benjamin G. Vincent, Eric T. Weimer, and Jason G. Powers. HLAProfiler utilizes k-mer profiles to improve HLA calling accuracy for rare and common alleles in RNA-seq data. *Genome Medicine*, 9(1):1–15, 2017. ISBN 1307301704736. ISSN 1756994X.

[96] Rose Orenbuch, Ioan Filip, Devon Comito, Jeffrey Shaman, Itsik Pe'Er, Raul Rabadan, and Yann Ponty. ArcasHLA: High-resolution HLA typing from RNAseq. *Bioinformatics*, 36(1):33–40, 2020. ISSN 14602059.

[97] Antti Larjo, Robert Eveleigh, Elina Kilpeläinen, and et al. Accuracy of programs for the determination of human leukocyte antigen alleles from next-generation sequencing data. *Frontiers in Immunology*, 8(DEC), 12 2017. ISSN 16643224.

[98] Chao Xie, Zhen Xuan Yeo, Marie Wong, Jason Piper, Tao Long, Ewen F. Kirkness, William H. Biggs, Ken Bloom, Stephen Spellman, Cynthia Vierra-Green, Colleen Brady, Richard H. Scheuermann, Amalio Telenti, Sally Howard, Suzanne Brewerton, Yaron Turpaz, and J. Craig Venter. Fast and accurate HLA typing from short-read next-generation sequence data with xHLA. *Proceedings of the National Academy of Sciences of the United States of America*, 114(30):8059–8064, 7 2017. ISSN 10916490.

[99] Haibao Tang, Andrew J. Rech, and Theodore Wong. xHLA GitHub (Accessed: 13-01-2021). URL `https://github.com/humanlongevity/HLA`.

[100] Yen-Yen Wang, Takahiro Mimori, Seik-Soon Khor, Olivier Gervais, Yosuke Kawai, Yuki Hitomi, Katsushi Tokunaga, and Masao Nagasaki. HLA-VBSeq v2: improved HLA calling accuracy with full-length Japanese class-I panel. *Human Genome Variation*, 6(1), 12 2019. ISSN 2054-345X.

[101] David L. Altshuler, Richard M. Durbin, Gonçalo R. Abecasis, and Et Al. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319): 1061–1073, 2010. ISSN 14764687.

[102] Laura Clarke, Susan Fairley, Xiangqun Zheng-Bradley, Ian Streeter, Emily Perry, Ernesto Lowy, Anne Marie Tassé, and Paul Flicek. The international Genome sample resource (IGSR): A worldwide collection of genome variation incorporating the 1000 Genomes Project data. *Nucleic Acids Research*, 45(D1):D854–D859, 2017. ISSN 13624962.

[103] Tianming Lan, Haoxiang Lin, Wenjuan Zhu, Tellier Christian Asker Melchior Laurent, Mengcheng Yang, Xin Liu, Jun Wang, Jian Wang, Huanming Yang, Xun Xu, and Xiaosen Guo. Deep whole-genome sequencing of 90 Han Chinese genomes. *GigaScience*, 6(9):1–7, 2017. ISSN 2047217X.

[104] IGSR. The International Genome Sample Resource (ABOUT) (Accessed: 27-11-2020). URL `https://www.internationalgenome.org/about`.

[105] IGSR. The International Genome Sample Resource (FAQ) (Accessed: 27-11-2020). URL `https://www.internationalgenome.org/faq/`.

[106] IGSR. The International Genome Sample Resource (bam) (Accessed: 27-11-2020). URL https://www.internationalgenome.org/category/bam/.

[107] Pierre Antoine Gourraud, Pouya Khankhanian, Nezih Cereb, Soo Young Yang, Michael Feolo, Martin Maiers, John D. Rioux, Stephen Hauser, and Jorge Oksenberg. HLA diversity in the 1000 genomes dataset. *PLoS ONE*, 9(7), 2014. ISSN 19326203.

[108] Laurent Abi-Rached, Philippe Gouret, Jung Hua Yeh, Julie Di Cristofaro, Pierre Pontarotti, Christophe Picard, and Julien Paganini. Immune diversity sheds light on missing variation in worldwide genetic diversity panels. *PLoS ONE*, 13(10):1–11, 2018. ISBN 1111111111. ISSN 19326203.

[109] Deleted HLA alleles (Accessed: 15-01-2021). URL http://hla.alleles.org/alleles/deleted.html.

[110] Helen M. Berman, Tammy Battistuz, T. N. Bhat, Wolfgang F. Bluhm, Philip E. Bourne, Kyle Burkhardt, Zukang Feng, Gary L. Gilliland, Lisa Iype, Shri Jain, Phoebe Fagan, Jessica Marvin, David Padilla, Veerasamy Ravichandran, Bohdan Schneider, Narmada Thanki, Helge Weissig, John D. Westbrook, and Christine Zardecki. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6 I):899–907, 2000. ISSN 09074449.

[111] Y. Li and R.A. Mariuzza. Structure of MHC class II molecule HLA-DR1 complexed with phosphopeptide MART-1 (Accessed: 21-10-2021), 2010. URL https://www.rcsb.org/structure/3l6f.

[112] LLC Schröodinger. PyMOL, The PyMOL Molecular Graphics System, Version 2.4, 2020. URL https://pymol.org/2/.

[113] Edita Karosiene, Michael Rasmussen, Thomas Blicher, Ole Lund, Søren Buus, and Morten Nielsen. NetMHCIIpan-3.0, a common pan-specific MHC class IIprediction method including all three human MHC class IIisotypes, HLA-DR, HLA-DP and HLA-DQ. *Immunogenetics*, 65(10), 2013. ISBN 6176321972. ISSN 15378276. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3624763/pdf/nihms412728.pdf.

[114] Kazuma Kiyotani, Tu H. Mai, and Yusuke Nakamura. Comparison of exome-based HLA class i genotyping tools: Identification of platform-specific genotyping errors. *Journal of Human Genetics*, 62(3):397–405, 3 2017. ISSN 1435232X.

[115] Brent S. Pedersen and Aaron R. Quinlan. Mosdepth: Quick coverage calculation for genomes and exomes. *Bioinformatics*, 34(5):867–868, 3 2018. ISSN 14602059.

[116] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009. ISSN 13674803.

[117] Sven Rahmann and Johannes Ko. Snakemake — a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2522, 2012.

[118] Maria Luisa Matey-Hernandez, Søren Brunak, and Jose M.G. Izarzugaza. Benchmarking the HLA typing performance of Polysolver and Optitype in 50 Danish parental trios. *BMC Bioinformatics*, 19(1), 6 2018. ISSN 14712105.

[119] David Weese, Manuel Holtgrewe, and Knut Reinert. RazerS 3: Faster, fully sensitive read mapping. *Bioinformatics*, 28(20):2592–2599, 2012. ISSN 13674803.

[120] András Szolek, Benjamin Schubert, and Christopher Mohr. Optitype Github (Accessed: 06-01-2020), 2014. URL `https://github.com/FRED-2/OptiType`.

[121] CBC 2.9.5. URL `https://github.com/coin-or/Cbc/releases/tag/releases%2F2.9.5`.

[122] Carl Kingsford and Heewook Lee. Kourami GitHub (Accessed: 06-01-2020), 2018. URL `https://github.com/Kingsford-Group/kourami`.

[123] Heewook Lee and Carl Kingsford. Accurate Assembly and Typing of HLA using a Graph-Guided Assembler Kourami. *Methods in Molecular Biology*, 1802, 2018.

[124] Alexander T. Dilthey, Jan Forster, and Justin Graham. HLA*LA GitHub (Accessed: 11-01-2021). URL `https://github.com/DiltheyLab/HLA-LA`.

[125] Alexander T. Dilthey. Fast and highly accurate HLA typing by linearly-seeded graph alignment - The HLA*LA blogpost (Accessed: 11-01-2021), 2017. URL `https://genomeinformatics.github.io/HLA-PRG-LA/`.

[126] Daehwan Kim and Chris Bennett. HISAT-genotype GitHub (Accessed: 11-01-2021). URL https://github.com/DaehwanKimLab/hisat-genotype.

[127] Daehwan Kim, Chris Bennett, and Chanhee Park. HISAT-genotype official website (Accessed: 11-01-2021). URL https://daehwankimlab.github.io/hisat-genotype/main/.

[128] Yang Jiao, Ran Li, Chao Wu, Yibin Ding, Yanning Liu, Danmei Jia, Lifeng Wang, Xiang Xu, Jing Zhu, Min Zheng, and Junling Jia. STC-Seq manual (accessed 11-01-2021). URL https://bigd.big.ac.cn/biocode/tools/BT007068/manual.

[129] Chanhee Park, Ben Langmead, Yun (Leo) Zhang, Steven Salzberg, and Daehwan Kim. HISAT2 official website (Accessed: 18-01-2021). URL https://daehwankimlab.github.io/hisat2/main/.

Appendices

APPENDIX A

Statistics from the IPD-IMGT/HLA database

The number of registered HLA alleles in the IMGT has been rising rapidly in the last years as seen in figure A.0.1. A current overview of the number of registered alleles for each gene is shown in figure A.0.2[5].

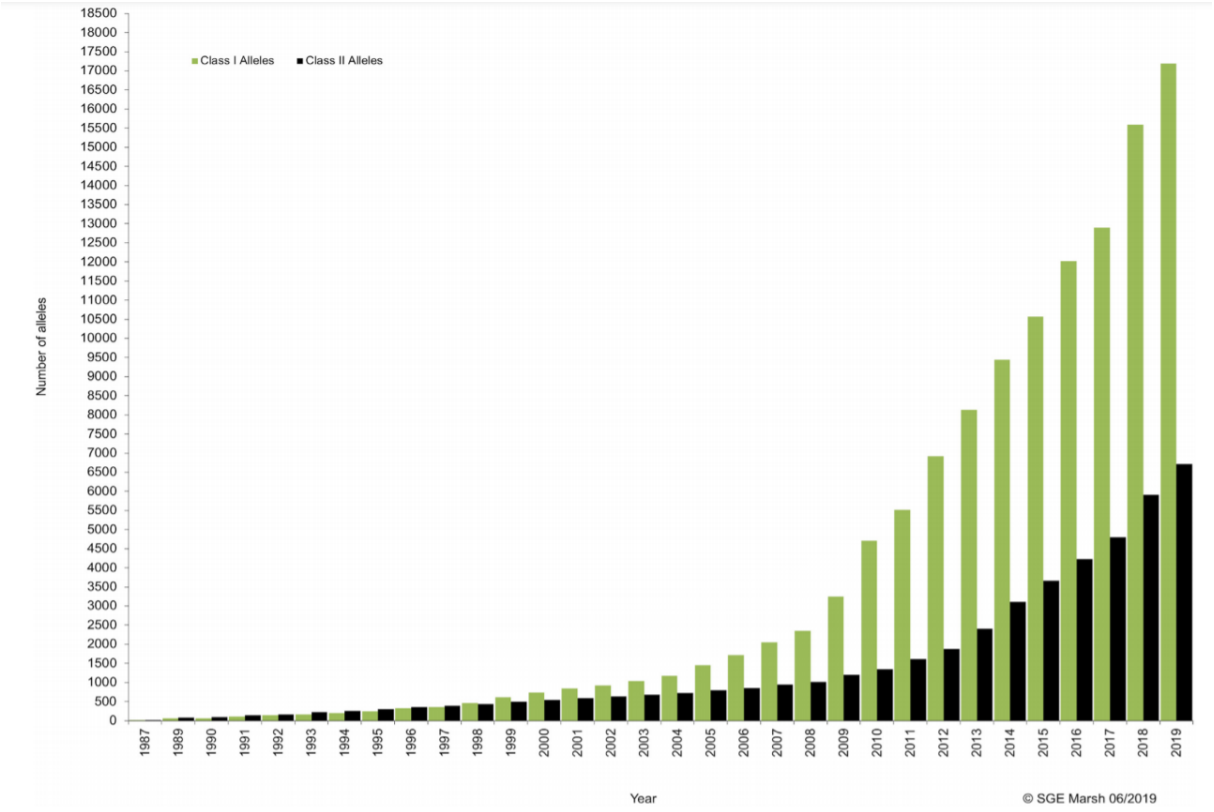

Figure A.0.1: The number of known HLA alleles during the last 30 years. These are named by the WHO Nomenclature Committee for Factors of the HLA System registered in the IPD-IMGT/HLA database (https://www.ebi.ac.uk/ipd/index.html). Next-generation sequencing has helped increase the rate of discovery, so that more alleles were registered in the first 3 months of 2019 than were present in the database at the turn of the milennium[5].*Figure taken from [5].*

Numbers of HLA Alleles	
HLA Class I Alleles	20,597
HLA Class II Alleles	7,723
HLA Alleles	28,320
Other non-HLA Alleles	466
Number of Confidential Alleles	2

HLA Class I						
Gene	*A*	*B*	*C*	*E*	*F*	*G*
Alleles	6,291	7,562	6,223	256	45	82
Proteins	3,896	4,803	3,618	110	6	22
Nulls	325	253	272	7	0	4

HLA Class I - Pseudogenes											
Gene	*H*	*J*	*K*	*L*	*N*	*P*	*S*	*T*	*U*	*V*	*W*	*Y*
Alleles	61	19	6	5	5	5	7	8	5	3	11	3
Proteins	0	0	0	0	0	0	0	0	0	0	0	0
Nulls	0	0	0	0	0	0	0	0	0	0	0	0

HLA Class II													
Gene	*DRA*	*DRB*	*DQA1*	*DQA2*	*DQB1*	*DPA1*	*DPA2*	*DPB1*	*DPB2*	*DMA*	*DMB*	*DOA*	*DOB*
Alleles	29	3,536	264	38	1,930	216	5	1,654	6	7	13	12	13
Proteins	2	2,476	114	11	1,273	80	0	1,064	0	4	7	3	5
Nulls	0	149	6	0	84	5	0	84	0	0	0	1	0

HLA Class II - DRB Alleles									
Gene	*DRB1*	*DRB2*	*DRB3*	*DRB4*	*DRB5*	*DRB6*	*DRB7*	*DRB8*	*DRB9*
Alleles	2,838	1	363	180	142	3	2	1	6
Proteins	1,973	0	272	121	110	0	0	0	0
Nulls	93	0	17	21	18	0	0	0	0

Other non-HLA Genes					
Gene	*HFE*	*MICA*	*MICB*	*TAP1*	*TAP2*
Alleles	6	224	212	12	12
Proteins	4	104	37	6	5
Nulls	0	5	204	1	0

Figure A.0.2: Overview of the number of HLA-alleles for each gene. Taken from the IMGT-HLA database (https://www.ebi.ac.uk/ipd/imgt/hla/stats.html) on the 6th of November 2020.

APPENDIX B

Typing accuracies for individual HLA genes

This appendix shows how well Kourami, HLA*LA, Optitype, HISAT-genotype and STC-Seq perfomed using the full gold standard dataset for each of the five genes HLA-A, -B, -C, -DRB1 and -DQB1.

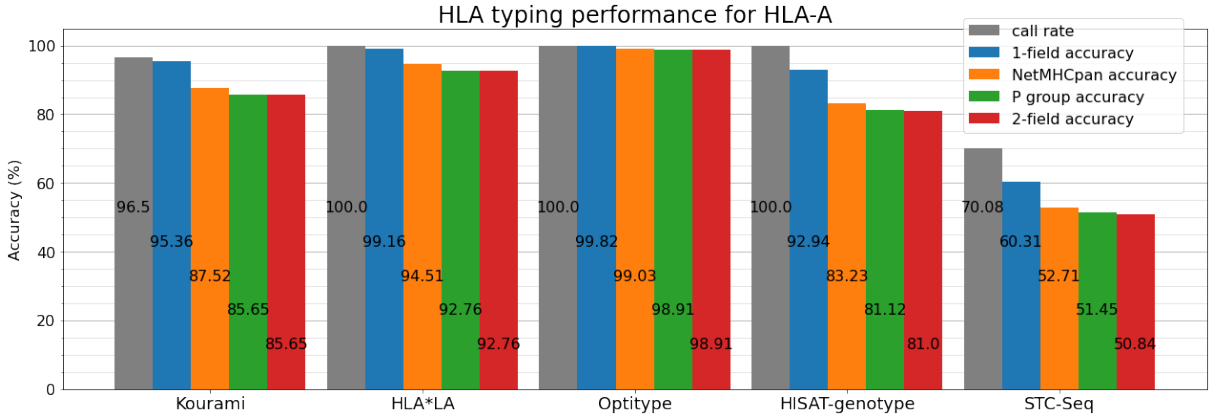

Figure B.0.1: Typing accuracy for HLA-A

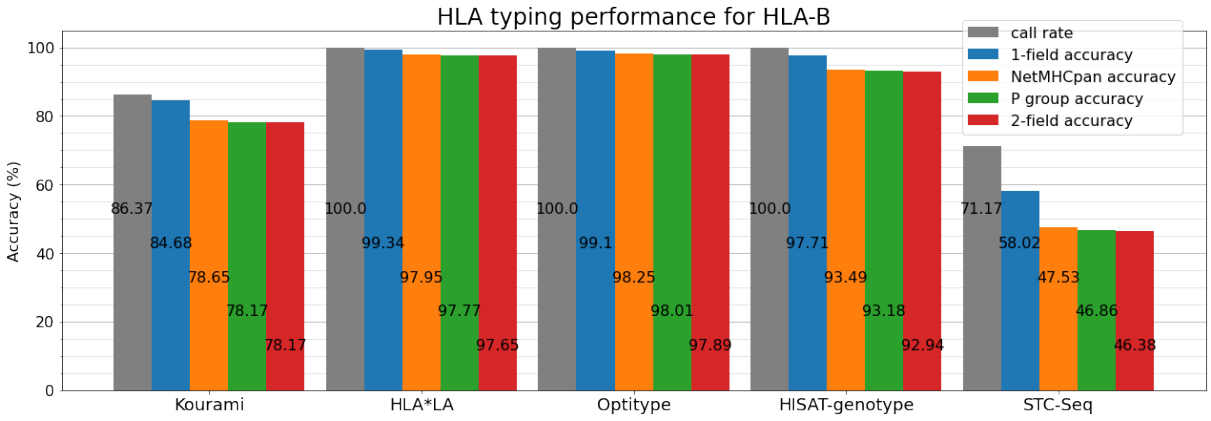

Figure B.0.2: Typing accuracy for HLA-B

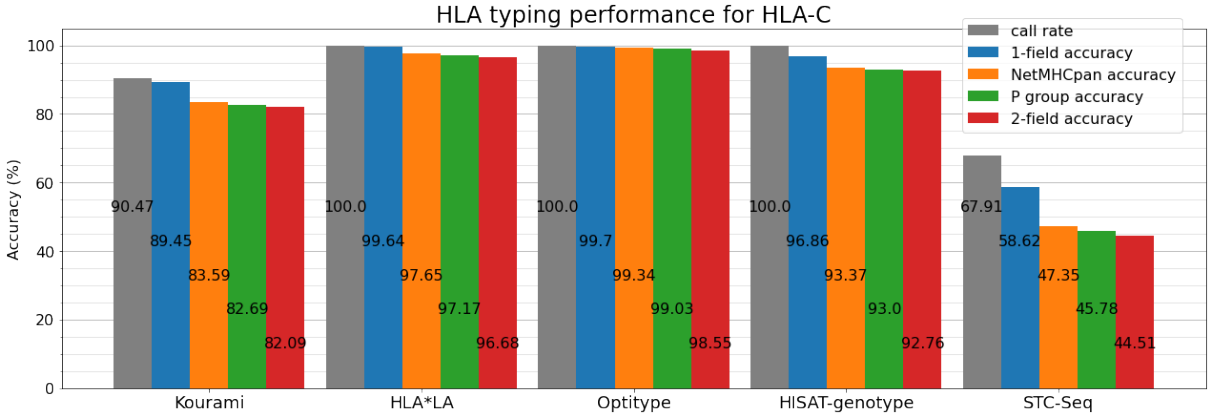

Figure B.0.3: Typing accuracy for HLA-C

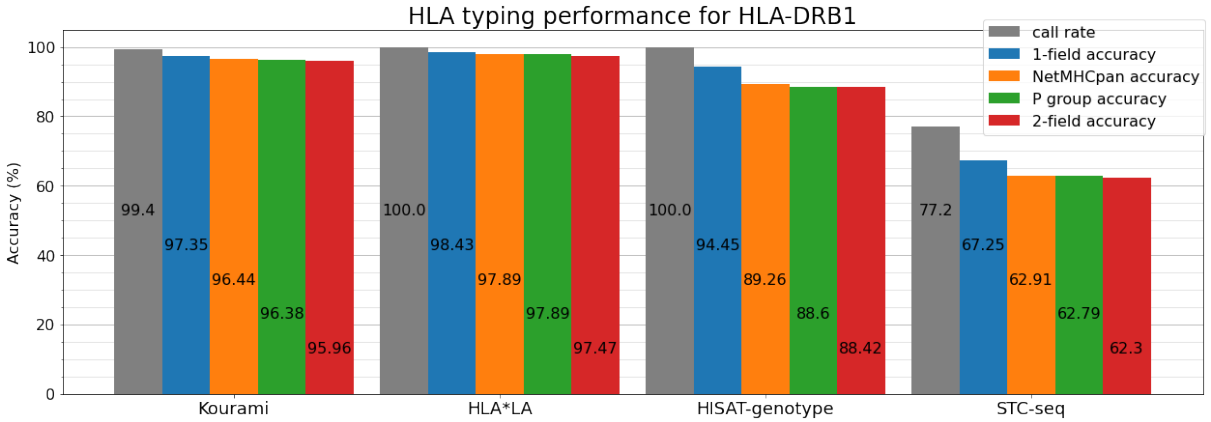

Figure B.0.4: Typing accuracy for HLA-DRB1

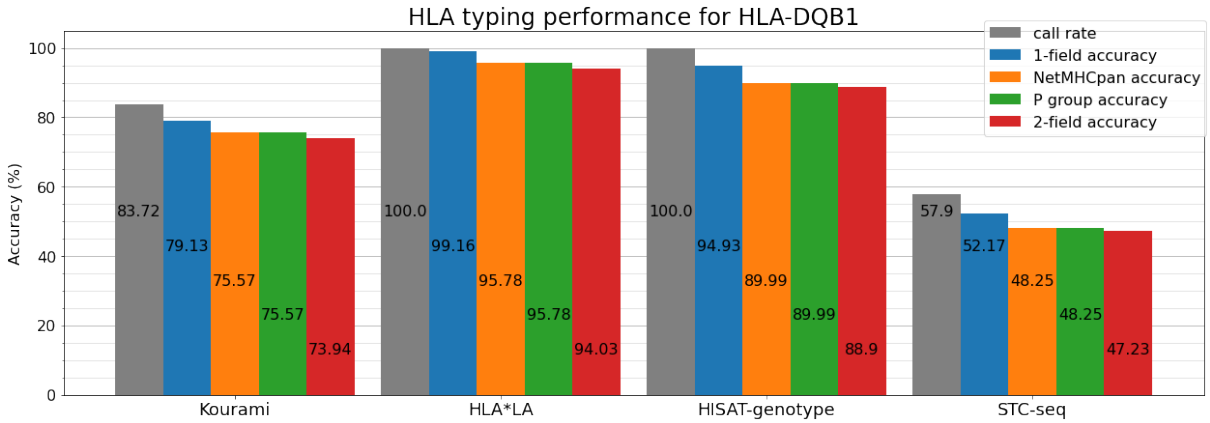

Figure B.0.5: Typing accuracy for HLA-DQB1